W NUMERZE:

Komputer w rękach zwycięzcy

Szyfr już złamany. Na ogłoszony w pierwszym numerze „IKS'a", informatycznego dodatku „Żołnierza Wolności", konkurs wpłynęło kilkanaście tysięcy prawidłowych rozwiązań. Zwycięzcę wybrał los. Pięcioletnia Monika wylosowała trzy koperty. Nadawcą pierwszej i zarazem triumfatorem naszego konkursu okazał się Bogdan Olszewski z Sosnowca. Dwie drugie nagrody otrzymują Bożena Silezin-Budek i Michał Stajszczak — oboje z Warszawy.

Z tej to okazji odbyła się w redakcji miła uroczystość wręczenia nagród. Redaktor naczelny „Żołnierza Wolności" płk Zdzisław Janoś wręczył zwycięzcy zestaw komputerowy ufundowany przez znaną na naszym rynku firmę ICL. Sprzęt i kilkanaście programów narzędziowych służyć będą inżynierowi z Sosnowca i jak zapowiedział pan Bogdan spróbuje na nim rozwiązać nawet niektóre zadania profesjonalne (oczywiście na miarę możliwości).

Natomiast dwa radioodbiorniki „Maria" trafiły do rąk warszawiaków, którym przysłowiowego łutu szczęścia zabrakło do wylosowania głównej nagrody.

A teraz jeszcze kilka słów o konkursie. Naszym zdaniem odczytanie kryptogramu nie było zadaniem prostym. Najważniejszą sprawą było oczywiście odnalezienie klucza, a potem krok po kroku należało odczytać niezrozumiały ciąg znaków, w którym znalazły się dodatkowe utrudnienia. Zatem najważniejszy był klucz. W kolejnych numerach staraliśmy się podać o nim bliższe informacje. Ale ten trud okazał się niepotrzebny, już wkrótce bowiem po ogłoszeniu konkursu zaczęły napływać rozwiązania — prawidłowe oczywiście!!!

Po odczytaniu nasz kryptogram miał postać:
WSZYSTKIE BOJOWE
OPERACJE MIAŁY
JEDEN AZYMUT:
+ WOLNA POLSKA +

Redaktor naczelny „ŻW" płk Zdzisław Janoś przekazuje komputer zwycięzcy — Bogdanowi Olszewskiemu.

A to już wszyscy laureaci w komplecie i ostatnie gratulacje.

pozostałe bajty programu. Na końcu programu wyświetlania wstawimy rozkaz JVB i adres początku programu wyświetlania. Teraz zapamiętujemy te bajty w RAM. Mogą one być w dowolnym miejscu; upewnijmy się tylko, że nie pokrywają się z obszarem grafiki gracz—pocisk lub wzorcami znaków i nasz JVB wskazuje początek programu wyświetlania. Następnie musimy wyłączyć ANTIC na ułamek sekundy, na czas przepisania adresu programu wyświetlania. Zrobimy to przez wstawienie 0 do SDMCTL pod adresem 559 ($22F). Wtedy zapamiętamy adres nowego programu wyświetlania do komórek o adresach 560 i 561 ($230 i $231). W końcu włączymy ANTIC wstawiając 34 ($22) do SDMCTL. W czasie powrotu pionowego, gdy ANTIC jest nieaktywny, system operacyjny wprowadzi te nowe wartości do jego licznika programu.

Działanie z własnym programem wyświetlania

Pamięć ekranu może być umieszczona gdziekolwiek w przestrzeni adresowej komputera. Normalnie program wyświetlania określa początek pamięci ekranu w pierwszym rozkazie wyświetlania — rozkaz LMS. Jednak ANTIC może wykonać nowy rozkaz LMS w każdej linii programu wyświetlania. W ten sposób informacja z całej przestrzeni adresowej komputera może być wyświetlona na jednym ekranie. Może to być cenne przy ustawianiu niezależnych okien tekstowych.

Istnieje kilka ograniczeń w umieszczaniu pamięci ekranu w RAM komputera. Po pierwsze, nie może ona przekraczać granicy adresu 4 KB. Jeśli nie możemy tego uniknąć (jak w trybie 8 BASIC'a, który używa 8 KB RAM), musimy ponownie załadować licznik pamięci ekranu za pomocą nowego rozkazu LMS. Po drugie, jeśli chcemy używać dowolnej procedury systemu operacyjnego (SO) obsługującej ekran, musimy dostosować się do konwencji używanych przez SO. Może to być dosyć trudne, gdy używamy zmodyfikowanego programu wyświetlania z BASIC'a. Jeśli zmieniliśmy standardowy program wyświetlania i próbujemy użyć instrukcji **PRINT** lub **PLOT**, system operacyjny wykona je tak, jakby program wyświetlania był niezmieniony. Spowoduje to prawdopodobnie uszkodzenie obrazu.

Istnieją trzy możliwe przyczyny zniknięcia obrazu podczas próby wyprowadzania informacji. Po pierwsze, BASIC może odmówić wykonania operacji na ekran, jeśli wybrany jest tryb graficzny, w którym taka operacja jest niemożliwa. System operacyjny przechowuje wartość aktualnego trybu graficznego w komórce o adresie 87 ($57). Możemy „przekonać" SO do współpracy przez wstawienie tam innej wartości. Pamiętajmy o wstawieniu liczby określającej tryb BASIC'u, nie ANTIC'u.

Drugi rodzaj uszkodzenia może wystąpić podczas mieszania linii obrazu wymagających różnych ilości bajtów pamięci ekranu. Pewne linie obrazu wymagają 40 bajtów na linię, inne — 20 bajtów, a niektóre tylko 10 bajtów na linię. Powiedzmy, że wstawiamy 20-bajtową linię obrazu do programu wyświetlania z 40-bajtową linią obrazu. Podczas wyprowadzania tekstu na ekran wszystko powyżej tej linii jest wyświetlane bez zmian, poniżej niej znaki są przesunięte o 20 pozycji w prawo. Jest tak, ponieważ system operacyjny przyjmuje, że każda linia wymaga 40 bajtów i odpowiednio pozycjonuje znaki. Lecz ANTIC, po napotkaniu takiej linii, wysyła tylko 20 bajtów. System operacyjny uważa, że powinna być linia 40-bajtowa. ANTIC interpretuje pozostałe 20 bajtów jako należące do następnej linii i tam je wyświetla. Powoduje to, że linia następna i dalsze będą przesunięte o 20 pozycji w prawo.

Jedynym sposobem uniknięcia tego problemu jest odstąpienie od używania instrukcji PRINT i PLOT do wyprowadzania informacji na ekran podczas stosowania własnego programu wyświetlania. Szybkim i prostym rozwiązaniem jest zorganizowanie ekranu w grupy linii, które zawie-

```
XS 185 ? :? "CZY CHCESZ MIEC WIDO
   CZNE OSIE X-Y";:INPUT A$:IF A$
   (1,1)="T" THEN W=1
TT 190 LST=1536
UD 195 GOSUB 30000
LE 200 POKE 559,0
HN 210 POKE 560,0
LH 220 POKE 561,6
RA 230 POKE 559,34
AZ 240 X=0:GOSUB 31000
ZM 250 POKE 87,0:POKE 752,1
EN 260 POSITION 13,0:? #6;"Wykres
    funkcji"
FR 265 POSITION 13,1:? #6;"------
   --------"
XU 268 X=1:GOSUB 31000:POKE 87,2:
   POSITION 0,0:? #6;"
            "
LH 270 ON N GOTO 1000,1100,1200,1
   300,1400,1500,1600,1700
BG 300 X=2:GOSUB 31000
WL 310 POKE 87,8:COLOR 1
SR 380 IF W<>1 THEN 410
KS 390 FOR I=0 TO 319 STEP 4:PLOT
    I,96:NEXT I
II 395 FOR I=0 TO 191 STEP 3:PLOT
    160,I:NEXT I
HJ 400 REM WYKRESLANIE
GI 410 FOR T=0 TO 360*U STEP KROK
JO 420 Q=T/57.3
YT 430 ON N GOTO 510,520,530,540,
   550,560,570,580
LL 500 REM ROWNANIA DLA R
FE 510 R=B*Q:GOTO 610
UV 520 R=A*(1+COS(Q)):GOTO 610
XL 530 R=A*(1-SIN(Q)):GOTO 610
YC 540 R=A*SIN(B*Q):GOTO 610
XB 550 R=A*COS(B*Q):GOTO 610
GJ 560 R=COS(A*SIN(B*Q)):GOTO 610
FH 570 R=SIN(A*COS(B*Q)):GOTO 610
ZW 580 R=A:GOTO 610
LC 600 REM WYKRESLANIE X,Y
UA 610 X=INT((R*COS(Q))*SK)
WJ 620 Y=INT((R*SIN(Q))*SK)
TR 630 IF ABS(X)>159 OR ABS(Y)>95
    THEN 670
IH 640 IF T=0 THEN PLOT 160+X,96-
   Y
ON 650 DRAWTO 160+X,96-Y
KO 660 NEXT T
IN 670 W=0
IH 680 X=3:GOSUB 31000:POKE 87,1
KC 690 POSITION 6,0:? #6;"nacisni
   j":POSITION 2,1:? #6;"DOWOLNY
   KLAWISZ"
HJ 700 U=1:OPEN #1,4,0,"K:":GET #
   1,X:POSITION 0,0:PUT #6,125
YJ 710 POSITION 0,0:? #6;"czy kas
   owac ekran ?":GET #1,X:PUT #6,
   125
```

```
NL 720 IF X<>84 THEN GOTO 90
IO 730 X=2:GOSUB 31000:POKE 87,0:
   PUT #6,125:GOTO 90
QM 900 X=1:GOSUB 31000:POKE 87,2:
   POSITION 0,0:PUT #6,125:RETURN
XI 1000 POSITION 0,0:? #6;"
    R=";B;"*Q"
NR 1010 GOTO 300
ZL 1100 POSITION 0,0:? #6;"    R="
   ;A;"*(1+COS(Q))"
NT 1110 GOTO 300
JI 1200 POSITION 0,0:? #6;"    R="
   ;A;"*(1-SIN(Q))"
NV 1210 GOTO 300
EK 1300 POSITION 0,0:? #6;"    R="
   ;A;"*SIN(";B;"*Q)"
NX 1310 GOTO 300
YE 1400 POSITION 0,0:? #6;"    R="
   ;A;"*COS(";B;"*Q)"
NZ 1410 GOTO 300
AL 1500 POSITION 0,0:? #6;" R=COS
   (";A;"*SIN(";B;"*Q))"
OB 1510 GOTO 300
YP 1600 POSITION 0,0:? #6;" R=SIN
   (";A;"*COS(";B;"*Q))"
OD 1610 GOTO 300
KO 1700 POSITION 0,0:? #6;"
    R=";A
OF 1710 GOTO 300
DH 20000 REM ********************
IW 20001 REM *   Podprogram    *
DS 20002 REM * uklada program  *
OC 20003 REM *  wyswietlania    *
EB 20004 REM ********************
MK 30000 ON (START=1) GOTO 30380:
   SEG=1:DIM S$(12000):S$(1)=CHR$
   (0):S$(12000)=S$:S$(2)=S$:STAR
   T=1:I=1
IA 30010 LOK=LST
NX 30020 GEKR(0)=ADR(S$)
UX 30030 GRAN=INT((LST/1024+1)*10
   24)
UX 30040 GRAN2=INT(GEKR(0)/4096+1
   )*4096
BY 30050 REM
CC 30060 REM
CG 30070 REM
PM 30080 GEKR(SEG)=GEKR(SEG-1)
NK 30100 TRYB=NRGR(1+2*(I-1)):REP
   EAT=NRGR(2*I):I=I+1
YU 30110 IF TRYB<0 THEN OP=65:ADD
   R=LST:GOSUB 30330:RETURN
CM 30120 PRZYR=40
UM 30130 IF TRYB>=6 AND TRYB<=12
   THEN PRZYR=20
FM 30140 IF TRYB=8 OR TRYB=9 THEN
    PRZYR=10
WV 30150 FOR X=1 TO REPEAT
VO 30160 IF LOK<>GRAN-3 THEN 3020
   0
```

rają całkowite wielokrotności standardowych wymagań bajtów. To znaczy, nie wstawiamy 20-bajtowej linii do 40--bajtowego obrazu; zamiast tego wstawiamy dwie linie 20--bajtowe lub jedną 20-bajtową i dwie 10-bajtowe. Dopóki zachowujemy odpowiednie całkowite wielokrotności, unikamy przesuwania poziomego.

Rozwiązanie to podkreśla trzeci problem związany z własnymi programami wyświetlania: przesunięcie pionowe. System operacyjny pozycjonuje dane ekranu w pionie przez obliczenie liczby bajtów do przeskoczenia w dół od góry ekranu. W standardowej, 40-bajtowej linii obrazu BASIC może wstawić znaki w 10 linii po przeskoczeniu 360 bajtów. Jeśli mamy wstawić cztery linie 10-bajtowe, BASIC zakończy je trzy linie dalej w dół ekranu niż moglibyśmy oczekiwać. Ponadto różne linie obrazu zużywają różne ilości linii ekranu, zatem pozycja na ekranie nie będzie taka, jak oczekiwaliśmy, jeśli nie weźmiemy pod uwagę kosztów linii ekranu.

Jak możemy zobaczyć, mieszane tryby obrazu mogą być trudne do użycia. Często musimy oszukiwać system operacyjny, aby otrzymać żądane obrazy. W celu wydrukowania lub wykreślenia w wybranym oknie, wstawiamy **(POKE)** numer trybu BASIC'u tego okna do komórki o adresie 87, następnie wstawiamy adres lewego, górnego punktu elementarnego tego okna do komórek 88 i 89 ($58 i $59). W trybach znakowych wykonujemy **POSITION 0,0** w celu ustawienia lewego górnego rogu okna. W trybach kreślenia wszystkie instrukcje **PLOT** i **DRAWTO** będą wykonywane względem lewego górnego rogu okna jako początku układu współrzędnych.

Własny program wyświetlania może być używany do wytworzenia interesujących obrazów. Jego najbardziej oczywistym zastosowaniem jest mieszanie tekstu i grafiki. Na przykład, możemy przygotować ekran z wyraźnym tytułem w trybie 2 BASIC'a, średniej wielkości podtytułem w trybie 1 i małym napisem w trybie 0. Następnie możemy przejść do obrazu w trybie 8 BASIC'a w środku, z pewnym dłuższym tekstem u dołu ekranu. Dobry przykład tej techniki dostarczany jest przez obraz w programie Atari States and Capitals.

Zastosowania programów wyświetlania

Jednym z prostych zastosowań zmodyfikowanego programu wyświetlania jest pionowe odsunięcie linii na ekranie przez wstawienie pustych linii. Dzięki temu możemy uwypuklić krytyczne komunikaty i zwiększyć czytelność pewnych obrazów.

Innym ważnym zastosowaniem manipulacji programem wyświetlania jest umożliwienie wykorzystania własności niedostępnych z BASIC'a. Istnieje tryb tekstowy dostarczany przez ANTIC, którego nie obsługuje BASIC. Tylko manipulowanie programem wyświetlania umożliwia nam dostęp do tego trybu. Istnieją także przerwania programu wyświetlania i możliwości płynnego przesuwania obrazu, które są dostępne tylko po zmodyfikowaniu programu wyświetlania.

Manipulowanie rozkazem LMS i jego argumentami oferuje wiele możliwości twórczemu programiście. Na przykład, przez zmianę LMS w czasie powrotu pionowego strumienia elektronów, programista może zmieniać obrazy wyświetlane na ekranie. Można to robić z małą szybkością, w celu wymiany wcześniej przygotowanych obrazów, bez konieczności ponownego kreślenia każdego z nich. Dane obrazu mogą przebywać w pamięci nawet wtedy, gdy nie są używane, lecz mogą być dostępne w każdej chwili. Technika ta może być także używana do animacji. Przez wymianę ciągu obrazów możemy uzyskać cykliczną animację. Program, który to będzie realizował, może manipulować tylko dwoma bajtami adresu do wyświetlania wielu tysięcy bajtów RAM.

Możliwe jest także nakładanie obrazów przez ich wymianę z dużą szybkością. Oko ludzkie ma czas rozdzielczości

```
QN 30170 OP=1:ADDR=GRAN:GOSUB 303
   30
TT 30180 LOK=LOK+1:GRAN=GRAN+1023
YV 30190 GOTO 30200
IV 30200 POKE LOK,TRYB
SR 30220 IF (SEG<>1 OR X<>1) AND
   (GRAN2-GEKR(SEG))=PRZYR) THEN
   30280
SL 30240 OP=TRYB+64
AO 30250 ADDR=GEKR(SEG)
FT 30260 IF GRAN2-GEKR(SEG)<PRZYR
    THEN GRAN2=GRAN2+4096:ADDR=AD
   DR+PRZYR
PP 30270 GOSUB 30330
CK 30280 LOK=LOK+1
TR 30290 GEKR(SEG)=GEKR(SEG)+PRZY
   R
MS 30300 NEXT X
HH 30305 IF TRYB=15 THEN GEKR(SEG
   )=GEKR(SEG)+PRZYR
VF 30310 SEG=SEG+1
BN 30320 GOTO 30080
ZE 30330 POKE LOK,OP
BX 30340 LOK=LOK+1
UW 30350 POKE LOK,ADDR-(INT(ADDR/
   256)*256)
CF 30360 LOK=LOK+1
PW 30370 POKE LOK,INT(ADDR/256)
EO 30380 RETURN
EJ 30900 REM ********************
JX 30910 REM *   Podprogram     *
TY 30920 REM *  ustala adres    *
WZ 30930 REM * pamieci ekranu   *
EZ 30940 REM ********************
PT 31000 POKE 88,GEKR(X)-(INT(GEK
   R(X)/256)*256)
RB 31010 POKE 89,INT(GEKR(X)/256)
   :RETURN
```

około 1/16 sekundy, zatem program może krążyć między trzema obrazami, każdy co 1/50 s, tak że każdy powtarza się co 1/16 sekundy. W ten sposób jednocześnie może wystąpić na ekranie do trzech obrazów. Oczywiście istnieją pewne wady tej metody. Pierwsza, oddzielne obrazy mogą zajmować sporo RAM. Druga, każdy obraz będzie rozmyty, ponieważ pokazywany jest tylko jedną trzecią czasu. Oznacza to, że tło wszystkich obrazów powinno być czarne, a każdy obraz — jasny. Ponadto podczas używania tej techniki wystąpią pewne nieprzyjemne migotania ekranu. Opisana technika może być także używana do zwiększenia rozdzielczości kolorów i jaskrawości. Dzięki wymianie trzech wersji tego samego obrazu, gdy każda wersja akcentuje jeden kolor lub zakres jaskrawości, dostępny jest większy zakres kolorów i jaskrawości.

Automatyczne tworzenie programu wyświetlania

Dla tych czytelników, którym udało się dobrnąć do końca poprzedniego artykułu oraz nie przestraszyli się trudności, jakie trzeba pokonać podczas tworzenia własnego programu wyświetlania, mamy niespodziankę. Pisaliśmy wcześniej, że na początku musimy zaplanować obraz za pomocą kartki papieru, opisaliśmy kolejno, krok po kroku, jakie czynności należy wykonać, aby uzyskać własny program wyświetlania. Omówiliśmy trudności, jakie należy pokonać, aby uzyskać żądane efekty na ekranie, przedstawiliśmy możliwe błędy i ich skutki. A teraz prezentujemy program, który uwalnia nas od troszczenia się o wszelkie szczegóły tworzenia programu wyświetlania i wykonuje za nas wszystkie obliczenia. Musimy, oczywiście, dostarczyć niezbędnych danych, do których należą: liczba segmentów, na jakie chcemy podzielić ekran, numery trybów ANTIC'a w poszczególnych segmentach oraz liczbę linii obrazu dla każdego segmentu. Informacje te podajemy w liniach 8 (ilość segmentów) oraz 10—43 (numer trybu oraz ilość linii obrazu dla każdego segmentu) programu nr 1. Program wyświetlania umieszczony będzie na stronie 6 pamięci RAM, od adresu 1536 (linia 140).

Linie 30000—30380 zawierają procedurę tworzenia programu wyświetlania na podstawie danych umieszczonych w tablicy NRGR, która w każdym wierszu zawiera numer trybu i liczbę linii obrazu. Procedura ta uwzględnia ograniczenia maksymalnej długości programu wyświetlania i maksymalnej ilości danych ekranu. Celowo nadaliśmy jej tak wysokie numery wierszy, aby każdy z zainteresowanych mógł go wykorzystać we własnych programach.

Podprogram umieszczony w liniach 30900—31010 aktualnie nie jest używany, lecz ułatwia rozwiązanie problemu adresowania ekranu dla poszczególnych segmentów. Jako dane wejściowe wykorzystuje on numer segmentu (zmienna X) oraz tablicę GEKR, utworzoną przez podprogram nr 1, która zawiera adresy początku pamięci ekranu dla każdego segmentu.

Program nr 2 demonstruje wykorzystanie omówionych podprogramów. Ma już określone następujące dane: liczba segmentów (zmienna N) oraz numery trybów i liczba linii obrazu dla każdego segmentu (linia 42). Przygotowany został w taki sposób, aby uniknąć dwukrotnego wprowadzania podprogramów nr 1 i 2 z poprzedniego programu. Należy po prostu program numer 2 dołączyć do programu nr 1 (za pomocą ENTER "C") w celu otrzymania gotowego programu. Utworzony przez niego program wyświetlania generuje obraz zawierający: 2 linie w trybie 0, jedną linię w trybie 2, 144 linie w trybie 8 oraz dwie linie w trybie 1 (są to tryby BASIC'u).

W liniach 200—300 następuje przekazanie systemowi operacyjnemu adresu programu wyświetlania. Linie 240—265 umieszczają na ekranie napisy w trybie 0, linie 270—290 umieszczają napis w trybie 2, linie 300—360 powodują wyświetlenie rysunku w trybie 8, a linie 370—395 umieszczają napisy w trybie 1.

Widoczne jest, że umieszczenie tekstu lub grafiki w wybranym segmencie wymaga: podania numeru segmentu, wywołania podprogramu nr 2, wstawienia do komórki o adresie 87 numeru trybu BASIC'a używanego w tym segmencie oraz wyprowadzenia informacji właściwych dla danego trybu. W wybranym segmencie adresowanie linii obrazu odbywa się względem lewego górnego rogu segmentu, który aktualnie jest początkiem układu współrzędnych tego segmentu.

Program nr 3 przygotowany został przez nas w celu zademonstrowania, jakich możliwości dostarcza manipulowanie programem wyświetlania. Najciekawszą z nich jest uzyskanie obrazu o wysokości 240 linii ekranu. Dzięki temu uzyskaliśmy dodatkowe 48 linii na umieszczenie różnorodnych napisów przy niezmienionej liczbie linii przeznaczonych na rysunek (192). Przyznajemy, jest to nieco rozbieżne z informacjami zawartymi w poprzednim artykule, ale tamte dane pochodzą z wszelkiego rodzaju opisów i instrukcji, a ten efekt wynikł z naszych poszukiwań i prób. Wymagało to wprowadzenia pewnych modyfikacji do procedury tworzenia programu wyświetlania, dlatego zamieszczamy pełny wydruk gotowego programu. Innym ciekawym efektem jest wykorzystanie dwóch obrazów: jeden z nich posiada standardowy program wyświetlania, a drugi — przygotowany przez nas. Dodatkową ciekawostką jest fakt, że dane do wyświetlania drugiego obrazu umieszczone są

wewnątrz programu, w zmiennej tekstowej S, a nie jak zwykle w obszarze pamięci poza programem.

Oprócz tego zadaniem programu nr 3 jest wyznaczenie i wyświetlenie wykresów takich jak: spirala, kardioida, rozeta, itp. Na początku prezentuje menu i prosi o wybranie funkcji, a następnie o jej parametry, przyrost i skalę. Przyrost jest ilością stopni (krokiem), które komputer używa do zwiększenia kąta T od 0 do 360 stopni. Musimy określić, czy preferujemy szybkość działania, czy dokładność. Mały przyrost, np.: 0,1, będzie kreślił dokładny wykres bardzo wolno. Większy przyrost, np.: 5.0 wykreśli szybciej lecz mniej dokładnie. Dobrym kompromisem jest 1.0. Skala umożliwia dobranie wielkości rysunku na ekranie. Jej wartość w zakresie 10 do 100 powinna być wystarczająca dla większości funkcji. Linie 175—180 sprawdzają, czy nie wprowadziliśmy wartości zerowej, która mogłaby przerwać działanie programu i pojawienie się komunikatu błędu. Linia 182 to pytanie, czy mają być wyświetlone osie x—y układu współrzędnych, a linie 390—395 kreślą je. Linie 400—690 zawierają procedury obliczania i kreślenia x i y. W linii 410 występuje zmienna U dla funkcji nr 1, która określa ilość obrotów spirali, jej wartość może być zmieniona w linii 170. Linia 430 kieruje program do obliczania wartości funkcji wybranej na początku. Linie 610—620 obliczają współrzędne x, y. Linia 630 sprawdza, czy nie przekroczyliśmy granic ekranu. W przypadku przekroczenia zatrzymuje kreślenie i nie dopuszcza do wystąpienia komunikatu błędu. W liniach 680—690 program czeka na naciśnięcie klawisza w celu przejścia do kreślenia innej funkcji i ewentualnego oczyszczenia ekranu. Jeśli wartości funkcji są zbyt duże do kreślenia, od razu pojawia się propozycja naciśnięcia klawisza (należy zmniejszyć wartość skali).

Ze względu na to, że nie każde wartości parametrów dają interesujące efekty na ekranie, proponujemy wykorzystać następujące wartości dla poszczególnych funkcji:

1)	A=1,	B=1,	KROK=60,	SKALA=4
	A=1,	B=1,	KROK=45,	SKALA=5
	A=1,	B=1,	KROK=120,	SKALA=5
2)	A=2,	B=1,	SKALA=20	
3)	A=2,	B=1,	SKALA=20	
4)	A=2,	B=3,	SKALA=80	
6)	A=2,	B=6,	SKALA=90	
	A=2,	B=2,	SKALA=90	
	A=3,	B=1,	SKALA=90	
	A=1,	B=100,	SKALA=90	
7)	A=1,	B=1,	SKALA=90	
	A=4,	B=2,	SKALA=90	

Mamy nadzieję, że przedstawione propozycje zachęcą dociekliwych czytelników do dalszych poszukiwań interesujących efektów graficznych.

Do wprowadzania programów do pamięci komputera radzimy wykorzystać Edytor BASIC'u zamieszczony w nr. 4 „IKS-a" z bieżącego roku. Pamiętajmy: dwa znaki z lewej strony numeru linii nie są częścią programu, lecz sumą kontrolną danej linii, dlatego nie wprowadzamy ich do komputera.

Programy należy przechować w pamięci zewnętrznej (taśmowej lub dyskowej) przed ich uruchomieniem albo za pomocą funkcji „L" Edytora BASICU'u, albo za pomocą instrukcji LIST, w przypadku wprowadzania programu w sposób konwencjonalny.

NOWOŚCI

Pomimo wprowadzenia na rynek rewelacyjnych 16-bitowych komputerów domowych serii ST, Atari dba również o posiadaczy sprzętu ośmiobitowego (niesłusznie uważanego u nas za przestarzały). Jednym z przejawów dbałości jest zestaw RAMBO XL, umożliwiający rozszerzenie pamięci RAM komputera 800 XL do 256 KB i uczynienia go zgodnym ze 130 XE. Dzięki temu możliwe jest efektywne wykorzystanie systemu operacyjnego DOS 2.5 ze standardowym RAM-dyskiem o pojemności 64 KB oraz najnowszej wersji języka BASIC XE. Nowy handler RD.COM dostarczany ze Sparta DOS Construction Set, udostępnia RAM-dysk o pojemności 192 KB. Jest to pamięć wystarczająca do skopiowania dyskietki o podwójnej gęstości zapisu w jednym przebiegu!

Powyższe rozszerzone możliwości „poczciwej 800" dostępne są również rodzimym posiadaczom tych komputerów i w dodatku za złotówki. KAREN, firma zajmująca się naprawami sprzętu ATARI, oprócz montowania BASIC'u XL (za około 26 tysięcy złotych), oferuje również rozszerzenie pamięci 800 XL do 256 KB za około 70 000 złotych oraz BASIC XE (najnowszy produkt Optimized System Software) za około 30 000 złotych.

Tomasz MROWIEC, Ludwik PIELA

ATARI

O zerowaniu zmiennych

W podręcznikach opisujących zasady programowania w języku Atari-Basic twierdzi się, że „po zatrzymaniu programu klawiszem RESET lub zakończeniu działania programu, w którym użyto instrukcji END, wszystkie zmienne są wyzerowane, tzn. zmienne liczbowe mają wartość 0, a tekstowe zawierają łańcuch spacji".

Zobaczmy, jak to jest w rzeczywistości? Posłuży nam do tego krótki program:

```
10 DIM A$(3)
20 INPUT A$,B,C
30 PRINT"A$=";A$: PRINT "B=";B: PRINT "C=";C
40 D = D + 1
50 PRINT"D=";D
60 END
```

Po uruchomieniu programu instrukcją RUN i wpisaniu wartości zmiennych

```
A$ = "ALA"
B = 1
C = 2
```

otrzymujemy na ekranie wynik działania programu w postaci

```
A$ = ALA
B = 1
C = 2
D = 1
```

Jeżeli teraz uruchomimy program instrukcją GOTO 30, otrzymamy ten sam wynik, przy czym D = 2. Po naciśnięciu klawisza RESET i uruchomieniu programu instrukcją GOTO 30, otrzymamy znów wynik (3) ale przy D = 3.

Przeprowadzone badania wskazują, że ani instrukcja END, ani klawisz RESET nie zerują zmiennych! Jaka zatem instrukcja powoduje ich zerowanie? Aby odpowiedzieć na to pytanie, wyeliminujemy z programu (1) instrukcję END. Po uruchomieniu programu instrukcją RUN i wprowadzeniu wartości zmiennych (2) otrzymujemy wynik (3). Jeżeli teraz wyeliminujemy linię 20 i uruchomimy program instrukcją RUN, otrzymamy wynik

```
A$ =
B = 0
C = 0
D = 1
```

Z powyższego wynika niezbicie, że to właśnie instrukcja **RUN** a nie **END** czy **RESET** powoduje wyzerowanie zmiennych. Nieuwzględnienie tego faktu wszędzie tam,

gdzie wielokrotnie uruchamiamy program (podprogram) instrukcją inną niż RUN, może prowadzić do powstania różnego rodzaju błędów. Można ich uniknąć nadając zmiennym pożądane wartości (w tym również zero) instrukcją podstawienia przed rozpoczęciem podprogramu czy pętli.

Oczywiście zmienne zeruje również instrukcja CLR. Używając jej należy jednak pamiętać, że zeruje ona wszystkie zmienne programu (co nie zawsze nam odpowiada), a po jej użyciu należy ponownie zadeklarować zmienne indeksowane i tekstowe.

Andrzej GRZESIAK

Pogwarki o CPC 464

Pogawędka druga — firmware, czyli zdolności wrodzone

Dawno, dawno temu, gdy nie było jeszcze komputerów osobistych, a z dużymi komputerami moje kontakty ograniczały się do przekazywania kart dziurkowanych i odbierania wyników, elektroniczną maszynę cyfrową traktowałem jak komplet bezdusznych, metalowych mebli, migocących kolorowymi lampkami i obracających bezustannie krążki taśm magnetycznych. Całkiem odmienne wrażenie wywarł na mnie pierwszy, „poznany" mikrokomputer. Początkowo do złudzenia przypominał wyrośnięty kalkulator, ale gdy tylko został uruchomiony, złudzenie prysło. Zupełnie, jakby zbudzona została żywa istota. Każde polecenie, każda instrukcja powodowała szybką reakcję, która przy szczątkowej nawet znajomości języka angielskiego mogła kojarzyć się z prymitywnym dialogiem. Oczywiście przyczyną takiego zachowania, martwego skądinąd, narzędzia jest program. Wiadomo również, że w każdym „szanującym się" mikrokomputerze aktualnie realizowany program (lub jego część) musi znajdować się w pamięci wewnętrznej maszyny. Zatem, aby natychmiast po włączeniu urządzenia wyświetlony został powitalny napis:

Amstrad 64K Microcomputer (v1)
© 1984 Amstrad Consumer Electronics plc and Locomotive Software Ltd.

a następnie zgłosił się interpreter języka BASIC, producent musiał umieścić w pamięci odpowiednie zestawy instrukcji. Co ważne, nie mogą one ulec wyzerowaniu przy wyłączeniu komputera, ani tym bardziej przypadkowemu uszkodzeniu. Pamięć maszyny podzielono więc na część przeznaczoną na programy użytkowników, zwaną RAM (Random Access Memory — pamięć o dostępie przypadkowym) i część, którą zarezerwował dla siebie producent, czyli ROM (Read Only Memory — pamięć tylko do odczytu). Wbudowanie niewymazywalnej pamięci ROM jest istotną cechą mikrokomputerów osobistych, ułatwiającą ich eksploatację użytkownikowi bez przygotowania informatycznego. Nie oznacza to jednak, że wszystkie mikrokomputery muszą mieć ROM. Gdy są go pozbawione, jedną z pierwszych operacji po włączeniu zasilania jest wczytanie z pamięci zewnętrznej (dysku, taśmy, a w starych maszynach również z dziurkowanej taśmy papierowej) odpowiedniego programu zarządzającego, zwanego systemem operacyjnym. Dopiero wówczas można wprowadzić program użytkowy. Analogicznie postępujemy chcąc wczytać CP/M 2.2, wtedy gdy do naszego AMSTRADa jest przyłączona stacja dysków. Różnica polega na tym, że CPC 464 jest wyposażony w ROM i w dodatku z bardzo interesującą zawartością, a zatem CP/M jest systemem alternatywnym, pozwalającym zwiększyć możliwości komputera. O tym, jak istotną rolę odgrywa umieszczone w pamięci stałej oprogramowanie, niech świadczy fakt, że wszyscy, decydujący się na samodzielne budowanie mikrokomputerów, z tym elementem mają zazwyczaj najwięcej kłopotów. Najczęściej radzą sobie, kopiując produkt znanej firmy. W ten sposób unikają możliwości popełnienia wielu błędów, często niezwykle trudnych do wykrycia oraz zapewniają swojemu dziełu bibliotekę programów użytkowych. Warto w tym miejscu zaznaczyć, że różnice w programach zarządzających są jedną z ważnych przyczyn, iż np. ZX SPECTRUM i AMSTRAD CPC 464, pomimo zastosowania w nich mikroprocesorów tego samego typu, nie są wymienne programowo.

Przeglądając prospekty mikrokomputerów łatwo zauważyć, że wielkość pamięci stałej jest różna dla różnych typów maszyn. Jednak rozumowanie, że większa pojemność ROM, to większy program, a co za tym idzie większe możliwości komputera oraz komfort pracy może być złudne. Zbyt wiele zależy tu od przyjętej koncepcji i zawartości oprogramowania, a co bardzo ważne — również od jego poprawności.

Czym zatem producent wypełnia stałą pamięć komputera? W przypadku AMSTRADa CPC 464 w 32 Kbajtach ROMu firma Locomotive Software Ltd. wpisała interpreter języka BASIC i tak zwany **FIRMWARE**, to znaczy zbiór niskopoziomowych, czyli napisanych w kodzie maszynowym podprogramów, które sterują pracą urządzeń zewnętrznych, synchronizują i nadzorują pracę elementów komputera. Podprogramy te mogą być w pełni wykorzystane przez programistów piszących w asemblerze, a zdecydowana większość z nich jest dostępna także w BASICu i innych językach.

FIRMWARE został podzielony na pakiety dotyczące poszczególnych elementów systemu. Tak na przykład pakiet o wdzięcznej nazwie **KEY MANAGER** (co nieodparcie kojarzy mi się z „dyrektorem od klawiszy") jest przeznaczony do obsługi klawiatury oraz joysticków i realizuje trzy poziomy operacji. Najniższy poziom, to badanie „czy?" i „które?" klawisze zostały wciśnięte, poziom średni — przyporządkowanie tym klawiszom odpowiednich wartości liczbowych, a poziom najwyższy to zamiana wartości liczbowych na właściwe znaki. KEY MANAGER zawiera także procedury obsługi przerwań realizowanych przez wciśnięcie klawiszy ESC lub SHIFT, CTRL i ESC. Kolejny pakiet — **TEXT VDU** jest programem pozwalającym zapisywać na ekranie monitora znaki tekstu oraz odczytywać je. Kontroluje osiem różnych „strumieni", z których każdemu może odpowiadać wydzielony obszar ekranu, zwany oknem tekstowym (window). TEXT VDU traktuje ustalone znaki jako kody kontrolne, co umożliwia różnorodne efekty jak ruch kursora lub zmiana koloru znaków. Dostęp do pojedynczych punktów graficznych ekranu, to znaczy ich zapalanie lub określanie ich aktualnego stanu oraz rysowanie linii umożliwia pakiet zwany GRAPHICS VDU. W jego skład wchodzą między innymi podprogramy umożliwiające posługiwanie się układem współrzędnych, określające aktualną pozycję kursora graficznego, określające kolory tzw. pióra i tła albo inaczej „papieru". Układy elektroniczne związane bezpośrednio z tworzeniem obrazu oraz pakiety TEXT VDU i GRAPHICS VDU dopasowuje do siebie pakiet pośredniczący — **SCREEN PACK**. Zawarte w nim procedury pozwalają wybrać tryb graficzny, otrzymać „mieniące się" barwy, uzyskać zamiast kolorów odpowiadające im odcienie szarości w razie posługiwania się monitorem monochromatycznym. Najważniejsza funkcja tego programu polega na zamianie współrzędnych użytecznych w TEXT VDU i GRAPHICS VDU na adresy w pamięci ekranu.

Również układ elektroniczny generujący dźwięk, czyli **SOUND CHIP**, ma swojego „dyrektora" w pamięci ROM. SOUND MANAGER, bo o nim mowa, pozwala zmieniać wysokość i czas trwania dźwięku, kształtować jego obwiednię, a tym samym brzmienie oraz synchronizować ze sobą dźwięki wytwarzane w trzech kanałach. W AMSTRADzie CPC 464 sterowanie dźwiękiem realizowane jest nieomal całkowicie programowo.

Problem, który w tym mikrokomputerze został rozwiązany chyba najlepiej ze wszystkich znanych mi urządzeń tej klasy, to zapisywanie zbiorów (w tym i programów) na taśmie kasetowej, oraz wczytywanie ich do pamięci komputera. Zbiory zapisywane są w oddzielonych od siebie blokach zawierających 2K (2048) bajtów lub mniej, w wypadku gdy jest to ostatni blok w zbiorze. Zbiory i bloki mają odpowiednie nagłówki, czyli informacje umieszczone na początku danych oraz zakończenia, które w pełni je charakteryzują. Takie rozwiązanie znacznie ułatwia kopiowanie i wczytywanie danych, a także organizację bibliotek programowych, gdyż w razie wystąpienia błędów czytania, wystarczy powtórzyć wprowadzanie zbioru od bloku, w którym błąd się pojawił. Poza tym poszczególne bloki zbioru mogą być kopiowane niezależnie jeden od drugiego. Sterowanie zapisem i odczytem informacji oraz mechanizmem magnetofonu kasetowego, wbudowanego w komputer odbywa się w pełni programowo i jest realizowane przez pakiet FIRMWARE nazywany **CASSETTE MANAGER**.

Innym pakietem odpowiedzialnym za współpracę komputera z urządzeniami zewnętrznymi jest **MACHINE PACK**. Powoduje on, że między innymi układy interfejsu równoległego (PPI — port A), odpowiedzialnego za tryb wyprowadzania danych, a w tym za współpracę z generatorem dźwięku oraz układy sterujące monitorem ekranowym (Cathode Ray Tube Controller) i układy wyjścia **CENTRONICS**, służącego do podłączenia drukarki, przyjmują wymagane przez instrukcje programu stany fizyczne. A zatem MACHINE PACK synchronizuje pracę urządzeń zewnętrznych i podporządkowuje ją aktualnie realizowanemu programowi. Pakiet ten zawiera ponadto procedury wprowadzania i uruchamiania programu użytkowego.

Jądrem systemu operacyjnego, umieszczonego przez producenta w pamięci stałej, jest **KERNEL**, który koordynuje realizację programów, zarządza wykorzystaniem pamięci i wyborem przyłączonych układów ROM, a także uczestniczy w obsłudze przerwań i zdarzeń. Istnieje jeszcze jeden element FIRMWAREu — **JUMPER**, tworzący w chwili włączenia komputera, w pamięci **RAM — JUMPBLOCK**, czyli blok instrukcji skokowych umieszczonych pod określonymi, stałymi adresami. Procedury, które się tam znajdują, zapewniają możliwość wywołania rozmaitych podprogramów FIRMWAREu, pomimo że związane z tym parametry (np. adresy lub wartości pewnych zmiennych systemowych) mogą się zmieniać w trakcie realizacji programu. Wykorzystanie pośrednika, jakim jest JUMPBLOCK, powoduje, że zmiany te są niezauważalne dla użytkownika. Elementy FIRMWAREu są dla niego dostępne, podobnie jak inne podprogramy wyższego poziomu. Zatem, aby tworzyć bardziej zaawansowane „dzieła", w których konieczne jest sterowanie HARDWAREm, nie musimy poznawać dogłębnie zasad działania poszczególnych urządzeń i sposobu ich kontrolowania przez system operacyjny. W wielu wypadkach wystarczy znajomość tego pakietu. Oczywiście, ponieważ JUMPBLOCK znajduje się w pamięci RAM, bardziej zaawansowany użytkownik może dokonywać w nim zmian i w konsekwencji np. zastąpić pewne firmowe procedury własnymi. W tym celu niekoniecznie zresztą trzeba się dobierać do tego pakietu, wystarczy dobra znajomość asemblera i organizacji pamięci. Tak, tylko wykorzystanie JUMPBLOCKa pozwoli nam w dużej mierze uniknąć nieświadomego bałaganu, jaki może powstać przy takich operacjach.

A zatem, jest sobie JUMPBLOCK, a nawet cztery bloki instrukcji skokowych. Pierwszy i największy, zwany głównym... ale o tym w następnej pogawędce.

(ROMan)

Od zabawy do nauki

Rozmowa z kpt. Waldemarem Pająkiem, kierownikiem Ośrodka Obliczeniowego w Wyższej Oficerskiej Szkole Samochodowej im. gen. Aleksandra Waszkiewicza w Pile.

— Oprócz funkcji zawodowych pełni Pan także funkcję opiekuna uczelnianego Klubu Mikrokomputerowego „Aneks"...

— Owszem, od początków jego powstania, drugi rok. To chyba jeden z pierwszych klubów mikrokomputerowych w kraju, a na pewno w uczelniach wojskowych.

— Pytanie chyba zasadne w tym miejscu: jakie macie komputery?

— Nie jest źle, chociaż mogłoby być lepiej i zanosi się na to, że będzie. Zaczynaliśmy od sprzętu wypożyczonego. Wprawdzie kiedyś Rada Młodzieżowa Wojska Polskiego obiecywała, że wyposaży nas w mikrokomputery, ale tej obietnicy do tej pory nie dotrzymała. „Aneks" ma obecnie 14 mikrokomputerów. Siedem Spectrum należy do klubu, natomiast tyle samo Amstradów, przeznaczonych głównie do celów dydaktycznych, wypożyczyła nam uczelnia. W tym roku mamy nadzieję zwiększyć nasz stan posiadania o kolejnych kilka Amstradów. Są one bardziej przydatne dla naszych celów.

— A właściwie, jakie są to potrzeby?

— Klub Mikrokomputerowy powstał przy Ośrodku Obliczeniowym i jego pracownicy są równocześnie nauczycielami i instruktorami. Właśnie oni przy pomocy podchorążych z Naukowego Koła Informatyki uczą języka, programowania i obsługi komputera. Przede wszystkim opieramy się na programach przydatnych do studiów i odpowiadających profilowi szkoły. Obejmują one przykładowo charakterystyki poszczególnych elementów i podzespołów samochodu. W ten sposób ucząc się czegoś nowego można lepiej zrozumieć rozpatrywane zagadnienia i znacznie wzbogacić wiedzę.

Nie koniec na tym. Pracujemy nad różnymi programami użytkowymi, które przyczynią się do poprawy administrowania i dowodzenia szkołą. Na przykład, kpr. pchor. SPR Wojciech Karcz, zwolennik komputeryzacji, buduje program pomocny przy typowaniu na praktyki słuchaczy tutejszej Szkoły Podchorążych Rezerwy. Ofertę klubu dopełnia działalność typowo relaksowa, a więc wszelkiego rodzaju gry komputerowe.

— Kto może być członkiem „Aneksu"?

— Liczba miejsc jest ograniczona bazą, którą dysponujemy, czyli ilością sprzętu i lokalem. W Ośrodku Obliczeniowym jest ciasno, ale wkrótce mamy dostać pomieszczenia w remontowanym Klubie Podchorążych. Regułą jest, że należy ten, kto chce, ale też prawdą jest, że kto pierwszy, ten lepszy. W ubiegłym roku „Aneks" miał 35 członków, wyłącznie podchorążych wyższej szkoły oficerskiej. Czasami z komputerów korzystają podchorążowie SPR. Prócz tego raz w tygodniu prowadzimy zajęcia dla dzieci kadry zawodowej uczelni. Cieszą się one dużym wzięciem, przychodzą i pociechy, i ich rodzice.

— Z jakich korzystacie źródeł, z jakiej literatury?

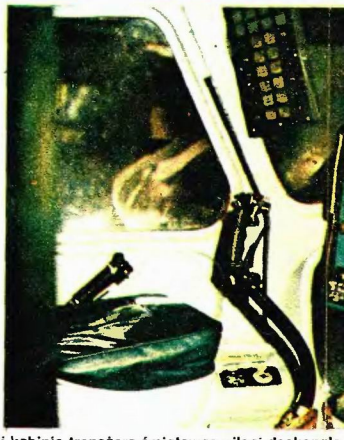
— Ze wszystkiego, co zdobędziemy. Mamy swoją biblioteczkę, do której członkowie i sympatycy „Aneksu" przynoszą każdą pozycję, jaką uda im się zdobyć. Prenumerujemy czasopisma fachowe: „Iksa", „Bajtka" i „Komputer". Przegrywamy ciekawsze programy.

— A plany?

— Rozwinąć działalność, zdobyć jeszcze lepszy sprzęt, zainteresować komputeryzacją kogo się da. No i pogłębiać swoje umiejętności, aby przyczyniły się one do opanowania tajników wiedzy wymaganej od specjalistów samochodziarzy.

Rozmawiał: **J. RAJCH**

To urządzenie pozwala na szkolenie nawigatorów w warunkach symulowanych sytuacji powietrznych.



W tej kabinie-trenażera śmigłowca piloci doskonalą w zdobywaniu umiejętności działania na wypadek r przewidzianych zdarzeń. Przy użyciu sprzętu bojowe byłoby to niemożliwe do osiągnięcia.

Starty i lądowania bez samolotów

Marek SIENIAWSKI

Na wskaźniku stacji radiolokacyjnej zaczyna pojawiać się coraz więcej celów powietrznych. Do odparcia ataku kierowane są siły obrony powietrznej kraju. Do akcji wkraczają samoloty bojowe, a więc piloci, kierownicy lotów, nawigatorzy. Uruchomiony zostaje cały system obrony. Alarm bojowy! Setki celów przeciwnika, a także własnych środków powietrznych znajdują się w akcji. W takiej sytuacji ludzie odpowiedzialni za kontrolę lotów jeszcze nigdy się nie znaleźli. Została ona tylko wymyślona. I byłoby najlepiej, gdyby do niej nie doszło. Gdyby jednak... Czy sobie poradzą?

Aby sprawdzić zachowanie się obsługi naziemnej w warunkach ekstremalnych, należałoby je realnie stworzyć. Innymi słowy w powietrzu umieścić odpowiednio oznakowane samoloty przeciwnika i własne. Taki sprawdzian jest jednak bardzo trudny do przeprowadzenia (nawet jednorazowo) i bardzo kosztowny. A jak zaplanować strącenia samolotów? Choćby z tych powodów imitacja jest trudna.

Każdy samolot, który znajduje się w powietrzu, **jest ciągle obserwowany i kierowany z ziemi.** Tak jest w lotnictwie cywilnym i wojskowym. Zajmijmy się wojskowym, wyposażonym w statki powietrzne, zdolne do działania w krótkim czasie na duże odległości, z dużą prędkością (ponad trzykrotnie większą od dźwięku — około 1 km na sekundę), uzbrojone w precyzyjne rakiety różnego typu. I jeszcze jedno. Lotnictwo wojskowe charakteryzuje się bardzo krótkimi czasami gotowości do masowego użycia. A to oznacza bardzo dużą odpowiedzialność obsługi naziemnej za podejmowane decyzje.

Jak szkolić ludzi do tej obsługi? Wygląda to mniej więcej tak, że po przeszkoleniu w uczelni wojskowej, nawigatorzy odbywają praktyki na stanowiskach dowodzenia. Okazuje się, iż najlepszą metodą nauczania jest praca indywidualna mistrza z uczniem. Trwa to dość długo, zanim „czeladnik" uzyska uprawnienia, zanim praktycznie będzie zdolny samodzielnie kierować lotem maszyny bojowej. Takie szkolenie, poza tym że jest bardzo pracochłonne, jest również kosztowne.

Obecnie technika komputerowa pozwala już na stworzenie urządzeń, w których można **imitować dowolne sytuacje powietrzne.** Ale sam pomysł nie wystarczy. Idea skonstruowania takiego imitatora była jednak na tyle kusząca, że zajęli się nią specjaliści z Instytutu Technicznego Wojsk Lotniczych.

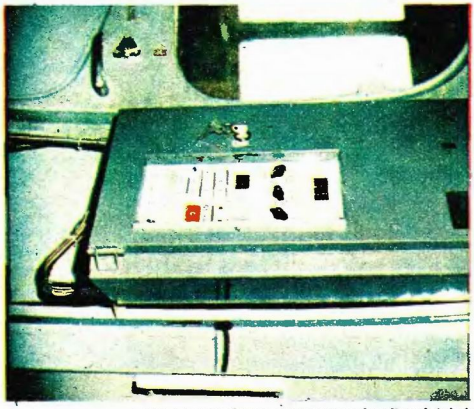
Jak bardzo to trudne zadanie? Przyjrzyjmy się pracy ludzi z obsługi naziemnej lotnictwa wojskowego. Wróćmy zatem do sytuacji skrajnej, gdy ogłoszono alarm bojowy. Złożoność jej widać na ekranach stacji radiolokacyjnych.

Kierownik lotów zarządza start samolotów dyżurnych. Gdy znajdują się one w powietrzu, kieruje nimi nawigator-operator. Prowadzi maszyny tylko w obrębie kilkudziesięciokilometrowego lotniska. Ma za zadanie **nie dopuścić do zagrożenia,** gdy wzrośnie liczba samolotów. Dla współczesnego myśliwca, tylko przy prędkości równej prędkości dźwięku, jedna minuta lotu oznacza przebycie około dwudziestu kilometrów.

Za strefą lotniska następuje przekazanie maszyny nawigatorowi naprowadzania. Z ziemi kieruje on „swój" samolot w rejon działania obcego celu. Kiedy pilot przechwyci cel — czyli widzi go na radarze pokładowym, wówczas samodzielnie wykonuje zadanie. Odpala rakiety. Znajduje się wtedy setki kilometrów od macierzystego lotniska.

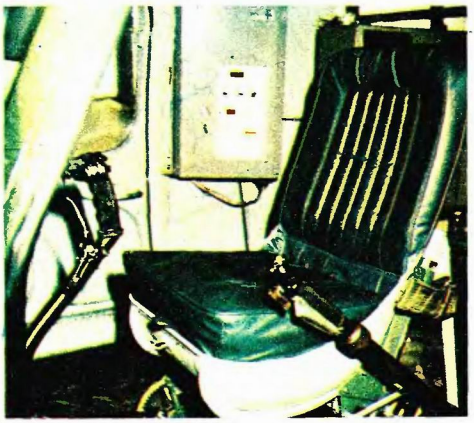
Wraca. I znowu zostaje po kolei przekazywany „z rąk do rąk" osobom funkcyjnym na ziemi. Sytuacja spiętrza się, gdy w tym samym czasie w rejonie lotniska znajduje się dużo samolotów. Trzeba określić, który z nich ma **lądować, i w jakiej kolejności.** Maszyny są więc poukładane w przestrzeni powietrznej jakby w szufladkach. Nie mogą się tam znajdować długo. Mają za mały zapas paliwa. A krążenie wokół lotniska jest kosztowne, niepotrzebne i ryzykowne.

Decyzję kierownika lotów o lądowaniu wykonuje kierownik systemu lądowania. Doprowadza samolot precyzyjnie do pasa startowego. Najtrudniejsze są **sytuacje awaryjne,** np. gdy samolot jest uszkodzony i musi lądować w pierwszej kolejności. Wtedy



Instruktor może zaprogramować najbardziej skomplikowaną sytuację, aby w ten sposób określić przydatność ludzi do kierowania ruchem samolotów.

ią się
k nie-
owego



Kabina trenażera nie różni się niczym od prawdziwego sprzętu latającego.

wszystkie inne trzeba szybko poszufladkować i wytyczyć pustą drogę temu jednemu. Tego manewru, choćby ze względu na bezpieczeństwo lotów, nie uda się nigdy przeprowadzić dla celów szkoleniowych. A wiadomo, że ludzie muszą być do tego przygotowani. Potrzebny jest więc odpowiedni trenażer.

Te i wiele innych barier udało się pokonać naukowcom z ITWL. Zespół kierowany przez płk. mgr. inż. Janusza Kucfira i płk. dr. inż. Henryka Gajewskiego opracował całą rodzinę imitatorów do kierowania lotami samolotów przy użyciu komputera.

Ale przecież komputer nie rozumie ludzkiej mowy, może ktoś zarzucić. Tymczasem trzeba zagwarantować naturalne warunki pracy kierownikowi lotów, nawigatorowi i pilotom. Funkcję pośrednika między ludźmi i komputerem zupełnie dobrze pełni operator lub operatorzy, gdy występuje więcej ćwiczących lub kontrolowanych.

Tak więc sposób kierowania samolotami przy użyciu imitatora IKS (zbieżność z tytułem miesięcznika jest przypadkowa), nie różni się niczym od kierowania lotami realnymi. Szczególną cechą urządzenia jest **modułowość w sensie technicznym i programowym.** Można szkolić specjalistów indywidualnie, w różnych grupach, włącznie z pilotami i dowolnie dobranymi zespołami. Maksymalnym wykorzystaniem wszystkich możliwości IKS-a (co już zrealizowano) jest prowadzenie walki powietrznej przez całą jednostkę lotniczą. Stwarzano różne skomplikowane sytuacje. W trakcie ćwiczenia ingerowano w jego przebieg, co świadczy o tym, że **imitator nie uczy schematów działania,** jest elastyczny w użyciu. Można zmienić liczbę celów powietrznych, imitować zestrzelenia, awarie, wypadki, katastrofy, a zatem sprawdzać zachowanie się wszystkich uczestników w skrajnie trudnych warunkach.

Obecnie więc bez startów, lotów i lądowań samolotów, bez narażania zdrowia ludzi, sprzętu na straty, a także wystawiania nerwów na poważne próby, wszystko może się odbywać w pomieszczeniach imitatora. Różnorodne sytuacje planuje dowódca lub instruktor. Włącznie z zakłóceniami radiowymi, atmosferycznymi, warunkami krytycznymi. Czyli takimi, jakie mogą realnie zaistnieć. Obsługa naziemna jest już do nich przygotowana.

Próby, w których uczestniczyli mistrzowie w specjalnościach kierowania lotami, ujawniły, że nie zawsze potrafili oni wyjść obronną ręką z trudnych zadań. Bo w takich warunkach dość często się nie znajdowali.

Jak wynika ze statystyk wypadków lotniczych, aż połowa z nich spowodowana jest winą pilotów i ludzi z obsług naziemnych, czyli innymi słowy wynika z niedoszkolenia. Już te fakty wskazują, że zastosowanie IKS-a uzupełnia tę lukę znakomicie.

Jak wielka jest to luka, świadczą też opinie specjalistów lotnictwa wojskowego w całym świecie. Twierdzą oni, że możliwości super nowoczesnych statków powietrznych w zestawieniu z wyszkoleniem obsług naziemnych, zgrania z pilotami, są niewykorzystane w dość dużym procencie.

Imitatory kierowania samolotami stwarzają więc ogromne **możliwości zgrywania systemów obrony powietrznej,** prowadzenia walki i wykorzystania do końca wszystkich walorów sprzętu bojowego oraz umiejętności ludzi.

W państwach socjalistycznych jest to pierwszy tego typu imitator o tak uniwersalnym zastosowaniu i dużych walorach użytkowych. Nic dziwnego, że zainteresowanie nim naszych sąsiadów i sojuszników jest bardzo duże. Ale nie tylko do celów lotnictwa wojskowego IKS może być dostosowany. Dzięki swojej modułowej budowie może równie dobrze wykonywać zadania w szkoleniu kontrolerów ruchu powietrznego na każdym lotnisku. Jeszcze jedną właściwością jest bowiem dostosowanie komputera do warunków każdego lotniska i obszaru dowolnego państwa.

Względy ekonomiczne (koszty paliwa, zużycia sprzętu, pracy wielu urządzeń lotniskowych), a także bezpieczeństwa lotów i tempa, w jakim można przygotować oraz kontrolować dokładność pracy obsługi naziemnej, zgrania z pilotami, są niewątpliwe.

Marek SIENIAWSKI

Sztuki i sztuczki (9)

Zmienne systemowe ROM-SPECTRUM

Wskaźniki stanu

FRAMES 23672..4
FLAGS 23611
FLAGS2 23658
TYFLAG 23612
FLAG-X 23665
Seed 23670/1
ULA (Uncommited Logic Array) jest modułem odpowiedzialnym między innymi za przekazanie pamięci ekranu z RAM na monitor. Od momentu włączenia komputer wysyła informacje w ustalonym formacie 256x192 punkty co 20 ms. Zrozumiałe, że ULA przerywa wtedy wszelkie operacje wykonywane przez mikroprocesor. Jednocześnie wymusza na nim wykonanie procedury o adresie 56 (RST 8), o której wspomniałem w odcinku 1. Anglicy nazywają to zjawisko ogłupieniem procesora. Oprócz przeszukiwania klawiatury dokonuje się wtedy zliczanie przerwań. Z bajtów FRAMES możemy dowiedzieć się, ile razy od momentu włączenia komputera ULA „ogłupiła" mikroprocesor.

Ponieważ dokładność taktowania zegara jest wystarczająco duża (błąd rzędu 0.01%) można założyć, że w jednej sekundzie przerwań będzie 50. Stąd prosta droga, aby liczba przerwań określała czas w sekundach od inicjalizacji systemu.

20 PRINT FN z ()/50; „s".

Nasz sztuczny „mechanizm zegarowy" jest następujący: po 255 przerwaniach (5. 1s.) zliczanych w 23672 następuje zerowanie tego bajtu i inkrementacja bajtu 23673. Po 65535 przerwaniach nastąpi wyzerowanie obu i inkrementacja kolejnego bajtu. „Zegar" wróci do pozycji wyjściowej po 2124-1 (czyli 16777215) przerwaniach. Chcąc wyrazić to w jednostkach czasu, otrzymamy 3 dni, 21 godzin, 12 minut, 24.32 sekundy. Mając taki argument w ręku, można praktycznie do każdego programu wprowadzić ten „czas". Należy jednak pamiętać, że niektóre procedury wyłączają przerwania i czas „stoi w miejscu". Są to wszystkie komendy współpracujące z urządzeniami zewnętrznymi, a więc **LOAD**, **SAVE**, **MERGE**, **VERIFY**, **COPY** i **BEEP**.

Wykorzystamy zegar do sprawdzenia czasu reakcji wzrokowej i manualnej:

```
10 DEF FN z()=PEEK 23672+256*PEEK 23673
+65536*PEEK 23674
20 PRINT FN z()
```

```
 10 DIM t(10)
 20 DEF FN z()=PEEK 23672+256*PEEK 23673
+65536*PEEK 23674
 30 LET b=1: LET s=0: CLS
 40 PRINT "program sprawdza twój refleks"
 50 PRINT "       s start"
 60 IF PEEK 23556<>83 THEN GOTO 60
 70 CLS: PRINT AT 10,0;
 80 FOR a=0 TO 9
 90 PRINT PAPER 4;a;PAPER 7;"  ";: NEXT a
100 RANDOMIZE INT (RND*65536)
110 LET r= INT (RND*10)
120 PRINT AT 1,0;"wciskaj odpowiedni klaw
isz numeryczny"
130 POKE 23674,0: POKE 23673,0: POKE 2367
2,0
140 PRINT AT 10,r*3: FLASH 1;r
150 IF INKEY$<>STR$ r AND FN z()<=5 THEN
GOTO 150
160 LET t(b)=FN z()/50
170 IF t(b)>5 THEN LET t(b)=5
180 LET b=b+1: IF b<11 THEN GOTO 70
190 FOR a=1 TO 10: LET s=s+t(a): NEXT a
200 LET b=s/10: CLS
210 PRINT "sredni czas reakcji """"    ";
b;" s."
220 REM histogram
230 DRAW 150,0
240 FOR a=1 TO 10: PLOT 10*a,0
250 FOR b=0 TO 30*t(a): PLOT 10*a,b
260 DRAW 5,2: NEXT b: NEXT a
270 PLOT 10,30*s: DRAW OVER 1;150,0
```

Flagi systemowe przechowują bardzo istotne informacje o stanie, w jakim znajduje się system operacyjny. Każda flaga zajmuje 1 bit, a więc może wynosić 0 lub 1. Poniżej przedstawione są znaczenia poszczególnych flag:

FLAGS	stan 0	stan 1
bit 0	drukuj spację przed znakiem	wydruk spacji zabroniony
bit 1	wyjście na ekran (kanał S)	wyjście na drukarkę (kanał P)
bit 2	drukuj kursor K	drukuj kursor L
bit 3	tryb K	tryb L
bit 5	nie wykryto klawisza	naciśnięto klawisz
bit 6	wynik ostatniej operacji jest łańcuchem tekstowym	wynik jest liczbą
bit 7	sprawdzanie syntaktyki linii	egzekucja linii

FLAGS 2	stan 0	stan 1
bit 0	czyszczenie ekranu nie jest konieczne	ekran główny jest czysty
bit 1	bufor drukarki jest pusty	bufor drukarki powinien być wyzerowany
bit 2	dwukropek rozdziela wyrażenia w linii	dwukropek jest znakiem w łańcuchu
bit 3	tryb L (low letters)	tryb C (capitals)
bit 4	aktualnym kanałem nie jest K	aktualnym kanałem jest K

TVFLAG	stan 0	stan 1
bit 0	wydruk na ekranie głównym	wydruk na ekranie edycyjnym
bit 3	tryb pracy nie może się zmienić (INPUT)	tryb pracy może się zmienić (edycja)
bit 4	zwykły listing (LIST)	listing automatyczny
bit 5	ekran edycyjny nie wymaga czyszczenia po naciśnięciu klawisza	wyczyścić ekran edycyjny po naciśnięciu klawisza

FLAG-X	stan 0	stan 1
bit 0	operacja wycinania znaków (slicing)	operacja na całym stringu
bit 1	zmienna już istnieje	nowa zmienna
bit 5	tryb edycji (EDIT)	tryb wprowadzania (Input)
bit 6	INPUT tekstowy	INPUT NUMERYCZNY
bit 7	INPUT bez frazy LINE	INPUT LINE

Zastosowanie flag w BASIC'u jest niewielkie i w zasadzie nie spotyka się ich w programach. Najczęściej wykorzystuje się zmianę trybu wydruku liter na ekranie.

POKE 23658,8

ustawia tryb C (duże litery), a

POKE 23658,0

powoduje wydruk liter małych w trybie L.

Zatrzymanie egzekucji programu w dowolnym miejscu z raportem OK odbędzie się po instrukcji:

POKE 23611, PEEK 23611-128.

Częściej używa się flag systemowych przy programowaniu w języku wewnętrznym, lecz należy to robić ze szczególną ostrożnością.

Na koniec zmienna **SEED**, która znalazła się w tej grupie dodatkowo. Jeśli używamy funkcji **RND**, która generuje liczby pseudolosowe, to jest ona obliczana właśnie na podstawie zawartości **SEED**. Wniosek, że umieszczenie tam jakiejś liczby z zakresu 1..65535 spowoduje narzucenie zawsze stałego ciągu wartości funkcji **RND**, jest więc prawidłowy. Można taki proces zainicjować instrukcją **RANDOMIZE n**, której jedynym efektem jest przeniesienie **n** do **SEED**. Stąd sekwencja rozkazów:

```
10 RANDOMIZE 70
20 PRINT INT (RND*6)+1
30 GOTO 10
```

spowoduje zawsze wygenerowanie liczby 1.

Zmienną SEED wykorzystuje się często do konwersji liczby 16-bitowej na postać dwubajtową np.:

```
10 INPUT n
20 IF n<> INT n THEN GOTO 10
30 IF n<0 OR n>65535 THEN GOTO 10
40 RANDOMIZE n
50 PRINT n;"(";PEEK 23670;",";PEEK
   23671;")"
```

Ostatnia część cyklu będzie poświęcona kalkulatorowi i arytmetyce zmiennoprzecinkowej w Spectrum.

Krzysztof MAMCARZ

Komputer w medycynie

Tomograf jądrowego rezonansu magnetycznego

W ostatnim dziesięcioleciu diagnostyka medyczna wzbogaciła się o liczną rodzinę urządzeń zwanych tomografami, umożliwiającymi za pomocą różnych metod fizyki, bezinwazyjne — a więc nie wymagające użycia skalpela, badanie dowolnego narządu wewnętrznego pacjenta. Poznaliśmy już bliżej tomograf komputerowy, w którym czynnikiem penetrującym organizm pacjenta jest promieniowanie Roentgena. Charakter tego promieniowania niesie za sobą pewne ograniczenia w wykorzystaniu tomografu komputerowego ze względu na szkodliwość częstych prześwietleń. Urządzenie to doskonale nadaje się do obrazowania na monitorze telewizyjnym trójwymiarowych kształtów badanego narządu, głównie części kostnych, ale nie dostarcza wielu informacji o wewnętrznej budowie oglądanych struktur. Tomografia komputerowa jako pierwsza i względnie tania metoda diagnostyki medycznej, wykorzystująca elementy wiedzy medycznej, fizyki i informatyki, jest jednak powoli wypierana przez inną, bardziej precyzyjną metodę tomografii jądrowego rezonansu magnetycznego.

Nowa metoda obrazowania wewnętrznych struktur organizmu na monitorze telewizyjnym wkroczyła przebojem na scenę medyczną, dokonując przełomu w technice wykrywania chorób nowotworowych. Tomograf jądrowego rezonansu magnetycznego, podobnie jak rentgenowski tomograf komputerowy, pozwala obserwować zmiany nowotworowe w organiźmie, czyli narastanie guzów niszczących tkanki i narządy; jest jednak jedynym tego rodzaju urządzeniem wskazującym również miejsca zagrożone możliwością powstania raka.

Aby zrozumieć zasadę działania tego rewelacyjnego urządzenia, musimy przypomnieć sobie kilka znanych z lekcji fizyki wiadomości o budowie atomu. Atom składa się z jądra atomowego, zbudowanego z protonów i neutronów oraz z elektronów krążących w dość dużej odległości wokół jądra. W interesującym nas zjawisku ograniczmy się do pamiętania o jądrach atomowych, które możemy wyobrazić sobie jako bączki wirujące beztrosko w przestrzeni. Okazuje się, że w przypadku atomów wodoru, umieszczonych w polu działania dostatecznie dużego magnesu, jądra atomowe tych pierwiastków, a więc nasze bączki, zaczynają ustawiać swoje osie obrotu równolegle do kierunku linii pola magnetycznego. Za pomocą dodatkowego, pulsującego pola elektromagnetycznego możemy tak rozhuśtać wirujące bączki, że zmieni się ich prędkość obrotowa i położenie osi obrotu. O jądrach znajdujących się w takim właśnie, wzbudzonym stanie mówimy, że zostały wprowadzone w rezonans, a całe zjawisko nazywa się jądrowym rezonansem magnetycznym. Po ustaniu pulsującego pola magnetycznego jądra atomów wracają do poprzedniego stanu, zwracając przyjętą energię w postaci fal radiowych — czyli wzbudzone jądra atomowe głośno sygnalizują powrót do normalności. Czas trwania tego zjawiska, zwany czasem relaksacji jądrowego rezonansu magnetycznego, wskazuje czy narząd, którego atomy wodoru wprowadziliśmy w stan rezonansu magnetycznego, jest zdrowy, czy odbiega od normy na skutek zmian nowotworowych. Czas relaksacji u osób chorych lub zagrożonych nowotworem jest dłuższy niż u ludzi zdrowych. Tyle należałoby powiedzieć, aby zrozumieć teoretyczną zasadę pracy tomografu jądrowego rezonansu magnetycznego, możemy zatem przejść do samego urządzenia.

Jest bardzo kosztowne. W jego skład wchodzi potężny elektromagnes, generator pulsującego pola elektromagnetycznego oraz komputer, sterujący urządzeniem i przechowujący zbierane parametry fizyczne badanego narządu. Pacjent umieszczany jest w specjalnej kapsule, wewnątrz której wytwarzane jest pole magnetyczne 60 000 razy przekraczające ziemskie pola magnetyczne. Pole to porządkuje jądra atomowe wodoru, ustawiając ich osie obrotu w kierunku linii pola magnetycznego. Następnie do pracy wkracza generator pulsującego pola elektromagnetycznego, wprowadzając kolejno w rezonans magnetyczny jądra atomowe, leżące wzdłuż osi ustalonego przez komputer przestrzennego układu współrzędnych. Dla stworzenia grafiki komputerowej wybranego narządu wewnętrznego obszar zawierający ten narząd podzielony jest w pamię-

****

NOWOŚCI ********** M O C D L A B I U R ********** NOWOŚCI

****

OFFICEPOWER - uznany został za najlepszy na świecie system komputerowy dla biur, którego autorem jest brytyjska ICL - International Computers Limited. System został stworzony dla amerykańskiego Ministerstwa Transportu w wyniku kilkuletniej pracy grupy fachowców z ICL. System - po dwóch latach eksploatacji - okazał się bardzo dobry. Obecnie sprzedaje się go masowo dla dużych biur na całym świecie. OFFICEPOWER jest główną częścią ICL-owskiej oferty w zakresie systemów biurowych (Office Systems), których sprzedaż w 1991 roku szacuje się na 10 miliardów dolarów rocznie.

OFFICEPOWER jest kompletnym, zintegrowanym systemem służącym do automatyzacji prac biurowych. W przypadku pracy lokalnej obsługuje od 4 do 128 użytkowników (lub więcej w przypadku sieci). Software wykorzystuje system operacyjny UNIX V stający się międzynarodowym standardem, natomiast hardware stanowią 32-bitowe komputery ICL-owskie CLAN3 i CLAN7 (od jesieni tego roku również DRS300). System wykorzystuje również jako terminale ICL-owskie: DRS300, PC QUATTRO i OPD, IBM-owskie PC (i kompatibilne, w szczególności ITT-XTRA). System komunikuje się ze wszystkimi dużymi komputerami ICL i IBM. Główną jego zaletą jest bardzo wysoka jakość wszystkich jego aplikacji.

Na OFFICEPOWER składają się:

* dwupoziomowe przetwarzanie tekstów z wszystkimi znanymi obecnie możliwościami. Przetwarzanie dwupoziomowe oznacza, że pierwszy poziom przeznaczony jest dla osób, które usiadły po raz pierwszy do komputera, natomiast poziom drugi dla bardziej doświadczonych,
* poczta elektroniczna o nieograniczonej ilości "skrzynek pocztowych",
* notes i kalendarz na dowolną ilość lat,
* "przypominacz" pamiętający terminy zebrań i spotkań,
* listy nazwisk i adresów układanych wg. dowolnych kluczy,
* automatyczna adresacja pism i listów do wielu adresatów wyznaczonych z listy,
* arkusz elektroniczny zintegrowany z edytorem tekstów i grafiką wykresową,
* baza danych, raportowanie i kalkulator wielozadaniowy,

* informator telefoniczny zawiadamiający, że podczas naszej nieobecności ktoś, kto odebrał telefon ma dla nas wiadomość (lub wręcz ta wiadomość),
* możliwość zakładania folderów zawierających zbiory uszeregowanych plików,
* możliwość przesyłania informacji drogą telekomunikacyjną oraz zabezpieczania ich według własnych kluczy,
* możliwość definiowania własnych aplikacji dla tych, których pociąga eksperymentowanie z systemem.

OFFICEPOWER okazał się systemem na tyle "przyjaznym" w użyciu że, użytkujący go szybko pozbyli się barier psychologicznych związanych z zastosowaniem komputerów w codziennej pracy biurowej.

****

(sch)

ci komputera na cegiełki porównywalne rozmiarami z cząsteczkami chemicznymi. W pamięci komputera zarejestrowane zostają czasy relaksacji jąder atomów każdej cegiełki budującej badany obszar wraz z informacją o liczbie protonów i właściwościach magnetycznych każdej z nich. Dane te podlegają obróbce komputerowej, a jej rezultatem jest kolorowy, przestrzenny obraz wideo badanego narządu. Próbkę możliwości tej metody diagnostycznej daje zamieszczone powyżej zdjęcie przedstawiające grafikę komputerową głowy zdrowego człowieka. Na zdjęciu tym widoczne są wszystkie miękkie części głowy, takie jak mózg czy jama ustna. O ich kolorze i stopniu jego jaskrawości decyduje budowa i magnetyczne właściwości obrazowanych elementów anatomicznych.

Tomograf jądrowego rezonansu magnetycznego, dzięki swojej nieszkodliwej dla zdrowia pacjenta metodzie działania, staje się w rękach lekarzy narzędziem niezwykle pożytecznym i mimo swojej ceny, stosowanym w świecie na coraz większą skalę. Długą listę jego możliwości otwiera sztandarowy przykład diagnostyki chorób nowotworowych, nie należy jednak zapominać, że lista ta jest dużo dłuższa. Rozpoznawanie stanów przedzawałowych, alergii czy wczesnych stadiów cukrzycy — to tylko kilka z jego dalszych zastosowań. Dlatego też dobrze się stało, że kilka naszych rodzimych ośrodków medycznych podjęło intensywne starania, mające na celu zakup i zainstalowanie w Polsce pierwszych egzemplarzy tego niezwykle potrzebnego urządzenia.

(A.T.)

Elementy systemów mikrokomputerowych (5)

Intel 8080

Jacek WOJTALA

Mikroprocesor INTEL 8080 jest przedstawicielem linii rozwojowej, której kolejnymi konstrukcjami są mikroprocesory: firmy INTEL 8008, 8080, 8085; firmy ZILOG Z80 i Z8000 oraz rodzina firmy INTEL i APX 86, 88 czyli 8086, 8088, 80188, 80186, 80286, 80386.

INTEL 8080 jest układem wykonanym w technologii NMOS. Technologia ta umożliwiła zwiększenie liczby elementów umieszczonych na półprzewodnikowej strukturze układu oraz zdecydowany wzrost szybkości działania mikroprocesora, w stosunku do układów wykonanych we wcześniej wykorzystywanej technologii PMOS (np. INTEL 8008). Mikroprocesor, wprowadzony na rynek w 1974 roku należy do najpopularniejszych mikroprocesorów 8-bitowych.

ARCHITEKTURA MIKROPROCESORA

Mikroprocesor 8080, przedstawiony na rys. 10, zawiera m.in. jednostkę arytmetyczno-logiczną ALU, zespół rejestrów z układami adresowania, 8-bitowy rejestr rozkazów IP oraz wewnętrzną magistralę danych.

8-bitowa **ALU** przeznaczona jest do wykonywania operacji arytmetycznych, logicznych, przesunięć. W jej skład wchodzą: blok do prowadzenia operacji arytmetycznych i logicznych oraz rejestry — 8-bitowy **akumulator A,** 8-bitowy **rejestr buforowy,** 8-bitowy **rejestr pomocniczy (przejściowy),** 5-bitowy **rejestr wskaźników F.**

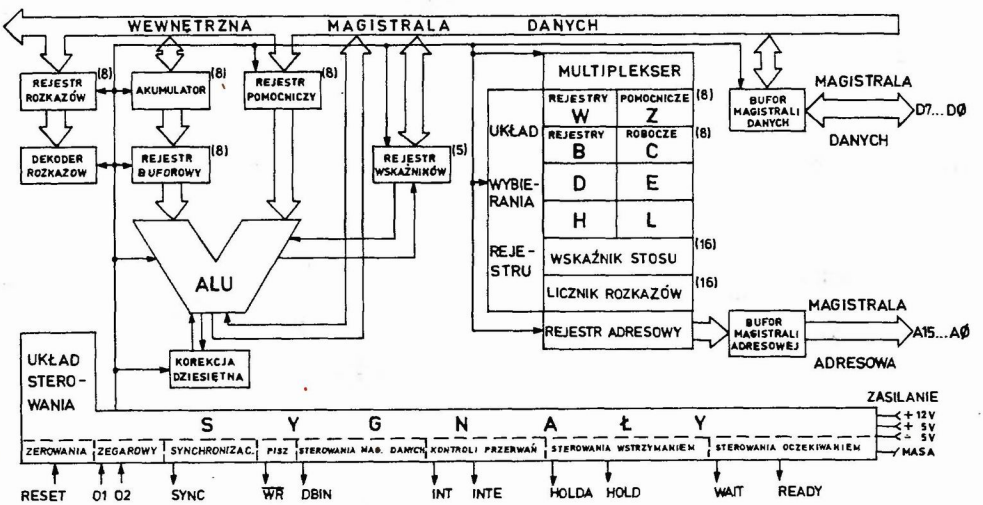
Zespół rejestrów z układami adresowania składa się z: 16-bitowego **licznika rozkazów PC;** 16-bitowego **wskaźnika stosu SP;** sześciu 8-bitowych **rejestrów roboczych,** oznaczonych literami B, C, D, E, H, L, tworzących trzy pary rejestrów oraz z dwóch rejestrów W, Z, niedostępnych programowo. Para rejestrów może tworzyć 16-bitowy **licznik danych DC,** adresujący komórki pamięci w przestrzeni adresowej 0,... 65535 (2^{16}).

DZIAŁANIE MIKROPROCESORA

Każdy cykl rozkazów mikroprocesora 8080, czyli czas konieczny na pobranie i wykonanie rozkazu, zawiera od 1 do 5 cykli maszynowych. Cykl maszynowy trwa od 3 do 5 taktów zegarowych. Pobranie rozkazu oznacza przesłanie go z pamięci do rejestru rozkazów. Wykonanie rozkazu oznacza zdekodowanie go i przetłumaczenie na odpowiednie mikrorozkazy.

W zależności od tzw. **słowa statusu** (status word) w mikroprocesorze 8080 wyróżnia się 10 następujących cykli maszynowych: pobranie rozkazu, odczyt z pamięci, zapis w pamięci, odczyt ze stosu, zapis na stos, odczyt z układu wejścia-wyjścia, zapis w układzie wejścia-wyjścia, potwierdzenie przerwania, potwierdzenie rozkazu HLT, potwierdzenie przerwania podczas stanu HALT.

Uproszczoną zasadę działania mikroprocesora 8080 można przedstawić następująco. Wykonanie każdego z rozkazów rozpoczyna się od pobrania kodu rozkazu i przekazania go do rejestru rozkazów, co następuje w pierwszym cyklu maszynowym.



Intel 8080

Licznik rozkazów zwiększany jest o 1, w układzie sterowania dekodowany jest kod rozkazu i mikroprocesor decyduje, czy konieczne jest pobranie następnych bajtów. Jeśli nie jest konieczne pobieranie następnych bajtów, wówczas rozpoczyna się cykl, w którym następuje wykonanie rozkazu.

ZESPÓŁ JEDNOSTKI CENTRALNEJ

W skład systemu 8080 wchodzą: mikroprocesor 8080, **generator sygnałów zegarowych** 8224 i **kontroler systemu** 8228.

Układ 8224 przeznaczony jest do generowania: sygnałów zegarowych, sterujących jednostką centralną; i sygnału sterującego kontrolerem systemu oraz do synchronizacji pracy jednostki centralnej.

Układ 8228 pełni funkcję kontrolera systemu i dwukierunkowego bufora do magistrali danych. Wytwarza sygnały kontrolne niezbędne do bezpośredniej współpracy jednostki centralnej z pamięciami i układami obsługującymi urządzenia wejścia-wyjścia. Zespół jednostki centralnej przedstawiono na rys. 11.

UKŁADY WSPÓŁPRACUJĄCE

System 8080 wymaga stosowania układów pomocniczych, poniżej przedstawiono niektóre z nich.

8205 dekoder binarny 1 z 8. Wykorzystywany jest do dekodowania struktur pamięci i układów wejścia-wyjścia. Jedno z ośmiu wyjść przechodzi w stan zera logicznego, gdy zostaną odpowiednio ustawione 3 wejścia zezwalające.

8212 8-bitowy rejestr buforowy. Wykorzystywany jest do wprowadzania i wyprowadzania informacji z systemu oraz do zsynchronizowania przepływu informacji wewnątrz systemu.

8214 kontroler priorytetu przerwań. Wykorzystywany do przejmowania maksymalnie 8 sygnałów żądania przerwania i wybrania sygnału o najwyższym priorytecie.

8216 4-bitowy dwukierunkowy wzmacniacz. Wykorzystywany w celu zapewnienia właściwej współpracy układów MOS i TTL, tzn. do wzmacniania sygnałów przesyłanych magistralą danych i magistralą adresową.

8251 programowany szeregowy układ wejścia-wyjścia (USART). Wykorzystywany do transmisji szeregowej synchronicznej i asynchronicznej między systemem a innymi urządzeniami.

8255 programowany równoległy układ wejścia-wyjścia. Wykorzystany może być jako dwa 8-bitowe porty wejścia-wyjścia A i B, 8-bitowy port wejścia-wyjścia C w dwóch 4-bitowych częściach i 8-bitowy dwukierunkowy bufor magistrali danych.

8253 układ programowanych liczników. Zawiera trzy niezależne układy czasowe, każdy z liczników programowany jest indywidualnie.

8257 układ bezpośredniego dostępu do pamięci DMA. Jest specjalizowanym procesorem, sterującym bezpośrednim przesyłaniem danych między pamięcią a urządzeniami zewnętrznymi, z pominięciem jednostki centralnej.

8259 programowany układ sterowania przerwań. Wykorzystywany do sterowania 8 poziomami przerwań. Umożliwia rozbudowę systemu przerwań do 64 poziomów oraz ma wiele dodatkowych możliwości w stosunku do 8214.

8271 kontroler stacji dysków elastycznych

8275 kontroler monitora ekranowego

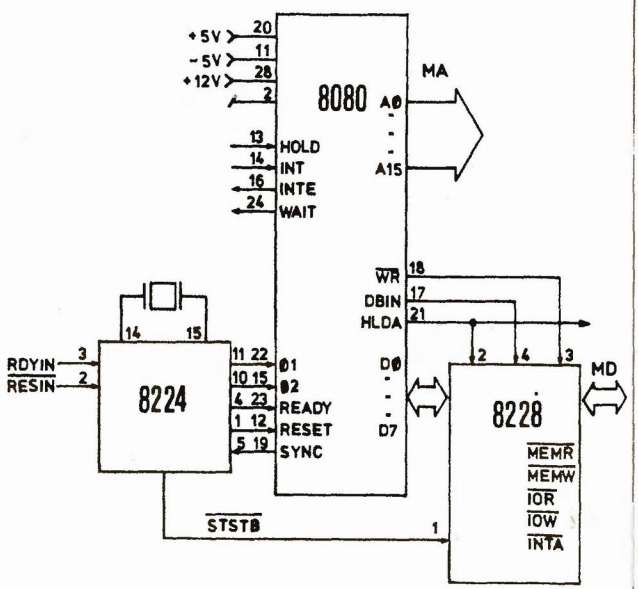
8279 programowany układ współpracy z klawiaturą oraz wskaźnikami cyfrowymi.

W systemie mogą być wykorzystywane m.in. następujące typy pamięci RAM:

— statyczne:
  2101 o pojemności 256×4 bity;
  2102 o pojemności 1024×4 bity;
— dynamiczne:
  2107 o pojemności 4096×1 bit;
  2116 o pojemności 16384×1 bit.

Jako pamięci stałe wykorzystuje się najczęściej:

— pamięci PROM:
  3601 o pojemności 256×4 bity;
  3604 o pojemności 512×8 bitów;
— pamięci EPROM:
  2708 o pojemności 1024×8 bitów;
  2716 o pojemności 2048×8 bitów.



PRZERWANIA

W wielu systemach mikroprocesorowych musi istnieć możliwość zawieszenia wykonywania programu i zareagowania mikroprocesora na sygnał przychodzący z zewnątrz. Mikroprocesor 8080 umożliwia włączenie lub wyłączenie **układu przerwań** (systemu przerwań). Po włączeniu tego układu do kontrolera 8228, mikroprocesor, kończąc każdy cykl rozkazowy, bada stan linii zgłoszenia przerwania. Jeśli przerwanie nastąpiło, mikroprocesor zawiesza wykonywanie programu i wykonywany jest skok do podprogramu, tzn. programu obsługi przerwania. Po zakończeniu obsługi przerwania następuje powrót do programu.

LISTA ROZKAZÓW

Mikroprocesor 8080 charakteryzuje się listą 78 rozkazów, które można podzielić na pięć grup:

— **rozkazy przesyłania informacji;**
— **rozkazy arytmetyczne;**
— **rozkazy logiczne;**
— **rozkazy skoków, wywoływania podprogramów;**
— **rozkazy sterujące, obsługi wejścia-wyjścia, stosu.**

Lista rozkazów mikroprocesora rozpocznie następny odcinek cyklu.

Jacek WOJTALA

SHARP

JANUSZ MILLER
SHARP 1500-PC
TAJNY ZAPIS

```
 10:"X":RANDOM :
    CLEAR :CLS :
    USING :WAIT 0
 20:INPUT "ILE LIT
    ER SZYFRU N=";
    N
 30:IF N<1OR N>16
    THEN BEEP 10,2
    5,50:PRINT "1<
    N<16":GOTO 20
 40:FOR K=1TO N
 50:A=(RND 26)+64:
    P$=P$+CHR$ A
 60:NEXT K:M=N:Y$=
    P$:Y=ASC Y$:A$
    ="OD A DO Z":
    PRINT A$;
 70:IF Q>10GOTO 18
    0

 80:INPUT ".TYPUJE
    :";X$
 90:CLS :X=ASC X$
100:IF X<YLET D$="
    ZA LITERA "
110:IF X>YLET D$="
    PRZED LITERA "
120:IF X=YLET L=L+
    1:GOTO 140
130:PRINT D$;X$;:Q
    =Q+1:GOTO 70
140:PRINT "ZGADLES
    ";L;" LITERE":
    BEEP 3,100,500
150:Z$=Z$+X$:M=M-1
    :IF Z$=P$GOTO
    170
160:Y$=RIGHT$ (Y$,
    M):Y=ASC Y$:Q=
    0:PRINT A$;:
    GOTO 80
170:WAIT :PRINT "S
     Z Y F R =";Z$
    :GOTO 190
180:CLS :WAIT 0:
    PRINT "A . L  A
      R  M":BEEP 7
    ,65,1000
190:INPUT "OD NOWA
     ? T=1 N=0 ";D
200:IF D=1GOTO 10
210:END
```

Tajny zapis

Janusz MILLER

Program wywołuje etykieta „X" (RU
'0). Podajemy *liczbę liter* tajnego zap
u.

W tym momencie rusza generat
czb losowych, który sam — bez n
zej wiedzy i woli — kolejno typuje ty
ter, ile podaliśmy.

Tak powstaje *tajny zapis*, który z r
uły nie ma żadnej sensownej treści
rzypomina kryptogram szyfru.

Tajny zapis odczytujemy następują
o:

— najpierw typujemy literę z zakr
u od A do Z,

— przypuśćmy, że wytypowaliśmy li
rę „M",

— teraz komputer udziela nam
wskazówki, iż tajna litera jest *„przed literą M" lub „za literą M"*,

— tak kolejno zawężamy pole po
szukiwań.

Gdy (po którymś pytaniu-wskazówce) odgadniemy, to wtedy pojawia się meldunek *„zgadłeś 1 literę"*.

W ten sposób, kolejno typując, odgadujemy wszystkie litery tajnego dokumentu, co potwierdza meldunek: *SZYFR =*" i tu jest wymieniony ciąg *tajnego zapisu*".

Nasz program pozwala na zadanie
0 pytań, aby odgadnąć jedną literę.

Uwaga: nie znamy „metody na szyfr". roponujemy zaś *metodę podziału*. Oto na:

— całość (od A do Z) podziel na powę i wytypuj literę krańcową podziału,

— wskazaną przez komputer „powę" znów podziel na połowę...

Teoria mówi, iż *wystarczy tylko 5 pyń*...

Czy to prawda?

ZX SPECTRUM

```
  9 REM tajny zapis
 10 LET q=1: DIM p$(16)
 12 DIM z$(16): DIM y$(16)
 15 LET l=0
 20 INPUT "podaj liczbe liter szyfru n ="$n
 30 IF n(1 OR n)16 THEN  GO TO 20
 40 FOR k=1 TO n
 42 LET a=INT (RND*26)+64
 44 LET p$(k)=CHR$ a
 46 NEXT k
 60 LET y$(1 TO 16)=p$(1 TO 16)
 61 LET k=1
 62 LET y=CODE y$(k)
 64 LET a$=" od A do Z": PRINT a$

 70 IF q)10 THEN  GO TO 180
 80 INPUT "Typuje "$x$
 90 LET x=CODE x$
100 IF x(y THEN  LET d$="za litera "
102 FOR n=1 TO 5: BEEP .1,7-n: NEXT n
110 IF x)y THEN  LET d$="przed litera "
112 FOR n=1 TO 5: BEEP .1,n: NEXT n
120 IF x=y THEN  LET l=l+1: GO TO 140
130 PRINT d$$x$: LET q=q+1: GO TO 70
140 PRINT "zgadles "$l$" litere": LET k=k+1
141 FOR n=1 TO 7: BEEP .2,n: NEXT n
150 LET z$(k-1)=x$
155 IF z$(1 TO 16)=p$(1 TO 16) THEN  GO TO 170
160 LET y=CODE y$(k): LET q=0
161 PRINT a$: GO TO 80
170 PRINT " szyfr = "$z$(1 TO 16): GO TO 190
180 PRINT " alarm "
185 BEEP 2,1: BEEP 2,2
190 INPUT "od nowa (tak/nie) "$w$
192 IF w$="tak" THEN  CLS : GO TO 10
210 STOP
```

Program kopiujący

Program służy do kopiowania gier i innych programów o pojemności do 36 kB, nagranych na kasecie.

Po uruchomieniu (RUN) naciskamy O (odczyt). Gdy usłyszymy pojedynczy sygnał dźwiękowy, przygotowujemy kasetę w magnetofonie oraz wciskamy RETURN. Jeśli kopiowany program składa się z kilku części, można wgrywać je kolejno, ale należy uważać, by nie przekroczyć zawartości wolnej pamięci. Po wgraniu programu otrzymuje się na ekranie wartość: pojemność kopiowanego programu oraz pozostałość wolnej pamięci RAM. Chcąc zapisać program na innej kasecie, należy przygotować magnetofon do nagrywania, nacisnąć Z (zapis), a następnie numer kopiowanej części.

Gdy wystąpi błąd w czasie wprowadzania, program zostaje zatrzymany, a na ekranie otrzymuje się potwierdzenie o wystąpieniu błędu.

Stanisław JANKOWSKI

```
ZI 100 DIM C$(7),D(9):GRAPHICS 0:PO
   KE 709,14:POKE 710,0:POKE 752,1
UD 110 C$(1,1)=CHR$(104):C$(2,2)=CH
   R$(104):C$(3,3)=CHR$(104):C$(4,4
   )=CHR$(170):C$(5,5)=CHR$(76)
GE 111 C$(6,6)=CHR$(86):C$(7,7)=CHR
   $(228):C=ADR(C$):D(0)=3994
EU 120 AD=868:CO=866:LE=872:D=39994
   :M=0:FOR I=1 TO 9:D(I)=0:NEXT I
ZQ 130 POSITION 1,0:? " *** Kopiow
   anie programow i gier *** ":POS
   ITION 21,23:? "wolne      bajto
   w";
LU 140 POSITION 27,23:? D-D(M);:OPE
   N #1,4,0,"K"
HF 160 GET #1,L
MP 170 IF L=79 THEN FLAG=0:M=M+1:PO
   SITION 2,M*2+2:? M:POSITION 5,M*
   2+2:? ": ... ";:POSITION 17,2:?
   "Odczyt":GOTO 200
NE 180 IF L=90 THEN 500
AU 181 IF L=77 THEN POKE 54018,52
LZ 182 IF L=32 THEN POKE 54018,0
HE 184 IF L=75 THEN RUN
OM 190 GOTO 160
IQ 200 OPEN #2,4+FLAG,128,"C:":POSI
   TION 7+2*FLAG,2+2*M:? "  "
AQ 220 L=D(M-1)+1:H=INT(L/256):L=L-
   H*256:POKE AD,L:POKE AD+1,H
JE 230 L=(FLAG=0)*D+(FLAG=4)*D(M)-D
   (M-1):H=INT(L/256):L=L-H*256:POK
   E LE,L:POKE LE+1,H
VM 240 POKE CO,7+FLAG
UL 250 L=USR(C,32):E=PEEK(867)
LW 260 CLOSE #2
ES 270 IF FLAG=4 THEN POSITION 17,2
   *M+2:? " ":M=N:POSITION 17,2:? "
      ":GOTO 160
KZ 280 IF E<>136 THEN POSITION 2,2+
   2*M:? "          ":POSITION 17,2:
   ? "Pomylka !":M=M-1:GOTO 160
JK 290 D(M)=D(M-1)+PEEK(LE)+256*PEE
   K(LE+1)
ZX 300 POSITION 7,2+2*M:? D(M)-D(M-
   1):POSITION 27,23:? "      ";CHR
   $(30);CHR$(30);CHR$(30);CHR$(30)
   ;CHR$(30);
EC 301 ? CHR$(30);D-D(M);:GOTO 160
EN 500 POSITION 17,2:? "Zapis "
HU 505 GET #1,L:IF L<49 OR L>48+M T
   HEN POSITION 17,2:? "      ":GOT
   O 160
PC 510 N=M:M=L-48:FLAG=4:POSITION 1
   7,2+M*2:? "*":GOTO 200
```

Test stacji dysków

Program DRIVETEST umożliwia sprawdzenie prędkości obrotowej stacji dysków ATARI 1050 oraz sygnalizuje ewentualne odchylenie tej prędkości od wartości uznawanej za poprawną. Pomiar dokonywany jest przy wykorzystaniu procedury w kodzie maszynowym (linie DATA). Procedura ta po wywołaniu z programu w języku BASIC zeruje na początku dwa najmniej znaczące bajty zegara systemowego (bajty o adresie $13 i $14), a następnie odczytuje z dyskietki sto razy sektor o numerze jeden. W czasie trwania odczytu zegar systemowy jest zwiększany o jeden co 1/50 sekundy. Po zakończeniu odczytu zawartość bajtów $13 i $14 (jest to czas trwania odczytu stu sektorów o tym samym numerze, inaczej stu obrotów, wyrażonych liczbą impul-

sów generowanych co 1/50 sekundy) przekazywana jest do programu wywołującego procedurę, gdzie następuje obliczenie liczby obrotów stacji na minutę. Jeżeli obliczona liczba obrotów mieści się w przedziale 284<LO<292, to wyświetlany jest komunikat „DRIVE OK". Jeżeli nie, to w zależności od obliczonej wartości program informuje o „obrotach zbyt wolnych" lub „zbyt szybkich".

H. KRASUSKI

```
WO 1 REM ************************
   ****
HG 2 REM DRIVE TEST
DU 3 REM (C) 1987 H. KRASUSKI
WR 4 REM ************************
   ****
PC 10 DIM SZYB$(66)
DX 15 GOSUB 200
XM 20 ? CHR$(125)
DK 30 ? "TEST SZYBKOSCI OBROTOW STA
   CJI"
HB 40 ? "          DYSKOW"
QL 41 ? "   (C) 1987 H.KRASUSKI"
FU 50 ? :? :?
QX 60 ? "Podaj numer drive'u (1-2)
   ";:INPUT ND
YQ 70 ? "Prosze chwile poczekac (ok
   . 1 min)..."
XF 100 X=USR(ADR(SZYB$),ND)
BC 110 SZYB=X/3600:SZYB=INT((100/SZ
   YB+0.5)*5/6)
CH 120 ? "Drive ";ND;"  ";SZYB;"  o
   br./min"
BN 130 IF SZYB<292 AND SZYB>284 THE
   N ? "drive OK"
AC 140 IF SZYB<=284 THEN ? "OBROTY
   ZBYT WOLNE"
BZ 150 IF SZYB>=292 THEN ? "OBROTY
   ZBYT SZYBKIE"
OC 160 END
HP 200 RESTORE 250:FOR I=1 TO 66
AU 210 READ A:SZYB$(I)=CHR$(A)
FW 220 NEXT I
ZE 230 RETURN
JM 250 DATA 104,104,104,141,1,3,169
   ,1
WP 251 DATA 141,10,3,169,0,141,11,3
   ,169,0
ZW 252 DATA 141,4,3,169,4,141,5,3,1
   69,82
UU 253 DATA 141,2,3,169,5,32,83,228
   ,198
ZZ 254 DATA 203,208,249,169,100,133
   ,203
PZ 255 DATA 169,0,133,19,133,20,32,
   83,228
DD 256 DATA 198,203,208,249,165,20,
   164,19
FB 257 DATA 133,212,132,213,96
```

PROGRAM 18

Golf

■ ■ ■

Program jest formą zabawy, w której siłę uderzenia piłeczki golfowej należy potraktować jako liczbę. Wprowadza się ją w L. 65 w wartościach od 1 do 20. Uderzenie piłeczki jest sygnalizowane dźwiękiem L. 68 : L. 72, a trafienie do dołka krótką melodią L. 98 do L. 103 oraz L. 500 ÷ L. 515. Aby ukończyć grę, należy pięciokrotnie trafić do dołka. Nie jest to proste, gdyż w zależności od podłoża, po jakim porusza się piłeczka oraz odległości do dołka, należy użyć różnej siły. Po trafieniu do ostatniego dołka, podawana jest suma uderzeń w całej grze.

Fragmenty grafiki w kolejnych wierszach umiejscawia się najczęściej za pomocą instrukcji POSITION. Można (jeśli to ułatwi) wykonać ją w inny sposób, a mianowicie przez zmianę położenia kursora. Dokonuje się tego przez naciśnięcie klawisza ESC, a następnie jednocześnie CONTROL i klawisza ze strzałką skierowaną w stronę, w którą ma być przesunięty kursor. Przesuwane są wszystkie znaki wprowadzone za strzałką do zamykającego je cudzysłowu. W niektórych sytuacjach (np. aby uzyskać czytelny wydruk) można znaki przyporządkować liczbie odpowiadającej im wg kodu ATASCII za pomocą instrukcji CHR$. Ilustracją obu przykładów jest L. 300.

Janusz JANIEC

■ ■ ■

```
8 OPEN #1,4,0,"K:"
9 CC=100
10 C=0
15 Z=0
20 E=INT(RND(1)*120+200)
25 Z=Z+1
27 ? CHR$(125)
30 POSITION 6,3:? "ODLEGLOSC DO
DOLKA Nr ";Z;":";E
35 C=C+1
40 IF E<10 THEN 300
45 FOR I=1 TO E/10
50 POSITION I,14:? "-"
55 NEXT I
60 POSITION 4,5:? "WYBIERZ LICZB
E W ZAKRESIE (1-20)"
65 POSITION 32,14:? ;:INPUT S
```

```
66 IF S<1 OR S>20 THEN 65
68 FOR H=15 TO 0 STEP -1
69 SOUND 0,250,10,H
70 FOR W=1 TO 6
71 NEXT W
72 NEXT H
75 ? CHR$(125)
80 E=E-INT(RND(0)*5*S+S)
85 E=ABS(E)
90 IF E>1 THEN 30
94 GRAPHICS 2+16
95 POSITION 5,4:? #6;"TRAFILES"
97 RESTORE
98 READ H1,H2,T
99 SOUND 0,H1,10,15
100 SOUND 1,H2,10,15
101 FOR CZ=1 TO T
102 NEXT CZ
103 IF T<>2 THEN 98
104 GRAPHICS 0
105 IF Z<5 THEN 20
110 POSITION 3,7:? "ZAKONCZYLES
Z ";C;" UDERZENIAMI PILKI."
115 IF C<CC THEN CC=C
120 POSITION 5,5:? "NAJLEPSZY WY
NIK :";CC;" UDERZEN"
150 POSITION 10,16:? "Czy grasz
jeszcze raz ?"
155 POSITION 22,18:? CHR$(212);"
ak/";CHR$(206);"ie"
160 GET #1,P
165 IF P=84 THEN 10
170 IF P<>84 THEN ? CHR$(125)
175 END
300 POSITION 14,8:? CHR$(2);CHR$
(160);CHR$(160);CHR$(136);CHR$(2
9);CHR$(30);CHR$(30);CHR$(30);CH
R$(30);
301 ? CHR$(2);CHR$(32);CHR$(32);
CHR$(7);CHR$(29);CHR$(30);CHR$(3
0);CHR$(30);CHR$(30);
302 ? CHR$(2);CHR$(13);CHR$(13);
CHR$(13);CHR$(29);CHR$(30);CHR$(
30);CHR$(30);CHR$(30);
303 ? CHR$(2);CHR$(29);CHR$(30);
CHR$(2);CHR$(29);CHR$(30);CHR$(2
);CHR$(29);CHR$(30);
304 ? CHR$(2);CHR$(29);CHR$(31);
CHR$(13)
306 POSITION 13,14:? CHR$(13);CH
R$(13);CHR$(7);CHR$(2);CHR$(13);
CHR$(13);CHR$(13);CHR$(13);CHR$(
13);
307 ? CHR$(13);CHR$(13);CHR$(13)
;CHR$(13)
310 POSITION E+16,13:? CHR$(20)
315 GOSUB 400+Z
320 GOTO 60
401 POSITION 15,8:? CHR$(148):RE
TURN
402 POSITION 15,8:? CHR$(144):RE
TURN
403 POSITION 15,8:? CHR$(224):RE
TURN
404 POSITION 15,8:? CHR$(128):RE
TURN
405 POSITION 15,8:? CHR$(251):RE
TURN
500 DATA 60,45,150,72,53,80
505 DATA 96,72,40,81,60,40
510 DATA 60,45,80,0,0,10,60,45,8
0
515 DATA 72,53,150,0,0,2
```

PROGRAM 19

Pojedynek

Jest to prosta, kolorowa gra zręcznościowa dla dwóch osób. Można w nią grać, używając klawiatury bądź joysticka. Gra polega na trafieniu poruszającego się przeciwnika.

Sposób poruszania się oraz oddawania strzałów znajdziecie w wyświetlonym na wstępie opisie gry.

AMSTRAD

```
10 REM P O J E D Y N E K
20 DEFINT a-z
30 MODE 0
40 GOSUB 800
50 GOSUB 1140
60 GOSUB 190
70 GOSUB 1270
80 GOSUB 1140
90 GOSUB 1050
100 REM start
110 IF koniec THEN GOTO 80
120 GOSUB 190
130 FRAME:IF zmie1 THEN GOSUB 480
    ELSE FRAME:FRAME
140 FRAME:IF zmie2 THEN GOSUB 510
    ELSE FRAME:FRAME
```

```
150 IF p1sa=-1 THEN GOSUB 540
160 IF p2sa=-1 THEN GOSUB 570
170 GOTO 100
180 IF j THEN 330 ELSE 410
190 CLS:PEN 6
200 PRINT:PRINT" nacisnij klawisze"
210 PRINT:PRINT:PRINT
220 PRINT " joystick   = 'J'"
230 PRINT:PRINT
240 PRINT " klawiatura = 'K'"
250 PRINT:PRINT:PRINT
260 PRINT " oraz [RETURN]"
270 LOCATE 4,10
280 LOCATE 12,10:IF j THEN PRINT"*":
    ELSE PRINT " "
290 LOCATE 12,11:IF j THEN PRINT" ":
    ELSE PRINT "*"
300 IF NOT(INKEY(45)) THEN j=-1
310 IF NOT(INKEY(37)) THEN j=0
320 IF NOT(INKEY(18)) THEN RETURN
    ELSE 280
330 p1=JOY(0):p2=JOY(1)
340 zmie1=(p1 AND 1)*-1+(p1 AND 2)*0.5
350 zmie2=(p2 AND 1)*-1+(p2 AND 2)*0.5
360 IF p1 AND 16 THEN p1sa=p1sa-1:
    IF p1sa=-1 THEN AFTER 15 GOSUB 600
370 IF p2 AND 16 THEN p2sa=p2sa-1:
    IF p2sa=-1 THEN AFTER 15 GOSUB 600
380 IF p1sa THEN zmie1=0
390 IF p2sa THEN zmie2=0
400 RETURN
410 zmie2=((INKEY(4)=0)*1)+
    ((INKEY(5)=0)*-1)
420 zmie1=((INKEY(69)=0)*1)+
    ((INKEY(71)=0)*-1)
430 IF INKEY(63)=0 THEN p1sa=p1sa-1:
    IF p1sa=-1 THEN AFTER 15 GOSUB 600
440 IF INKEY(10)=0 THEN p2sa=p2sa-1:
    IF p2sa=-1 THEN AFTER 15 GOSUB 600
450 IF p1sa THEN zmie1=0
460 IF p2sa THEN zmie2=0
470 RETURN
480 pt=p1wp+zmie1:IF pt>25 OR pt<6
    THEN RETURN ELSE p1wp=pt
490 zmie1=0
500 PEN 1:LOCATE 3,p1wp:CLS #3:PRINT
    CHR$(209);:RETURN
510 pt=p2wp+zmie2:IF pt>25 OR pt<6
    THEN RETURN ELSE p2wp=pt
520 zmie2=0
530 PEN 2:LOCATE 18,p2wp:CLS #5:PRINT
    CHR$(211);:RETURN
540 PAPER #4,4:WINDOW #4,4,17,p1wp,p1wp:
    CLS#4:FRAME:FRAME
550 PAPER #4,0:CLS #4
560 GOTO 480
570 PAPER #6,5:WINDOW #6,4,17,p2wp,p2wp:
    CLS#6:FRAME:FRAME
580 PAPER #6,0:CLS#6
590 GOTO 510
600 pwpe=(p1wp=p2wp)
610 IF p1sa AND NOT(p2sa) AND pwpe THEN
    p1sc=p1sc+1:SOUND 132,120,10,0,1,0:
    PRINT#1,a$(p1sc);:IF p1sc=9 THEN 680
620 IF p2sa AND NOT(p1sa) AND pwpe THEN
    p2sc=p2sc+1:SOUND 132,100,10,0,1,0:
    PRINT#2,a$(p2sc);:IF p2sc=9 THEN 680
630 IF p1sa THEN SOUND 132,40,70,0,1,1
640 IF p2sa THEN SOUND 132,56,70,0,1,1
650 p1sa=0
660 p2sa=0
670 RETURN
680 PEN 6
690 LOCATE 6,10:PRINT"KONIEC GRY"
700 IF p1sc=9 THEN INK 1,2,20
710 INK 2,0 ELSE INK 2,6,17:INK 1,0
720 SOUND 129,1000,0,12,3
730 SOUND 130,900,0,12,3
740 WHILE INKEY$<>"":WEND
750 t!=TIME:WHILE t!+2000>TIME:WEND
760 WHILE INKEY$="":WEND
770 CLS
780 koniec=-1
790 RETURN
800 a$(0)="111101101101111"
810 a$(1)="001001001001001"
820 a$(2)="111001111100111"
830 a$(3)="111001111001111"
840 a$(4)="100100101111001"
850 a$(5)="111100111001111"
860 a$(6)="111100111101111"
870 a$(7)="111001001010010"
880 a$(8)="111101111101111"
890 a$(9)="111101111001001"
900 FOR n=0 TO 9
910 dlug=LEN(a$(n))
920 FOR n2=1 TO dlug
930 IF MID$(a$(n),n2,1)="1"THEN
    MID$(a$(n),n2,1)=CHR$(143)ELSE
    MID$(a$(n),n2,1)=CHR$(32)
940 NEXT n2,n
950 b$="P O J E D Y N E K"
960 c$=CHR$(32)+"      "+CHR$(164)+
    "  CJS"
970 ENV 1,=9,2000:ENT -1,6,3,1
980 ENV 2,127,0,0,127,0,0,127,0,0,
    127,0,0,127,0,0
990 ENV 3,=9,9000
1000 BORDER 0
1010 INK 0,12:PEN #4,1:PEN #6,2
1020 PEN #1,1:PEN #2,2:PAPER #1,3
1030 PAPER #2,3:PEN #0,6
1040 RETURN
1050 INK 0,12:INK 1,2:INK 2,6:INK 3,13:
     INK 4,20:INK 5,17:INK 6,20
1060 WINDOW #3,3,3,6,25:WINDOW #5,18,18,
     6,25
1070 WINDOW #1,3,5,1,5:WINDOW #2,16,18,
     1,5:WINDOW #7,1,20,1,5:PAPER #7,3
1080 CLS:CLS#7:PRINT#1,a$(0);:PRINT#2,
     a$(0);:p1sc=0:p2sc=0:p1wp=5:p2wp=24:
     zmie1=1:zmie2=1
1090 GOSUB 480:GOSUB 510
1100 SOUND 1,1000,0,12,2
1110 SOUND 2,900,0,12,2
1120 p1sa=0:p2sa=0:koniec=0
1130 RETURN
1140 CLS
1150 PEN 7
1160 FOR n=1 TO LEN(b$)
1170 LOCATE 2+n,10
1180 FOR n2=LEN(b$) TO n STEP -1
1190 PRINT MID$(b$,n2,1)
1200 LOCATE 2+n,10
```

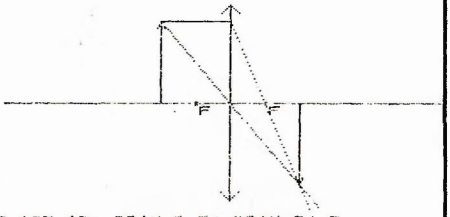
```
1210 SOUND 135,20*n2,5,12,2,1
1220 NEXT n2,n
1230 SOUND 135,100,0,13,3,1,20
1240 PEN 6:PRINT:PRINT:PRINT:PRINT c$
1250 FOR n=1 TO 5000:NEXT
1260 RETURN
1270 IF j THEN RETURN
1280 CLS
1290 LOCATE 6,4
1300 PRINT"STEROWANIE"
1310 PRINT:PRINT
1320 PRINT" GRACZ 1     GRACZ 2
1330 PRINT:PRINT:PRINT
1340 PRINT"    A      gora      6"
1350 PRINT:PRINT
1360 PRINT"    Z      dol       3"
1370 PRINT:PRINT
1380 PRINT"    X      ognia     7"
1390 t'=TIME:WHILE t'+4000>TIME:WEND
1400 LOCATE 7,5
1410 RETURN
```

PROGRAM 20 ZX SPECTRUM

```
  5 REM OBRAZY W SOCZEWCE SKUPIAJACEJ I ROZPRASZAJACEJ, autor JAN GOLLA
  10 INPUT "Podaj ogniskowa soczewki (dla soczewki rozpraszajacej przyjmij f(0)f
=";f: INPUT "Podaj wysokosc przedmiotu (przyjmij h(60) h=";h: INPUT "Podaj odleg
losc przedmiotu od soczewki (przyjmij takie wartosci,by rysunek zmiescil sie na
ekranie)x=";x
  15 LET y=f*x/(x-f): LET h1=f*h/(x-f): REM y-odleglosc obrazu od soczewki,h1-wy
sokosc obrazu
  20 PRINT "f=";f;",";"h=";h;",";"x=";x;",";"y=";INT (y*10)/10;",";"h1=";INT (h1
*10)/10
  30 PLOT 0,80: DRAW 255,0
  40 PLOT 130,20: DRAW 0,120: CIRCLE 130-f,80,1: CIRCLE 130+f,80,1
  42 IF f(0 THEN  GO TO 250
  43 PLOT 125,135: DRAW 5,5: DRAW 5,-5: PLOT 125,25: DRAW 5,-5: DRAW 5,5
  45 PRINT AT 12,32/255*(130-f);"F": PRINT AT 12,32/255*(130+f);"F"
  50 PLOT 130-X,80: DRAW 0,h
  55 PLOT 128-x,76+h: DRAW 2,4: DRAW 2,-4
  60 PAUSE 100
  70 FOR u=130-x TO 130
  75 LET v1=80+h+h/x*(130-x-u)
  80 PLOT u,80+h: PLOT u,v1
  90 NEXT u
 100 FOR u=130 TO 250
 102 LET v1=80+h+h/x*(130-x-u): LET v2=80+h+h/f*(130-u,
 105 IF v1(=0 THEN  GO TO 140
 106 IF v2(=0 THEN  GO TO 125
 107 IF v2)=160 THEN  GO TO 140
 110 PLOT u,v1: PLOT u,v2
 120 NEXT u
 121 IF f(0 OR x(f THEN  GO TO 140
 125 PLOT 130+y,80: DRAW 0,-h1

 130 PLOT 128+y,84-h1: DRAW 2,-4: DRAW 2,4
 135 STOP
 140 FOR u=130-x TO 130+y STEP -3
 145 LET v1=80+h+h/x*(130-x-u)
 150 PLOT u,v1
 160 NEXT u
 170 FOR u=130 TO 130+y STEP -3
 175 LET v2=80+h+h/f*(130-u)
 180 PLOT u,v2
 190 NEXT u
 200 FOR t=80 TO 80-h1 STEP 3
 201 PLOT 130+y,t
 202 NEXT t
 210 PLOT 128+y,76-h1: DRAW 2,4: DRAW 2,-4
 240 STOP
 250 PLOT 125,145: DRAW 5,-5: DRAW 5,5: PLOT 125,15: DRAW 5,5: DRAW 5,-5
 260 GO TO 45
```

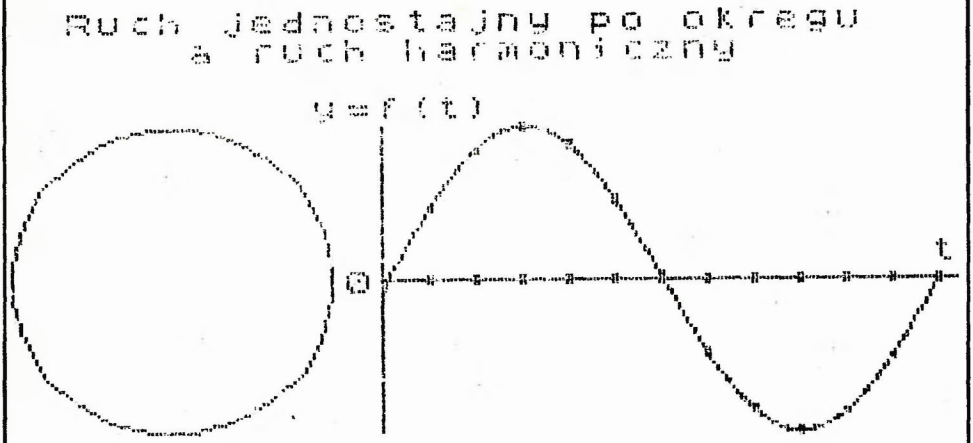
f=80,h=50,x=40,y=40,h1=50



```
 5 REM RUCH JEDNOSTAJNY PO OKRĘGU A RUCH HARMONICZNY      autor JAN GOLLA
 10 PRINT AT 1,3;"Ruch jednostajny po okregu": PRINT AT 2,7;"a ruch harmoniczny

 15 PAUSE 150
 20 DIM x(300): DIM y(300)
 30 CIRCLE 51,92,42
 40 FOR n=0 TO 11
 50 PRINT AT 10-6*COS (-n/6*PI+1/2*PI),6+6*SIN (-n/6*PI+1/2*PI);n
 60 PAUSE 100
 61 FOR n=106 TO 255
 62 PLOT n,92
 63 NEXT n
 70 PLOT 106,50: DRAW 0,84
 80 FOR n=118 TO 255
 90 CIRCLE n,92,1

100 LET n=n+11
110 NEXT n
111 PRINT AT 4,11;"y=f(t)"
112 PRINT AT 9,31;"t"
115 PAUSE 100
120 FOR t=0 TO 11
130 LET a=-t/6*PI+1/2*PI: REM a-kat promienia wodzacego w radianach
140 LET sx=42*SIN a: LET sy=42*COS a
150 PLOT 51,92: DRAW  OVER 1;sx,sy: REM wydruk promienia wodzacego
160 LET u=12*t+106
170 LET x(u)=u: LET y(u)=92+42*SIN (0.0439*u+33)
180 CIRCLE x(u),y(u),1
185 PLOT 51,92: DRAW  OVER 1;sx,sy: REM kasowanie promienia wodzacego
190 PAUSE 50
200 NEXT t
210 FOR t=106 TO 250 STEP 0.5
690 LET x(t)=t: LET y(t)=92+42*SIN (0.0439*t+33)
700 PLOT x(t),y(t)
710 NEXT t
```



Liga Myślących

Zadanie 1

W czasie eliminacji do konkursu należało odpowiedzieć na 30 pytań. Za każdą poprawną odpowiedź przyznawano 7 punktów, a za błędną odpowiedź lub jej brak odejmowano 12 punktów. Na ile pytań odpowiedział prawidłowo uczestnik eliminacji, uzyskując 77 punktów?

Zadanie 2

Gdyby Aleksander Wielki umarł o 5 lat wcześniej, panowałby 1/4 swego życia, gdyby zaś żył o 9 lat dłużej panowałby połowę swego życia. Ile lat żył, a ile panował Aleksander Wielki?

Zadanie 3

Statek ładowano za pomocą 3 dźwigów o tej samej mocy. Po jednej godzinie pracy uruchomiono dodatkowo 3 jednakowe dźwigi o większej mocy i ukończono załadunek statku po 2 godzinach wspólnej pracy wszystkich dźwigów. Gdyby uruchomiono wszystkie dźwigi jednocześnie, to załadunek statku trwałby tylko 2 godziny 24 minuty. Należy obliczyć, w ciągu ilu godzin załadowałby statek jeden dźwig o mniejszej mocy, a w ciągu ilu godzin załadowałby statek jeden dźwig o większej mocy?

Zadanie 4

Dwa samochody wyruszyły jednocześnie z miejscowości A do B. Oba samochody jechały tą samą trasą, ale z różnymi, chociaż stałymi, średnimi prędkościami, które wyrażają się liczbami naturalnymi. Różnica prędkości samochodów jest liczbą pierwszą. Odległość między A i B wynosi 100 km. Po dwóch godzinach jazdy odległość wolniejszego samochodu, od miejscowości A była pięć razy większa od odległości szybszego samochodu, od miejscowości B.
Z jaką średnią prędkością jechały oba samochody?

Rozwiązania zadań prosimy przysyłać pod adresem redakcji do końca września br., z dopiskiem „Liga Myślących". Punktacja zależy od liczby prawidłowych rozwiązań. Wśród uczestników rozlosujemy książki, a na zwycięzców „Ligi" czekają dodatkowe cenne nagrody — niespodzianki.

Ogłoszenia

Sprzedam grę telewizyjną z pistoletem (na gwarancji), stan bardzo dobry. Artur Borowiec, ul. Czachowskiego 19/40, 27—600 Sandomierz.

Wymienię programy na Commodore 64. Sebastian Czartoryski, ul. Łomżyńska 7B 18—300 Zambrów.

Commodore 116, 16, Plus 4 — programy wymienię, kupię, sprzedam. Wtyczki do joysticke'a. Informacje — koperta zwrotna i znaczek. Grzegorz Tesmer, 85—050 Solec Kujawski, skr. pocztowa 28.

Basic ZX Spectrum (polskie tłumaczenie instrukcji). Sprzedaż wysyłkowa. Informacje po nadesłaniu koperty zwrotnej. Jarosław Supłacz, ul. Szarych Szeregów 18 m 20, 09—408 Płock.

Sprzedam Commodore 16 (z pamięcią rozszerzoną do 64 KB), z firmowym magnetofonem, joystickiem i ponad 50 programami. Informacje po przesłaniu koperty ze znaczkiem. Jacek Witt, ul. Kiedrzyńskiego 3/1, 82—300 Elbląg.

Kupię rozszerzenie pamięci do ZX-81 na 64 KB lub 16 KB. Wojciech Suchodolski, ul. Śniadeckich 1/2, 41—808 Zabrze.

Sprzedam lub wymienię na inny Commodore VC-20 + magnetofon + 200 programów. Z. Petkowicz, ul. Jaworska 9 m 11, 58—575 Bolków

REGULAMIN
III konkursu na program gry edukacyjnej

1. Cel konkursu.

Biorąc pod uwagę że:
— głównym problemem rozwojowym w Polsce i na świecie jest bariera edukacyjna,
— jednym z najważniejszych wskaźników rozwoju w roku 2000 będzie ilość i jakość programów edukacyjnych,
Stowarzyszenie Mikrokomputerowe „Abakus" proponuje wszystkim zainteresowanym współudział w tworzeniu takich programów.

Celem konkursu jest tworzenie polskiej kultury mikroinformatycznej, pozyskanie sympatyków i członków Stowarzyszenia na terenie Polski, wzbudzenie szerokiego, społecznego zainteresowania zastosowaniem mikrokomputerów do edukacji oraz wzbogacenie klubowej biblioteki programów.

2. Czym jest gra edukacyjna?

Jest to program na mikrokomputer, który bawi i uczy. Ważniejsza jest jednak w naszym konkursie strona zabawowa gry edukacyjnej, gdyż uważamy, że motywacja ma podstawowe znaczenie w procesie nauczania.
Z drugiej strony kontakt z komputerem powinien być również maksymalnie pożyteczny i kształcący, tzn.: doskonalić umysł człowieka, nauczyć go czegoś nowego, umożliwiać człowiekowi rozwój i doskonalenie się etc.

3. Organizatorzy.

Organizatorem konkursu jest Stowarzyszenie Mikrokomputerowe „Abakus" przy współpracy z Centrum Komputerowym „System" oraz Redakcja miesięcznika „IKS"— dodatku „Żołnierza Wolności". Patronat nad konkursem objęło Biuro ds. Młodzieży Urzędu Rady Ministrów.

4. Uczestnicy.

Uczestnikiem konkursu może być każda osoba fizyczna lub prawna oraz ich zespoły.

5. Wymagania stawiane przesyłanym pracom.

Program gry edukacyjnej może być przeznaczony dla osób w dowolnym wieku i materiał nauczania może być dowolny. Powinien być do niego dołączony opis zawierający krótką charakterystykę, instrukcje ładowania, uruchamiania i możliwości wykorzystania.

Program powinien być przeznaczony na jeden z następujących mikrokomputerów: SPECTRUM 48, TIMEX 2048, ATARI 800 XL, ATARI 130XE, SPECTRAVIDEO 738, COMMODORE 64, AMSTRAD 6128, AMSTRAD 1512 i IBM XT. Prace (programy i opis) należy nadsyłać do 15 września 1987 r. na adres Stowarzyszenia Mikrokomputerowego „Abakus", Warszawa, ul. Senatorska 12

ICL

ICL — tej firmy przedstawiać nie trzeba. Od lat znana jest na polskim rynku. Polscy informatycy jeszcze dziś często pracują na elwrowskiej Odrze — młodszej siostrze ICL 1900. Ale to już historia.

Firma rozwija się dynamicznie, opanowuje nowe technologie i zdobywa nowe rynki zbytu. Warszawskie przedstawicielstwo oferuje DRS 20 i 30 — systemy przeznaczone do prac biurowych. ICL może też kompletnie wyposażyć placówki handlowe począwszy od kas, a na komputerach skończywszy. To oczywiście nie wszystko. Szlagierem jest DRS 300, którego już przedstawialiśmy. Istotną zaletą tego urządzenia jest jego bardzo dobre oprogramowanie. Użytkownik pracuje w pełnym komforcie, komputer podpowiada komendy — zresztą wybór odpowiednich funkcji odbywa się za pomocą myszki, czyli jest bardzo prosty. Jedyna wada to wysoka cena — nawet bogate zakłady nie będą sobie mogły na niego pozwolić.

ICL jest dobrym narzędziem nie tylko dla sekretarek i statystyków, ale także i dla inżynierów i zawodowców zajmujących się projektowaniem systemów informatycznych.

DRS 300 należy do komputerów „łatwych i przyjemnych".

W siedzibie firmy DRS 20 wykorzystywane są w praktyce.

Elektroniczne kasy są jednym z elementów proponowanych przez ICL systemów komputerowych.

MERITUM

Prezentowany program pozwala łączyć przyjemne z pożytecznym. Grający ma okazję dobrze poznać klawiaturę MERITUM, a przy okazji również się zabawić. Część akcyjna gry została napisana w BASICU, a dźwięki wywoływane instrukcjami GET, PUT, OPEN w języku wewnętrznym mikroprocesora Z-80.

Ulepszenie gry nową grafiką i dźwiękami pozostawiam Czytelnikom.

ADAM CYGAN
Uczeń Zasadniczej Szkoły Chemiczno-Mechanicznej w Gliwicach

```
1 ' GRA ZRECZNOSCIOWA "REFLEKS" . AUTOR ADAM CYGAN.
 5 ' GRA POLEGA NA NISZCZENIU BOMB SYMBOLIZOWANYCH
6 ' ZNAKAMI ZA POMOCA ODPOWIEDNICH KLAWISZY.

10 ' GDY ZNISZCZONE ZOSTANA REAKTORY PRZEGRYWASZ
20 CLEAR 1000:CLS:GOSUB 1100:GOSUB 500
25 FOR A=980 TO 992 STEP 4:C=C+1:K(C)=A:NEXT
30 FOR D=1 TO 4:POKE 15360+K(D),127:NEXT
35 V=V+1:L=0
40 FOR M=148 TO 160 STEP 4:POKE 15360+M,128:NEXT
45 FOR C=1 TO 4:GOSUB 500:NEXT
70 DW=0:FOR R=212 TO 224 STEP 4:DW=DW+1:V(DW)=R+15360:NEXT R
80 FOR E=1 TO 4
85 IF T(E)=0 POKE V(E)+S(E)*64,A(E):POKE V(E)+S(E)*64-64,32
87 NEXT
88 C$="":C$=CHR$(PEEK(16537))
90 FOR C=1 TO 4
100 IF C$=A$(C) THEN A$(C)="":GOSUB 500
110 NEXT
130 FOR C=1 TO 4
140 IF PEEK(V(C)+(S(C)+1)*64)<>32 GOSUB 400
150 NEXT
160 FOR F=1 TO 4:S(F)=S(F)+1:NEXT
165 FOR A=1 TO AC(V):NEXT
170 GOTO 80
400 IF PEEK(V(C)+(S(C)+1)*64)=127 GOTO 450
410 IF PEEK(V(C)+(S(C)+1)*64)=128 THEN S(C)=0:RETURN
420 POKE V(C)+(S(C)+1)*64,32:GOSUB 500
430 RETURN
450 GET:PUT:OPEN:OPEN:PUT:GET
460 POKE V(C)+(S(C)+1)*64,128:T1=T1+1:T(C)=1
470 IF T1=4 GOTO 1300
475 S(C)=0:GOSUB 500:RETURN
480 GOSUB 500:RETURN
500 IF L=W(V) GOTO 900
510 RANDOM:A(C)=RND(RL)+MM:A$(C)=CHR$(A(C)):L=L+1
518 POKE V(C)+S(C)*64,32:S(C)=0:PUT:RETURN
800 FOR D=1 TO 10:READ Q,P:FOR X=Q TO P
810 POKE 15360+X,191:NEXT X,D
820 FOR D=1 TO 16:READ Q,P,R:FOR X=Q TO P
830 SET(X,R):NEXT X,D
840 FOR D=0 TO 61:READ B:POKE 32000+D,B:NEXT
845 POKE 16762,0:POKE 16763,125:POKE 16732,11:POKE 16733,125
850 POKE 16768,22:POKE 16769,125:POKE 16771,43:POKE 16772,125
855 RETURN
900 IF V=5 GOTO 1200
910 RESTORE:FOR D=1 TO 20:READ M:NEXT
920 FOR D=0 TO 2:PRINT @D*64,STRING$(64,32);:NEXT
930 FOR D=1 TO 16:READ Q,P,R:FOR X=Q TO P
940 SET(X,R):NEXT X,D:GOTO 35
1100 INPUT"PODAJ STOPIEN TRUDNOSCI (1,2 LUB 3)";AD
1110 IF AD(1 OR AD)3 GOTO1100
1120 CLS:AD=INT(AD):ON AD GOTO 1130,1140,1150
1130 FOR K=1 TO 4:W(K)=30+K*10:AC(K)=250-K*40:NEXT
1135 RL=26:MM=64:RETURN
1140 FOR K=1 TO 4:W(K)=40+K*15:AC(K)=200-K*50:NEXT
1145 RL=34:MM=57:RETURN
1150 FOR K=1 TO 4:W(K)=50+K*20:AC(K)=0:NEXT
1155 RL=59:MM=32:RETURN
1200 PRINT @338,"ZDOLALES SIE OBRONIC.BRAWO!":END
1300 PRINT @333,"WOBEC BRAKU ENERGII MUSISZ SIE PODDAC.":END
2000 DATA 770,785,798,819,832,837,848,867,876,889,896
2010 DATA 919,924,939,948,957,960,975,994,1023
2020 DATA 39,42,8,47,50,8,55,58,8,63,66,8,38,71,7,38,73
2025 DATA 6,38,71,5,41,50,4,55,57,4,64,64,4
2030 DATA 41,44,3,52,54,3,58,63,3,40,44,2,40,43,1,40,42,0
2040 DATA62,255,71,211,254,16,254,61,32,248,201,62,0,71
2050 DATA 211,254,16,254,60,32,248,201,6,255,14,253,71,211
2060 DATA 254,16,254,152,71,211,254,16,254,159,254,128,32
2070 DATA 240,201,229,14,225,33,153,64,70,211,254,61,16
2080 DATA 254,211,254,13,32,245,225,201
```

W naszym komputerlandzie

Kierowanie firmą, w której pracował Spektruś, to proces niezwykle skomplikowany, zaistniała potrzeba powołania brakujących ciał. Spektrusiowi powierzono rolę doradcy naczelnego robota. Wprawdzie niechętnie brał on na siebie ten ciężar, nie był jednak jednostką aspołeczną. Zresztą, gdy ogarnął gospodarstwo wzrokiem, doszedł do wniosku, że faktycznie można wiele zrobić.

Gdy doradził, by poprawić robotom warunki pracy — drgnęło — na razie leciutko. Doradził więc, by podziałać motywacją. Naczelny był zdziwiony, że pieniądze mają dla sprawy aż takie znaczenie. — Popatrzcie, kolego, jak drgnęło! — mówił.

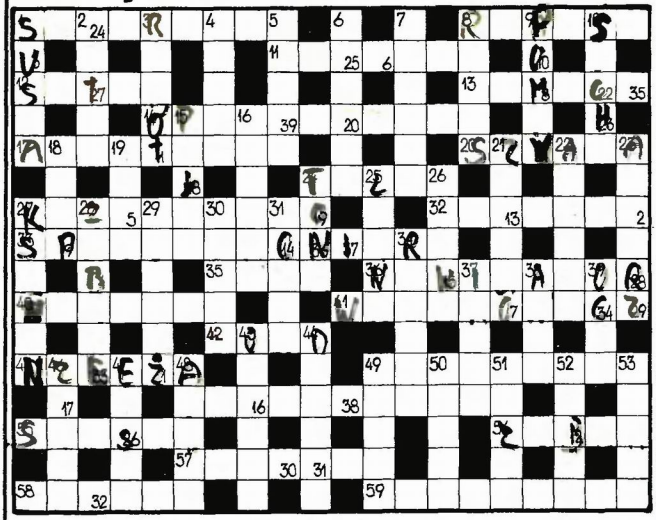
Krzywa na tablicy w gabinecie szefa wznosiła się niemal pionowo, gdy zabrakło surowców: taśm i papieru. Spektruś rozgrzany powodzeniem poradził, by programy drukować na odpadach.

Najtrudniejszy moment jednak miał dopiero nastąpić. Roboty bowiem zaproponowały praktycznie wszystko i gwałtownie spadło zapotrzebowanie na ich pracę. Po jakimś czasie bank poinformował, że firma jest goła.

Nie było wyjścia. Decydując się na gest szaleńca, Spektruś doradził ogłoszenie bankructwa. Dopiero zaczął się ruch w interesie. W największej komputerlandzkiej gazecie ukazał się wywiad z naczelnym, zaś w lokalnej — ze Spektrusiem. Mówili o twardych regułach gry ekonomicznej. Stali się bohaterami. Gazety napisały, że firmie trzeba pomóc, bo bez niej życie byłoby nie do zniesienia. Znalazły się dotacje, były też gratulacje i odznaczenia. Naczelnego przeniesiono do jednostki nadrzędnej, zaś zahartowany w bojach Spektruś awansował do rangi doradcy generalnego.

Podglądał:
Eugeniusz MLECZAK

Krzyżówka nr 7



POZIOMO: 1) metal (np. złoto) znajdowany w ziemi, w stanie rodzimym, 8) dyscyplina sportu dla atlety, 11) cudowny napój poszukiwany przez alchemików, mający przedłużać życie, 12) „mózg armii", 13) tytuł znanego filmu Jerzego Passendorfera, 14) niewielki utwór epicki pisany prozą, 17) kraina historyczna w płd.-zach. Francji (dorzecze Garonny), 20) do zmywania naczyń kuchennych, 24) kupon, kwit upoważniający do zakupu towarów reglamentowanych, 27) zakon krzyżacki, Krzyżacy, 32) słynne wzgórze w centrum Aten z muzealnym zespołem ruin, 33) żołnierz wojsk powietrznodesantowych, 35) wskazówka, 37) członek załogi statku wodnego wyznaczający jego kurs, v0) roślina z rodziny pierwiosnkowatych o kwiatach białych lub różowych, rośnie na skałach w wyższych rejonach górskich, 41) uczeń przebywający samowolnie poza szkołą podczas zajęć szkolnych, 42) zabójstwo, 45) podłużne naczynie drewniane, używane do wyrabiania ciasta, 49) rodzaj opaski, 54) człowiek przebywający gdzieś stale albo czasowo, 55) metoda, 56) pręgowany padlinożerny ssak drapieżny, 57) alkoholowy napój japoński z ryżu; sake, 58) akt prawny ustalany przez Sejm, 59) robotnik pracujący w tkalni przy krosnach.

PIONOWO: 1) długi okres bez deszczu, 2) drewienko do mieszania substancji sypkich i płynów, 3) automat zastępujący pracę człowieka, 4) dzieło ludwisarza, 5) kraj afrykański z Nairobi, 6) erudycja, 7) rzeka we Włoszech (w Jugosławii — Socza), 8) ścięta część pędu drzewa liściastego, 9) modne niegdyś sportowe spodnie męskie o krótkich nogawkach zapinanych pod kolanami, 10) dawne prymitywne narzędzie orne, 15) królewska rezydencja, 16) sztuczne włókno celulozowe, 18) sazan lub lustrzeniec, 19) bardzo twardy metal szlachetny, 21) ekstrakt ze zmielonej kawy naturalnej palonej, 22) płn.-amer. gatunek jelenia, 23) torowiec albo szosowiec, 24) 1000 kg, 25) tropikalne pnącze, 26) miano, 27) gaz szlachetny używany do wypełniania żarówek dużej mocy, 28) Górnik ..... — aktualny mistrz kraju w piłce nożnej, 29) żołnierz dopuszczający się gwałtów, grabieży, 30) dawna świątynia pogańska, 31) ciężki ... — czyli deuter, 34) fason płaszcza lub sukni, 37) rozkaz sułtana do wezyra (anagram wyrazu DIERA), 38) członek kocz. ludu tureckiego lub mongolskiego; od VI — IX wieku osiadłego nad Dunajem, 39) hacel, 43) gospoda, karczma, 44) palisander, 46) sytuacja, położenie bez wyjścia, 47) przelanie praw jednej osoby na drugą, 48) aromatyczna wydzielina z kaszalota, 49) tymczasowy budynek, 50) firmament, 51) archit. dolna część głowicy doryckiej i toskańskiej, 52) np. „Aida" — G. Verdiego, 53) paryski opryszek.

Po prawidłowym rozwiązaniu krzyżówki litery z kratek ponumerowanych dodatkowo w prawym dolnym rogu, czytane w kolejności odpowiadających im liczb od 1 do 39, dadzą hasło, które wystarczy nadesłać jako rozwiązanie zadania, pod adresem redakcji do końca września 1987 r. Wśród czytelników rozlosujemy bony pieniężne i nagrody książkowe.

Mikrociekawostki

Laserowa pamięć ROM

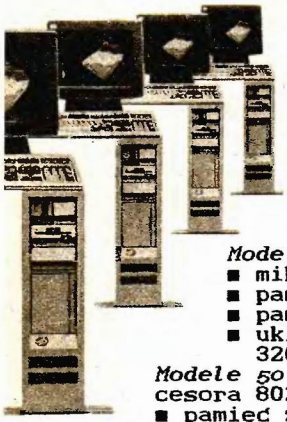
HITACHI oferuje lśniący dysk o średnicy 12 centymetrów. Pamięta on 552 megabajty, czyli 270 000 stron, które można przesłać w średniej kopercie.

A oto kolejne zalety: precyzyjny odczyt laserowy z prędkością 176 000 znaków na sekundę oraz ... praktyczna niezniszczalność.

Jedną wadą jest ROM: raz zapisany dysk można TYLKO ODCZYTYWAĆ (czy to na pewno wada?).

— Kto go tak zaprogramował, żeby biegał za nami po całym biurze?

„IKS" — dodatek „Żołnierza Wolności". Redagują: Wiesław Cetera (kierownik zespołu), Ryszard Rogoń. Stali współpracownicy: Włodzimierz Gogołek, Janusz Janiec, Krzysztof Mamcarz, Ireneusz Miernik, Janusz Miller, Jacek Szaniawski. Adres redakcji: 00-950 Warszawa ul. Grzybowska 77, telefon centrali 20-12-61 w. 486. Telex 313664. Rękopisów nie zamówionych redakcja nie zwraca i zastrzega sobie prawo do skrótów. Nakładem: Wydawnictwa „Czasopisma Wojskowe", Warszawa ul. Grzybowska 77, Fotoskład i druk rotograwiurowy — Wojskowe Zakłady Graficzne im. gen. dyw. A. Zawadzkiego. Nr zam. 9043. Nr. ind. 361682. K-87



I B M Personal System / 2

IBM PS/2 to prawdopodobnie kolejny milowy krok w historii mikrokomputerów ,podobnie jak sześć lat temu IBM PC . Firma *IBM* proponuje cztery modele: 30,50,60,80 .*IBM PS/2* posiadają stacje dysków elastycznych 3 1/2 cala o pojemności 720 KB lub 1,44 MB oraz dyski stałe 3 1/2 cala 20,44,70 lub 115 MB .Mogą mieć dołączoną zew - nętrzną stację dysków 5 1/4 cala oraz stację z zapisem laserowym WORM(write-once read-many) o pojemności 200 MB .

Model 30 jest ewolucją IBM PC/XT ,posiada :

- mikroprocesor 8086 o częstotliwości zegara 8 MHz;
- pamięć zapisywalną 640 KB ;
- pamięć stałą 64 KB z CBIOS (kompatybilne z BIOS);
- układ MCGA zapewniający grafikę o rozdzielczości 320 x 200 pixeli w 256 kolorach .

Modele 50,60 lepiej wykorzystujące możliwości mikropro - cesora 80286 wyposażono w :

- pamięć zapisywalną 1 MB o czasie dostępu 150 ns ,roz - szerzalną do 7 lub 15 MB ;
- pamięć stałą 128 KB z CBIOS i ABIOS ;
- układ VGA zapewniający grafikę o rozdzielczości 640 x 480 pixeli w 256 kolorach ,a po kolejnej rozbudowie 1024 x 768 pixeli.

Model 80 różni się od modeli 50 i 60 tym ,że posiada mikroprocesor 80386 (16 lub 20 MHz) oraz pamięć zapisywalną o czasie dostępu 80 ns i pojemności do 16 MB . Modele *60,80* posiadają jednostkę cent - ralną w obudowie typu *wieża* ,umieszczanej pionowo na podłodze.Ceny modeli w standardowej konfiguracji:£ 1106 ,£ 2658 ,£ 4464 ,£ 7056.

C O M P A Q P O R T A B L E I I I

Jest mikromokputerem przenośnym ,wyposażonym w mikroprocesor 80286 (12 lub 8 MHz) i koprocesor 80287 (8MHz).Pamięć standardowo 640 KB może być rozszerzona do 6,6 MB .Pamięć zewnętrzną stanowią : dwie stacje dysków 5 1/4 cala o pojemności 360KB lub 1,2MB i dysk stały 20 lub 40 MB .Mikrokomputer posiada kartę graficzną pozwalającą na ekranie ciekłokrystalicznym LCD na rozdzielczość w trybie graficz- nym 640 x 400 pixeli (zgodna z IBM CGA) lub kartę EGA . COMPAQ PORTABLE III posiada trzy porty we-wy :równoległy CENTRONICS ,sze- regowy RS 232 C oraz typu DIN do podłączenia kolorowego monitora zewnętrznego .Klawiatura posiada 84 klawisze .System operacyjny to MS DOS w.3.20 .

□□ ■ □□ M I K R O D E P E S Z E □□ ■ □□

■□ *W bieżącym roku należy spodziewać się prototypu mikroprocesora INTEL 80486 , w związku z tym firmy IBM i Microsoft pracują nad nową wersją systemu operacyjnego DOS.*■□ *Najnowsza propozycja firmy BORLAND to kompilator języka C o nazwie Turbo C.* ■□ *DMP 4000 to ostatni model drukarki firmy AMSTRAD kompatybilnej z drukarkami IBM i EPSON .*■□ *Kolejnym modelem plotera firmy ROLAND jest DPX3300 pozwalający wykonywać rysunki formatu A1 na papierze i przeźroczy- stej folii w ośmiu kolorach .* □□ ■ □□ ■ □□ ■ □□

Nr 5 1987/06/05

opracował : Jacek Wojtala

Czy komputer może uczyć?

(Rozmowa druga)

Włodzimierz GOGOŁEK

Kontynuując relację rozmowy z Robertem — jednym z absolwentów „skomputeryzowanej uczelni", uwagę moją zwróciła kwestia doboru informacji wyświetlanych uczącemu się przez komputer. Wykorzystywane są w tym celu elementy sztucznej inteligencji — komputer „rozmawia" z użytkownikiem, korzysta jednak z pewnych niezmiennych całości — umownych porcji. Porcje te zbudowane są zarówno z fragmentów treści nauczania, jak i innych komunikatów (np. aktywizujących pracę studenta — naukę).

Podczas konwersacji (komputer — uczący się) mającej na celu nauczenie umiejętnego wykorzystywania zwrotów w języku niemieckim, kontrola aktywności emocjonalnej (np. czasy reakcji) studenta wskazują, iż zbliża się on do stanu, który popularnie nazywamy nudą. Dowodzi tego znaczne wydłużenie się czasów odpowiedzi na zadawane przez maszynę pytania, a także treść i forma tych odpowiedzi. W tej sytuacji komputer sprawdza (w dostępnym mu zbiorze) dokładnie, z kim ma do czynienia — wiek, płeć, zainteresowania itp. Wartości tych danych to „drogowskazy" w wędrówce po podzbiorach „komunikatów aktywizujących". Mężczyzna 21 lat, kibic sportowy — to dane z kartoteki — nudzi się — pozwala to na identyfikację podzbioru, który zawiera komunikaty aktywizujące, w tym wypadku o treści łatwej do wyobrażenia — np. wyniki ostatnich rozgrywek ligi wraz z komputerowymi przewidywaniami wyników następnej rundy. Podczas uzgadniania poglądów, między komputerem a uczącym się (na temat meczów), maszyna kontroluje zmianę aktywności emocjonalnej i po przekroczeniu pewnego progu (student już się nie nudzi) temat zostaje zmieniony z „ligi" na naukę języka.

Oczywiście w praktyce „drogowskazami" drogi do tematu „pobudzającego rozmowę" są dziesiątki innych wartości między innymi: iloraz inteligencji, stan cywilny, wyniki w nauce, sprawność pamięci, oczu.

Podobnie liczba zbiorów tematów „pobudzających rozmów" jest znaczna, przewiduje bowiem wędkarzy, koneserów kuchni, kina, wielbicieli gwiazd (także śpiewających) itp.

Są to przykłady prostej modyfikacji sprawności uczenia się, stosowane jako ostateczne. Wcześniej istnieje bogaty zestaw innych środków „pobudzania", jakie daje nie treść, lecz forma wyświetlanych na ekranie monitora komputera obrazów (tekstów, rycin, cyfrowych odwzorowań np. fotografii). Okazuje się bowiem, że niewielkie modyfikacje postaci obrazu dostarczają cały arsenał drobnych (nieustannie stosowanych, jak przysłowiowe krople wody spadające na kamień) bodźców, które mają na celu aktywizowanie pracy studentów. Wykorzystano tu udział kolorów, słów i zwrotów na nasze samopoczucie. W praktyce polega to na ciągłym „poznawaniu przez komputer „swoich możliwości".

Na początku eksploatacji programu edukacyjnego (sterującego pracą komputera wspomagającego proces nauczania), wprowadzone zostały do niego między innymi ogólne prawidłowości oddziaływania kolorów na człowieka, na przykład: czerwień pobudza, zieleń uspokaja, niebieski z czernią drażni. Stosując te prawidłowości w praktyce (zmiana barw obrazu w pamięci komputera jest bardzo prostym zabiegiem) maszyna rejestruje reakcje uczących się — różnicę stanu aktywności przed i po zmianie kolorów — w odpowiednich „szufladkach". Przykładowe oznaczenie (identyfikator) takiej szufladki (podzbioru): 19 lat, niewiasta, nie lubi sportu, lubi książki — pamiątki. Mieszczą się w niej kolory: zielony z białym i czarny, jako najskuteczniejszy podczas zwiększania aktywności emocjonalnej studentek mających 19 lat, przedkładających książki nad sport — wniosek ten uzasadniają wyniki 634 prób stosowania tych i innych kolorów w podobnych okolicznościach. Innymi słowy komputer (centralny) kontrolując codziennie pracę 750 uczących się, prezentuje każdemu z nich co najmniej 20 kolorowych obrazów — zdobywa zatem doświadczenie ze stosowania 15 000 obrazów — „widzi", jakie kolory i jak pobudzają użytkowników. Zważywszy na różnice uczących się (płci, wieku itp.) kilka tygodni eksploatacji programu edukacyjnego wystarcza, by „maszyna wiedziała", komu, kiedy i jak ubarwić obraz, by poza treścią niósł określony bodziec (modyfikacji stanu aktywności emocjonalnej).

Aby usprawnić nauczanie, w którym komputer wyręcza pracę nauczyciela, podobnie jak w wypadku kolorów, następuje klasyfikacja niektórych słów, zwrotów. Tworzą one komunikaty, „wtrącane" przez maszynę podczas nauki, modyfikujące aktywność emocjonalną studenta. Oto niektóre przykłady: „Osiągasz wyraźne postępy", „Uczysz się bardzo efektywnie", „Pośpiesz się!", „Szybciej!", „Skoncentruj się!" itp. Dziesiątki tego typu komunikatów komputer wykorzystuje, "poznając" ich skuteczność wobec określonych osób i sytuacji (szufladkuje komunikaty min. wg wieku, płci, zamiłowań).

Przedstawione uwagi dowodzą, że podczas nauki wspomaganej przez komputer — na równi z treścią — forma i sposób przekazywania informacji przez komputer decyduje o efektach współpracy uczącego się z maszyną. Zadbano o to z wielką skrupulatnością — dając uczącym się swego rodzaju rekompensatę za brak „żywego słowa" wypowiadane-

go przez nauczyciela, który poza przekazywaniem wiedzy także się uśmiecha i gestykuluje... Doceniając ten problem część zajęć w uczelni Roberta prowadzą nauczyciele, którzy „uzbrojeni" w komputery dysponują między innymi ekranami monitorów wielkości tablicy szkolnej. Stworzyło to warunki do niemal nieograniczonych możliwości ilustrowania wykładów. Tablica-ekran poza tradycyjnym przeznaczeniem (kredę zastępuje „kolorowe pióro") jest pośrednikiem prezentacji tysięcy obrazów zapisanych w pamięci komputera. Każdy z nich nauczyciel może odszukać i wyświetlić (oczywiście za pomocą maszyny) w czasie krótszym niż trwa wydanie polecenia wykonania tej czynności. Temat ten wiąże się z budową i tak zwanym przetwarzaniem obrazów, które po przetwarzaniu danych i tekstów stało się znaczącym obszarem zastosowań komputerów w ogóle, a w uczelniach w szczególności. Zważywszy na atrakcyjność tego zagadnienia — poprawnej konstrukcji i modyfikacji obrazów w pamięci komputera — stało się ono przedmiotem naszej kolejnej rozmowy z absolwentem skomputeryzowanej szkoły.

W szponach Atari (6)

Procesor obrazu

Tomasz MROWIEC, Ludwik PIELA

Rastrowy system wyświetlania

Dla zrozumienia możliwości graficznych komputera Atari, musimy przypomnieć sobie, jak wyświetlany jest obraz na ekranie telewizora. Z tyłu kineskopu generowany jest strumień elektronów i wystrzeliwany w kierunku ekranu. Wzdłuż swojej drogi przelatuje on między zespołem cewek odchylania poziomego i pionowego, które mogą odchylić strumień. Dzięki temu może on być skierowany do dowolnego punktu ekranu. Elektronika zawarta wewnątrz odbiornika powoduje, że strumień elektronów przemieszcza się po ekranie w sposób regularny. Startuje on w lewym górnym rogu i przesuwa się poziomo. Zmiany intensywności strumienia, występujące w miarę jego ruchu, malują obraz na ekranie. Po osiągnięciu prawego brzegu jest wygaszany i przechodzi do lewego brzegu następnej linii (nie uwzględniamy w opisie wybierania międzyliniowego). Następnie jest włączany i znowu przesuwa się poziomo. Po wypełnieniu całego ekranu i osiągnięciu jego dolnego brzegu strumień elektronów jest wygaszany i wraca do lewego górnego rogu. Następnie cały cyki zaczyna się od nowa. Powtarzany jest 50 razy na sekundę. Taki sposób tworzenia obrazu nazywamy **rastrowym systemem wyświetlania**.

Zdefiniujmy teraz pewne pojęcia. Pojedyncze przejście strumienia przez ekran nazywamy *„poziomą linią ekranu"*. Pozioma linia ekranu jest podstawową jednostką miary odległości pionowej na ekranie. Wysokość obrazu określamy przez podanie liczby poziomych linii ekranu, które go tworzą. Okres, w którym strumień elektronów wraca od prawego do lewego brzegu, nazywamy *„powrotem poziomym"*. Okres, w którym strumień wraca na górę ekranu, nazywamy *„powrotem pionowym"*. Cały proces kreślenia obrazu na ekranie zajmuje 20 tysięcy mikrosekund. Powrót pionowy trwa około 1500 μs, a powrót poziomy — około 14 μs.

Większość odbiorników telewizyjnych projektowana jest z nadmiarem linii; oznacza to, że kreślą one obraz w taki sposób, że brzegi obrazu wykraczają poza ekran. W efekcie nie mamy ramki na ekranie. Jest to jednak niewygodne dla komputerów, ponieważ informacja, która jest poza brzegami ekranu, jest tracona. Z tego powodu obraz tworzony przez komputer musi być trochę mniejszy, niż telewizor może teoretycznie wyświetlić. Atari wykorzystuje tylko 192 poziome linie ekranu. Dlatego granicą rozdzielczości pionowej odbiornika telewizyjnego, używanego z tym komputerem, są 192 punkty elementarne.

Standardową jednostką odległości poziomej jest *„takt koloru"*. Szerokość obrazu określamy przez podanie liczby taktów koloru, które zajmuje. W pojedynczej poziomej linii ekranu istnieje 228 taktów koloru, z których maksymalnie 176 jest widocznych. Dlatego granicą rozdzielczości poziomej standardowego odbiornika telewizyjnego jest 176 punktów elementarnych. Możliwe jest także sterowanie półtaktami koloru, co daje rozdzielczość poziomą 352 punktów elementarnych.

Procesor obrazu — ANTIC

Z tego krótkiego przypomnienia widać, jak skomplikowane i czasochłonne jest tworzenie obrazu na ekranie telewizora lub monitora. Gdyby tym wszystkim zajmował się mikroprocesor, to miałby niewiele czasu na realizację innych funkcji. Dlatego zaprojektowany został specjalizowany układ scalony — procesor obrazu o nazwie **ANTIC** (Alpha-Numeric Television Interface Controller), który zajmuje się tworzeniem obrazu na ekranie i odciąża mikroprocesor. ANTIC jest prawdziwym mikroprocesorem: ma własny zestaw rozkazów. Wykorzystując je możemy napisać program, zwany programem wyświetlania (ang. display list), który steruje wyjściem graficznym komputera. Dostarcza on następujących informacji: gdzie mogą być znalezione dane do wyświetlania, jakich trybów wyświetlania należy używać do interpretacji trybów ekranu i jakie specjalne opcje wyświetlania powinny być używane.

Normalnie program wyświtlania tworzony jest przez system operacyjny podczas realizacji instrukcji **GRAPHICS**. ANTIC wykonje kolejne rozkazy programu wyświetlania. W zależności od kodu rozkazu zawartość pamięci ekranu interpretowana jest jako dane tekstowe lub graficzne. ANTIC wysyła informacje sterujące obrazem do innego procesora (układ GTIA — Graphics Television Interface Adaptor). Jak każdy procesor, ANTIC wykonuje rozkazy programu wyświetlania w następujących etapach:

1. Pobranie kodu rozkazu programu wyświetlania i zapamiętanie go w rejestrze rozkazów.
2. Interpretacja zawartości pamięci jako danych graficznych lub znakowych, na podstawie kodu rozkazu.

3. Jeśli rozkaz wskazuje dane znakowe, ANTIC pobiera bajty z pamięci ekranu, przeszukuje zestaw wzorców znaków i przesyła wzory znaków do wyświetlenia.
4. Jeśli rozkaz wskazuje dane graficzne, ANTIC przesyła je bezpośrednio do wyświetlania.
5. Zwiększenie licznika programu wyświetlania, który wskazuje następny rozkaz do wykonania.
6. Zwiększenie licznika linii ekranu o liczbę bajtów przesłanych z pamięci ekranu do wyświetlania.
7. Powtórzenie tych kroków od początku.

W tym miejscu winni jesteśmy pewne wyjaśnienie. Rozkazy programu wyświetlania określają sposób wyświetlania jednej linii. Lecz nie linii ekranu, ale tak zwanej linii obrazu, która w zależności od używanego trybu, składa się z kilku linii ekranu. Na przykład, w trybie graficznym 2 linia obrazu składa się z 16 linii ekranu, a w trybie 7 — z dwóch linii ekranu. Jak widać, każda linia obrazu może być różnie wyświetlana, w zależności od użytego rozkazu procesora ANTIC. Jednak w trybach graficznych, definiowanych w języku BASIC, wszystkie rozkazy programu wyświetlania określają zwykle ten sam tryb wyświetlania. W efekcie na ekranie otrzymujemy jednolite linie obrazu. Jednak nic nie stoi na przeszkodzie, aby utworzyć dowolny ciąg linii obrazu na ekranie, możemy je zdefiniować za pomocą odpowiednich rozkazów programu wyświetlania.

Program wyświetlania powinien być tak skonstruowany, aby zapewnić:

— zrównoważenie linii obrazu nie mieszczących się na ekranie, do tego przeznaczone są rozkazy wyświetlania pustych linii,
— załadowanie licznika pamięci obrazu adresem początkowym obszaru pamięci, zawierającej aktualne dane graficzne lub tekstowe do wyświetlania,
— posiadanie aktualnych instrukcji wyświetlania określających, który tryb lub tryby graficzne są używane,
— przejście na początek programu wyświetlania (rozkazy skoku) po wykonaniu całego; w pewnych wypadkach rozkazy skoku niezbędne są do kontynuowania programu wyświetlania lub pamięci ekranu poza adresy graniczne.

Zestaw rozkazów procesora ANTIC

Istnieją cztery klasy rozkazów, umożliwiające realizację wymienionych zadań: wyświetlanie grafiki, wyświetlanie znaków, wyświetlanie pustej linii i skoki. Dodatkowo, zgodnie z tymi rozkazami możliwe są następujące opcje: ładowanie licznika pamięci obrazu (LMS), przesuwanie obrazu i przerwania programu wyświetlania (DLI).

Rozkazy wyświetlania grafiki powodują wyświetlanie przez ANTIC linii obrazu zawierającej punkty elementarne w jednolitym kolorze. Wyświetlany kolor pochodzi z rejestru koloru. Wybór tego rejestru określony jest przez wartość danych ekranu. W trybach kreślenia czterokolorowych (tryby BASIC'a 3, 5, 7 i 15 oraz tryby ANTIC'a 8, A, D i E) do wybrania rejestru koloru potrzebna jest para bitów.

Ze względu na to, że do określania jednego punktu elementarnego potrzebne są dwa bity, z każdego bajtu danych ekranu dekodowane są cztery punkty. Na przykład, bajt danych ekranu zawierający wartość $1B (00011011) może wyświetlić 4 punkty; pierwszy będzie w kolorze tła, drugi w kolorze rejestru 0, trzeci w kolorze rejestru 1 i czwarty w kolorze rejestru 2.

W dwukolorowych trybach kreślenia (tryby BASIC'a 4, 6, 14 i 8 i tryby ANTIC'a 9, B, C i F) każdy bit określa jeden z dwóch rejestrów koloru. Wartość 0 wybiera kolor tła dla punktu, a wartość 1 wybiera rejestr koloru 0. W jednym bajcie ekranu może być zawarta informacja o ośmiu punktach.

Istnieje osiem różnych trybów wyświetlania grafiki. Różnią się one liczbą kolorów (2 lub 4), wysokością linii obrazu (1, 2, 4 lub 8 linii ekranu) i liczbą punktów elementarnych,

```
JY 3 REM ************************
PG 4 REM * Program #1: umozliwia*
JV 5 REM * utworzenie dowolnego *
TP 6 REM * programu wyswietlania*
KC 7 REM ************************
WB 8 ? CHR$(125);"   PODAJ ILOSC
   SEGMENTOW EKRANU":INPUT N:N=N+
   1
WP 10 DIM GEKR(N),NRGR(2*N)
PQ 20 GRAPHICS 8
XB 40 FOR I=1 TO N-1:? CHR$(125):
   ? " SEGMENT nr: ";I:? "      Po
   daj tryb ANTIC'u: ";:INPUT A:N
   RGR(1+(I-1)*2)=A
PM 42 ? "      Podaj ilosc  linii:
    ";:INPUT A:NRGR(2*I)=A:NEXT I
DP 43 NRGR(2*N-1)=-1:NRGR(2*N)=0
TJ 140 LST=1536
TG 150 GOSUB 30000
GC 160 ? "PROGRAM WYSWIETLANIA GO
   TOWY"
TO 420 STOP
ZT 20000 REM ********************
EL 20001 REM * Podprogram nr 1: *
AI 20002 REM *  uklada program  *
JD 20003 REM *   wyswietlania   *
AN 20004 REM ********************
RO 30000 SEG=1
IA 30010 LOK=LST
OD 30020 GEKR(0)=PEEK(88)+PEEK(89
   )*256
UX 30030 GRAN=INT((LST/1024+1)*10
   24)
UX 30040 GRAN2=INT(GEKR(0)/4096+1
   )*4096
DQ 30050 FOR X=LOK TO LOK+2
NB 30060 POKE X,112:LOK=LOK+1
RC 30070 NEXT X:I=1
PM 30080 GEKR(SEG)=GEKR(SEG-1)
UP 30100 TRYB=NRGR(1+2*(I-1)):POW
   T=NRGR(2*I):I=I+1
YU 30110 IF TRYB<0 THEN OP=65:ADD
   R=LST:GOSUB 30330:RETURN
CM 30120 PRZYR=40
UM 30130 IF TRYB>=6 AND TRYB<=12
   THEN PRZYR=20
FM 30140 IF TRYB=8 OR TRYB=9 THEN
    PRZYR=10
UY 30150 FOR X=1 TO POWT
VO 30160 IF LOK<>GRAN-3 THEN 3020
   0
QN 30170 OP=1:ADDR=GRAN:GOSUB 303
   30
TT 30180 LOK=LOK+1:GRAN=GRAN+1023
YV 30190 GOTO 30200
IV 30200 POKE LOK,TRYB
SR 30220 IF (SEG<>1 OR X<>1) AND
   (GRAN2-GEKR(SEG))>=PRZYR) THEN
   30280
```

```
SL 30240 OP=TRYB+64
AO 30250 ADDR=GEKR(SEG)
FT 30260 IF GRAN2-GEKR(SEG)<PRZYR
    THEN GRAN2=GRAN2+4096:ADDR=AD
   DR+PRZYR
PP 30270 GOSUB 30330
CK 30280 LOK=LOK+1
TR 30290 GEKR(SEG)=GEKR(SEG)+PRZY
   R
MS 30300 NEXT X
HH 30305 IF TRYB=15 THEN GEKR(SEG
   )=GEKR(SEG)+PRZYR
VF 30310 SEG=SEG+1
BN 30320 GOTO 30080
ZE 30330 POKE LOK,OP
BX 30340 LOK=LOK+1
UW 30350 POKE LOK,ADDR-(INT(ADDR/
   256)*256)
CF 30360 LOK=LOK+1
PW 30370 POKE LOK,INT(ADDR/256)
EO 30380 RETURN
AV 30900 REM ********************
GN 30910 REM * Podprogram nr 2: *
KZ 30920 REM *   ustala adres   *
TD 30930 REM *  pamieci ekranu  *
BL 30940 REM ********************
PT 31000 POKE 88,GEKR(X)-(INT(GEK
   R(X)/256)*256)
RB 31010 POKE 89,INT(GEKR(X)/256)
   :RETURN
```

```
JX 2 REM ************************
UE 3 REM * Program nr 2  nalezy *
ED 4 REM *dograc do poprzedniego*
FV 5 REM *przy pomocy komendy   *
SJ 6 REM *      ENTER "C:"      *
KC 7 REM ************************
HF 8 N=5
WB 40 FOR I=1 TO N-1:READ A,B:NRG
   R(1+(I-1)*2)=A:NRGR(2*I)=B:NEX
   T I
RD 42 DATA 2,2,7,1,15,144,6,2
QY 160 REM
LE 200 POKE 559,0
HN 210 POKE 560,0
LH 220 POKE 561,6
RA 230 POKE 559,34
AZ 240 X=0:GOSUB 31000
ZM 250 POKE 87,0:POKE 752,1
EN 260 POSITION 13,0:? #6;"Wykres
    funkcji"
FR 265 POSITION 13,1:? #6;"-----
   ---------"
BM 270 X=1:GOSUB 31000
KY 280 POKE 87,2
CZ 290 POSITION 6,0:? #6;"y=sin(x
   )"
BG 300 X=2:GOSUB 31000
WL 310 POKE 87,8:COLOR 1
EU 320 PLOT 0,0:DRAWTO 0,140:PLOT
    0,70:DRAWTO 319,70
```

które mieszczą się w jednej poziomej linii obrazu (40, 80. 160 lub 320). Dlatego pewne tryby dają lepszą rozdzielczość; będą one oczywiście wymagać więcej pamięci RAM ekranu. W tablicy 1 prezentujemy zestawienie charakterystyk poszczególnych trybów wyświetlania grafiki.

Rozkazy wyświetlania znaków zmuszają ANTIC do wyświetlania linii obrazu ze znakami. Każdy bajt pamięci ekranu określa jeden znak. Istnieje sześć trybów wyświetlania znaków. W tablicy 2 prezentujemy zestawienie ich charakterystyk.

Rozkazy kreślenia pustych linii powodują wyświetlenie pewnej liczby poziomych linii ekranu w kolorze tła. ANTIC dysponuje ośmioma takimi rozkazami różniącymi się ilością pustych linii ekranu. Najczęściej używany jest rozkaz wysłania ośmiu pustych linii (kod 112 lub 70 szesnastkowo) na ekran. Tablica 3 zawiera zestawienie charakterystyk poszczególnych rozkazów.

Tablica nr 1

Kod rozkazu		Tryb BASIC a	Ilość punktów w poziomie	Ilość bajtów na linię	Ilość linii ekranu	Ilość bitów na punkt
Dziesiętny	Szesnastkowy					
8	08	3	40	10	8	2
9	09	4	80	10	4	1
10	0A	5	80	20	4	2
11	0B	6	160	20	2	1
12	0C	14	160	20	1	1
13	0D	7	160	40	2	2
14	0E	15	160	40	1	2
15	0F	8	320	40	1	1

Tablica nr 2

Kod rozkazu		Tryb BASIC a	Ilość punktów w poziomie	Ilość bajtów na linię	Ilość linii ekranu	Ilość bitów na punkt
Dziesiętny	Szesnastkowy					
2	02	0	40	40	8	8
3	03	—	40	40	10	8
4	04	12	40	40	8	8
5	05	13	40	40	16	8
6	06	1	20	20	8	8
7	07	2	20	20	16	8

Tablica nr 3

Kod rozkazu		Liczba linii ekranu
Dziesiętny	Szestnastkowy	
0	00	1
16	10	2
32	20	3
48	30	4
64	40	5
80	50	6
96	60	7
112	70	8

```
MZ 330 DEG :PLOT 0,70
GE 340 FOR I=1 TO 319
DM 350 DRAWTO I,70-65*SIN(2*I)
GF 360 NEXT I
CB 370 X=3:GOSUB 31000
KM 380 POKE 87,1
RA 390 POSITION 2,0:? #6;"wykonal
   :    TM&LP"
NF 395 POSITION 2,1:? #6;"WARSZAW
   A   1987"
MN 400 GOTO 400
```

```
JW 1 REM ************************
JJ 2 REM *                      *
KW 3 REM *      PROGRAM NR 3    *
JL 4 REM *                      *
KA 5 REM ************************
HF 8 N=5
AI 10 DIM GEKR(N),NRGR(2*N),A$(1)
CU 20 GRAPHICS 8:START=0
WB 40 FOR I=1 TO N-1:READ A,B:NRG
   R(1+(I-1)*2)=A:NRGR(2*I)=B:NEX
   T I
TM 42 DATA 2,2,7,1,15,192,6,2
DP 43 NRGR(2*N-1)=-1:NRGR(2*N)=0
SG 90 GRAPHICS 0:? CHR$(125):CLOS
   E #1
ZF 100 POSITION 6,1:? "WYKRESY FU
   NKCJI BIEGUNOWYCH"
DY 110 POSITION 3,3:? "MENU FUNKC
   JI:":?
RM 115 ? "     1> R=B*Q
     SPIRALA"
DC 120 ? "     2> R=A*(1+COS(Q))
     CARDIOIDA"
XR 125 ? "     3> R=A*(1-SIN(Q))"
AP 130 ? "     4> R=A*SIN(B*Q)
     ROZETA"
KL 135 ? "     5> R=A*COS(B*Q)"
OU 140 ? "     6> R=COS(A*SIN(B*Q)
   )"
OR 145 ? "     7> R=SIN(A*COS(B*Q)
   )"
EH 150 ? "     8> R=A
     WIELOKAT"
AS 155 ? :? "WPROWADZ:":TRAP 90
CV 160 ? "          NUMER FUNKCJI: ("
   ;N;")";:POSITION 27,15:INPUT N
   :? "       PARAMETR A    : (";A;
   ")";:POSITION 27,16:INPU
EM 162 ? "          PARAMETR B    : ("
   ;B;")";:POSITION 27,17:INPUT B
   :? "       PRZYROST      : (";KR
   OK;")";:POSITION 27,18:I
AA 163 ? "          SKALA         : ("
   ;SK;")";:POSITION 27,19:INPUT
   SK
ML 165 N=N+1*(N=0)
DI 170 IF N=1 THEN U=4
AN 175 A=A+1*(A=0)
AY 180 B=B+1*(B=0)
```

ANTIC używa dwóch typów rozkazów skoku. Pierwszy (JMP) jest prostym skokiem bezwarunkowym, który umieszcza w liczniku programu nowy adres, występujący w dwóch bajtach bezpośrednio po kodzie rozkazu i kontynuuje program wyświetlania od nowego adresu. Jego jedyną funkcją jest dostarczenie rozwiązania następującego problemu: rejestr licznika programu ANTIC ma tylko 10 bitów, dlatego program wyświetlania nie może przekroczyć 1KB. Jeśli musi przekroczyć tę granicę, to musi używać rozkazu JMP dla jej przekroczenia. Oznacza to, że program wyświetlania nie jest w pełni przemieszczalny.

Drugi rozkaz skoku (JVB) jest częściej używany. Wprowadza on wartość argumentu do licznika programu ANTIC i czeka na sygnał synchronizacji pionowej, po zakończeniu powrotu pionowego, który wykonuje odbiornik telewizyjny po wykreśleniu ostatniej linii ekranu. Rozkaz ten jest normalnie używany na końcu programu wyświetlania i kontynuuje od nowego adresu (najczęściej początku programu wyświetlania). Przejście na początek programu tworzy nieskończoną pętlę: oczekiwanie na koniec powrotu pionowego sprawia, że pętla jest zsynchronizowana z cyklem wyświetlania obrazu odbiornika telewizyjnego.

JMB i JVB są rozkazami 3-bajtowymi; pierwszy bajt jest kodem rozkazu, drugi i trzeci są adresem skoku.

Opcja LMS wymaga pewnego wyjaśnienia. Wybierana jest przez ustawienie 6 bitu bajtu rozkazu trybu wyświetlania grafiki lub znaków. Kiedy ANTIC spotyka taki rozkaz, wprowadza do licznika pamięci ekranu bajty argumentu. Licznik ten informuje procesor obrazu, gdzie zaczyna się RAM ekranu. Następnie ANTIC rozpoczyna pobieranie z tego obszaru danych do wyświetlania. Rozkaz LMS jest 3-bajtowy: jeden bajt kodu operacji i dwa bajty argumentu. W prostych programach wyświetlania rozkaz LMS używany jest tylko raz, na początku programu wyświetlania. Czasami może być konieczne użycie drugiego rozkazu LMS. Taka potrzeba występuje, kiedy obszar pamięci ekranu przekracza granicę 4 KB. Rejestr licznika pamięci ekranu ma długość 12 bitów, dlatego dane do wyświetlania nie mogą przekroczyć granicy 4 KB. Jeśli przekroczą, musi być użyty dodatkowy rozkaz LMS. Oznacza to, że dane do wyświetlania nie są w pełni przemieszczalne (przynajmniej dla niektórych trybów wyświetlania).

Tworzenie programu wyświetlania

Każdy program wyświetlania powinien rozpoczynać się od trzech rozkazów *„pustych ośmiu linii"*. Wykonuje się to w celu przesunięcia początku obrazu o 24 linie ekranu w dół. Po tym powinna być określona pierwsza linia. Jednocześnie powinno być użyte LMS do przekazania ANTIC, gdzie znajdzie on pamięć ekranu. Następnie zaczyna się właściwy program wyświetlania, który zawiera bajty dla linii obrazu na ekranie. Ogólna liczba poziomych linii ekranu wytworzonych przez program wyświetlania nie powinna przekraczać 192. ANTIC nie zachowuje wymagań czasowych ekranu telewizyjnego. Jeśli damy ANTIC zbyt wiele linii ekranu do wyświetlenia, zrobi to, lecz obraz prawdopodobnie będzie się przesuwał. Wyświetlanie mniej niż 192 linii ekranu nie spowoduje problemów. Programista musi obliczyć sumę poziomych linii ekranu wytworzonych przez program wyświetlania i zweryfikować ją. Program wyświetlania kończy się rozkazem JVB.

W celu utworzenia własnego programu wyświetlania musimy najpierw zaprojektować format obrazu. Wszystkie podręczniki zalecają zrobienie tego na papierze. Na razie zróbmy tak samo. Naszkicujmy obraz ekranu i przetłumaczmy go na ciąg linii obrazu. Przetłumaczmy ciąg linii obrazu na ciąg bajtów określających rozkazy ANTIC. Wstawmy trzy bajty *„pustych 8 linii"* na początku programu. Ustawmy 6 bit pierwszego bajtu programu. Utworzy to rozkaz ładowania licznika pamięci obrazu (LMS). Następne dwa bajty określają adres RAM ekranu. Po nich występują

INFORMATYKA KOMPUTERY SYSTEMY

CENA — 100 zł

Dodatek „Żołnierza Wolności" nr 7/8 /1987/ ISSN0860—2794

W szponach Atari (6)

Procesor obraz

str. 7/1