

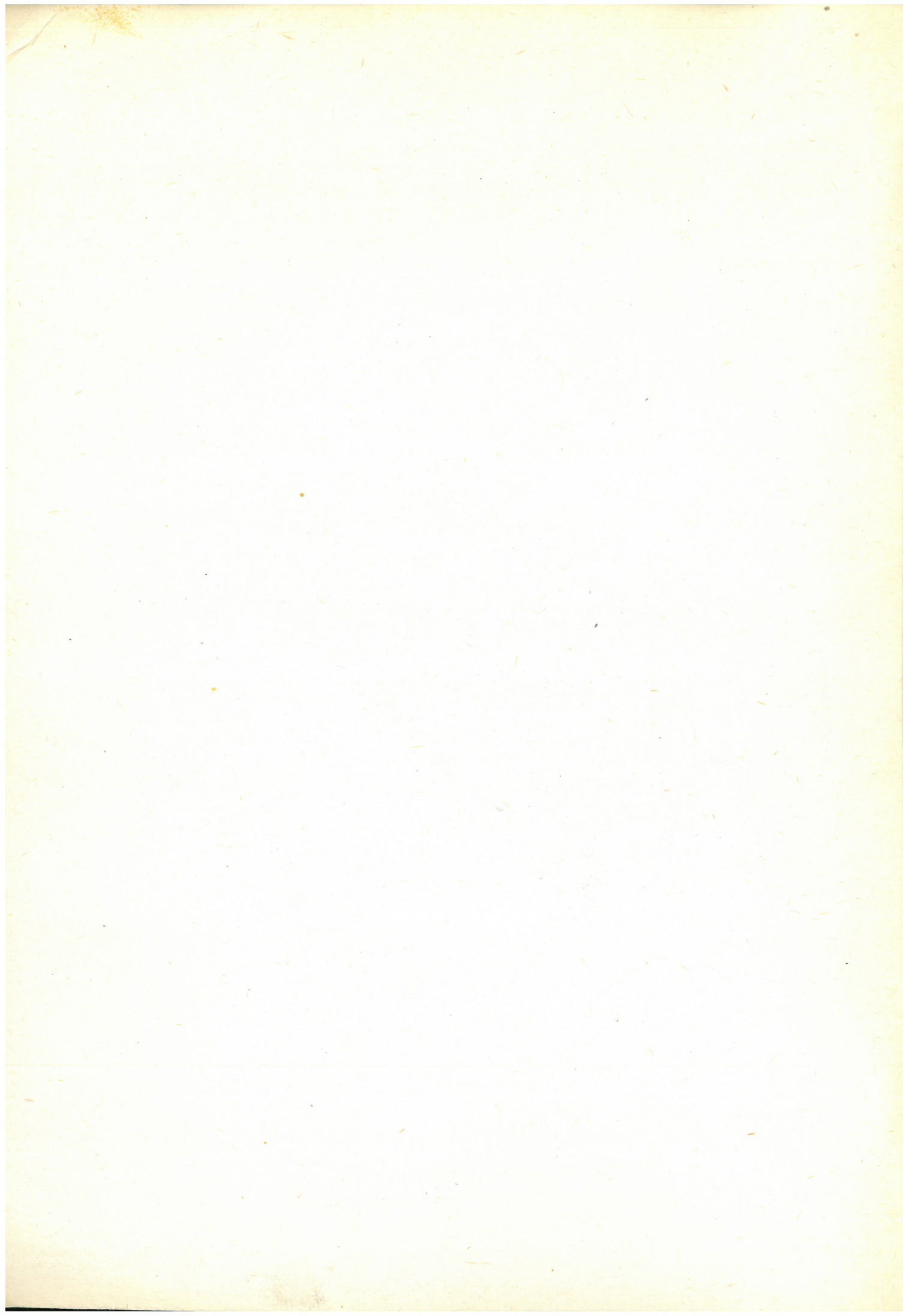
539/bv

004



MERITUM





MERITUM

BIULETYN UŻYTKOWNIKÓW KOMPUTERÓW OSOBISTYCH MERITUM

Do użytku wewnętrznego.

ZUK „MERA-ELZAB” – nakład 1500 egzemplarzy
ZABRZE, grudzień 1985 r.



539/6r

004.38(058)



10009165

WPROWADZONO
DO KOMPUTERA

BIULETYN
UŻYTKOWNIKÓW KOMPUTERÓW OSOBISTYCH

M E R I T U M

Pod redakcją:

mgr inż. Zygmunt KORGA – Mera-Elzab
mgr inż. Andrzej SMOLIŃSKI – Mera-Elzab
mgr inż. Edward SOLARSKI – OETO Pol. Śląskiej
Barbara SORNEK – Sekretarz Redakcji
Mera-Elzab

Adresy dla korespondencji

Zakłady Urządzeń Komputerowych
MERA – ELZAB
ul. Kruczkowskiego 39, 41–808 ZABRZE

Politechnika Śląska
Ośrodek Elektron. Techniki Obliczeniowej
ul. Pstrowskiego 16, 44–100 GLIWICE

SPIS TREŚCI

1.	Disassembler dla MERITUM	5
2.	Opis programu "DEBIT"	13
3.	Zastosowanie MERITUM – I w charakterze sterownika DIGIGRAFU	18
4.	Mikrokomputer na lekcjach matematyki	23
5.	Program "WIELOMIAN"	28
6.	Sprostowanie do artykułu "Wyeliminowanie rozbłysków ekranu w mikrokomputerze MERITUM" ..	36
7.	Propozycje i oferty	37
8.	Konkurs	44
9.	Oferta	46
10.	Wykaz punktów serwisowych MERITUM	50

ROMUALD J. ŻYŁŁA
INSTYTUT INŻYNIERII CHEMICZNEJ
POLITECHNIKA ŁÓDZKA

DISASSEMBLER DLA MERITUM

WSTĘP

Prezentowany disassembler służy do przetwarzania programów w kodzie maszynowym (zawartych w pamięci stałej ROM lub zapisywanej RAM mikrokomputera MERITUM) na kody mnemoniczne mikroprocesora Z-80. Disassembler jest pomyślany jako narzędzie dla zaawansowanego użytkownika mikrokomputerów. Program przetwarza wszystkie opublikowane kody operacji mikroprocesora Z-80.

Dzięki efektywnemu programowaniu wykorzystującemu zalety języka M-BASIC osiągnięto dużą szybkość przetwarzania kodu maszynowego. Język MERITUM BASIC umożliwia deklarację typu DEFINT dzięki czemu wszystkie obliczenia na takich liczbach są wielokrotnie szybsze od obliczeń dla typu SINGLE PRECISION. Deklaracja DEFINT dla listy zmiennych na początku programu uwalnia nas ponadto od pisania znaku procentu przy nazwie zmiennej, co skraca program (krótszy program na ogół szybciej działa). W celu przyspieszenia obliczeń wielokrotnie wykorzystano operacje logiczne AND i OR; na przykład w instrukcjach:

$$L\% = B\%/16 : P\% = B\% \text{ AND } 15 \quad *)$$

zamiast funkcjonalnie im odpowiadających dłuższych i wolniejszych instrukcji:

$$L\% = \text{INT}(B\%/16) : P\% = B\% - L\% \times 16$$

Program w wersji standardowej przetwarza i pokazuje treść programu w języku assamblera z szybkością 220 bajtów na minutę. Prezentowana wersja wykorzystująca krótszą, 11-to bajtową procedurę w kodzie maszynowym posiada szybkość przetwarzania 300 bajtów na minutę.

W programie umieszczono kilka linii komentarza (bez obowiązkowego REM), aby w przypadku błędnego działania pewnych fragmentów programu wykonanie jego zostało wstrzymane i zakończone wydrukiem komunikatu o błędzie. Jeśli program będzie przepisywany bez błędów i uruchomiony, to te linie nie zostaną wykonane i wydruk o błędzie nigdy się nie pojawi. Program był przetestowany przez autora i okazał się pomocny przy "rozgryzaniu" systemu operacyjnego i interpretatora BASIC-a. Dzięki niemu znaleziono w pamięci ROM przydatne procedury i zastosowano je do szybkiego drukowania adresów i kodów operacji w postaci szesnastkowej bez użycia instrukcji PRINT. W prezentowanej wersji programu użyto funkcji USR (N), która powoduje wydruk liczby heksadecymalnej dla argumentu dziesiętnego N <0,255>. Jeśli argument przekracza te granice to wydruk nie następuje. Podczas uruchamiania programu wskaźnik szczytu pamięci zawarty w komórkach 16561, 16562 jest przesuwany o 16 bajtów, a w wolne miejsce jest wpisywany własny kod maszynowy. Adres tej procedury jest wpisywany do komórek 16526, 16527. Procedura USR (N) woła istniejącą już w pamięci ROM procedurę o adresie 321 H wykorzystywaną przez system MONITOR (nie opisany w instrukcji obsługi). W programie zastosowanie instrukcji:

```
200 IF USR (T)
```

jest równoważne instrukcji:

```
200 IF <> 1 THEN PRINT H $ (T/16); H $ (T AND 15);
```

*) Instrukcja P= K AND M oznacza, iż zmienna P przyjmuje wartość iloczynu logicznego zmiennych K i M zapisanych w kodzie binarnym.

OBSŁUGA PROGRAMU

Po uruchomieniu pojawia się tekst:

ADRES HEX / DEC ?

można wtedy wpisać adres, od którego chcemy rozpocząć przetwarzanie kodu maszynowego. Adres można podać w postaci dziesiętnej, np. 6985 lub szesnastkowo 1B49H, przy czym liczba szesnastkowa musi być zakończona literą H i może zawierać cyfry 0 do 9 oraz litery A do F. Po wciśnięciu klawisza ENTER na ekranie pojawi się ten sam adres zapisany w obu systemach, a następnie zaczną się wydruki assemblera. Wstrzymanie wydruków następuje przez jednoczesne wciśnięcie klawiszy SHIFT @, a wznowienie przez wciśnięcie klawisza spacji. Program w czasie pracy reaguje także na klawisze D i M. Klawisz M powoduje przejście do monitorowania, tj. pisania zawartości komórek pamięci w postaci szesnastkowej. W kolumnie z prawej strony ekranu drukowane są znaki alfanumeryczne i semigraficzne o kodach odpowiadających zawartości komórek pamięci. Ta część programu jest przydatna do przeglądania fragmentów pamięci zawierających dane. W MERITUM są to między innymi następujące obszary: od 0105H do 012CH, od 06D2H do 0707H, od 15E3H do 1935H, od 3434H do 347BH oraz drobne fragmenty danych dla procedur matematycznych. Program zatrzymuje się sam po pokazaniu zawartości kolejnych 128 komórek i pyta o nowy adres. Jeśli zamiast podania adresu naciśnie się tylko ENTER to program kontynuuje swoją pracę począwszy od adresu, na którym przerwał. Można także zamiast adresu podać liczbę dziesiętną poprzedzoną znakiem plus lub minus, co spowoduje obliczenie nowego adresu względem ostatnio obsługowanego. W czasie monitorowania pamięci program reaguje na klawisze M, A i D, gdzie klawisz D powoduje przeskoczenie do disassemblera i pytanie o adres. Wciśnięcie tylko klawisza ENTER spowoduje disassemblację od adresu, na którym skończyło się monitorowanie.

Wyniki działania programu można wyprowadzić na drukarkę (np. KSR 180), jeśli w linii 190 zmieni się instrukcję PRINT na LPRINT.

WPISYWANIE PROGRAMU

Ze względu na czytelność programu słowa kluczowe i instrukcje są rozdzielone spacjami. Podczas przepisywania programu można je pominąć bez szkody dla działania programu. Wyjątek stanowią spacje w instrukcjach DATA i w podstawieniach za zmienne typu łańcuchowego. Instrukcje komentarza REM mogą być skrócone, ale program będzie nieprawidłowo działał po ich całkowitym usunięciu. Przed pierwszym uruchomieniem zaleca się kilkakrotne zapisanie programu na taśmie magnetofonowej.

- Uwaga: 1) Program nie może być wykorzystywany do celów zarobkowych bez uzgodnienia z autorem.
2) Program na taśmie można otrzymać od autora na zasadzie wymiany lub kupna po uzgodnieniu warunków.


```
10 POKE 16561,239: CLEAR 90
  : DEFINT A,B,D,E,F,P,I,J,K,H,L,R,S,T,U,Z
  : GOSUB 90: PRINT @AT-56,"Z - 80";
30 DIM H$(15),A$(7),B$(3),C$(7),E$(7)
  ,M$(7),P$(3),R$(7),T$(7),W$(7),Z$(3)
  : FOR I=0 TO 15: READ H$(I): NEXT
  : DATA 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
40 FOR I=0 TO 7
  : READ W$(I),C$(I),A$(I),R$(I),M$(I),T$(I),E$(I): NEXT:
  DATA NZ,NOP,B,RLCA,"ADD A," ,RLC ,LD:
  DATA Z , "EX AF,AF'" ,C,RRCA,"ADC A," ,RRC ,CP:
  DATA NC,DJNZ ,D,RLA,SUB ,RL ,IN:
  DATA C ,JR ,E,RRA,"SBC A," ,RR ,OUT
50
  DATA PO,NZ,H,DAA,AND ,SLA ,I:
  DATA PE,Z ,L,CPL,XOR ,SRA ,D:
  DATA P ,NC,(HL),SCF,OR ,SLI ,IR:
  DATA M ,C ,A,CCF,CP ,SRL ,DR
60 FOR I=0 TO 3: READ B$(I),P$(I),Z$(I): NEXT:
  DATA BC,BC,NIELEGALNY,DE,DE,BIT :
  DATA HL,HL,RES ,SP,AF,SET :
70 READ S$,P$,L$,J$: N$=Z$(0): DATA " ",",",",LD ,JP :
80 IF J$="JP " THEN ON ERROR GOTO 4300: GOTO 100
  ELSE PRINT"BLEDY W CZYTANIU DANYCH": STOP
90 L=PEEK(16416): H=PEEK(16417)
  : AT=L+256*(H AND3): RETURN
100 PRINT ,"DISASSEMBLER": GOSUB 3000
110 B=1: S=-1: T=S: U=S: R=PEEK(AD): Z=R/64
120 D=R AND7: P=R/8 AND 7: L=R AND 15: H=R/16
130 GOSUB 300: REM DEKODUJ BAJTY
140 H$="": ON B GOTO 180,170,160
150 H$=H$(U/16)+H$(U AND 15)
160 H$=H$(T/16)+H$(T AND 15)+H$
170 H$=H$(S/16)+H$(S AND 15)+H$
180 H$=H$(R/16)+H$(R AND 15)+H$
190 PRINT H$(AD!/4096)+H$(AD!/256 AND 15)
  +H$(AD!/16 AND 15) +H$(AD AND 15)+"H";
  : PRINT TAB(10) H$+"H"; TAB(20) C$
200 A$=INKEY$:
  IF A$="D" THEN 100 ELSE IF A$="M" GOTO 4000
210 AD!=AD!+B: AD=AD+B: GOTO 110
290 KONIEC GLOWNEJ PETLI.
  JESLI MASZ UWAGI DO DZIALANIA TEGO PROGRAMU
  TO ZADZWON DO ROMKA: LODZ, 36-55-22 W. 843 LUB 837
300 C$=N$: ON Z GOTO 550,560,570
310 ON D GOTO 360,400,500,510,520,530,540
320 IF P<2 LET C$=C$(P): RETURN ELSE GOSUB 950
330 IF P<4 LET C$=C$(P)+AD$ ELSE C$=C$(3)+C$(P)+P$+AD$
340 RETURN
360 IF L=1 GOSUB 900: C$=L$+B$(H)+P$+H$
  ELSE IF L=9 LET C$="ADD HL,"+B$(H)
370 RETURN
400 IF P>3 GOSUB 900 ELSE H$=B$(H)
410 D$="("+H$+")": IF P>3 GOTO 450
420 IF L=2 LET C$=L$+D$+" ,A": RETURN
  ELSE C$=L$+"A,"+D$: RETURN
450 ON P-4 GOTO 470,480,490
460 C$=L$+D$+" ,HL": RETURN
470 C$=L$+"HL,"+D$: RETURN
480 C$=L$+D$+" ,A" : RETURN
490 C$=L$+"A,"+D$ : RETURN
```

```
500 IF L=3 LET C$="INC "+B$(H): RETURN
    ELSE C$="DEC "+B$(H): RETURN
510 C$="INC "+A$(P): RETURN
520 C$="DEC "+A$(P): RETURN
530 GOSUB 910: C$=L$+A$(P)+P$+H$: RETURN
540 C$=R$(P): RETURN
550 IF I=118 LET C$="HALT": RETURN
    ELSE C$=L$+A$(P)+P$+A$(D): RETURN
560 C$=M$(P)+A$(D): RETURN
570 ON D GOTO 590,630,640,730,740,770,790
580 C$="RET "+W$(P): RETURN
590 IF L=1 LET C$="POP "+P$(H-12): RETURN
600 IF P=1 LET C$="RET"
    ELSE IF P=3 LET C$="EXX"
610 IF P=5 LET C$=J$+"(HL)"
    ELSE IF P=7LET C$=L$+"SP,HL"
620 RETURN
630 GOSUB 900: C$=J$+W$(P)+P$+H$: RETURN
640 ON P GOTO 1000,670,680,690,700,710,720
650 GOSUB 900: C$=J$+H$: RETURN
670 GOSUB 910: C$="OUT (" +H$+"),A": RETURN
680 GOSUB 910: C$="IN A, (" +H$+")": RETURN
690 C$="EX (SP),HL": RETURN
700 C$="EX DE,HL":RETURN
710 C$="DI": RETURN
720 C$="EI": RETURN
730 GOSUB 900: C$="CALL "+W$(P)+P$+H$: RETURN
740 ON P GOTO 760,750,1500,750,2000,750,1500
750 C$="PUSH "+P$(H-12): RETURN
760 GOSUB 900: C$="CALL "+H$: RETURN
770 GOSUB 910: C$=M$(P)+H$: RETURN
790 I=R-199: C$="RST "+H$(I/16)+H$(I AND15)+"H"
: ON P GOTO 810,820,830,840,850,860,870
800 C$=C$+" RESTART": RETURN
810 B=2: S=PEEK(AD+1)
: C$=C$+" POROWNAJ SYMBOL "+CHR$(S): RETURN
820 C$=C$+" SPRAWDZ NAST. SYMBOL": RETURN
830 C$=C$+" POROWNAJ DE:HL": RETURN
840 C$=C$+" TESTUJ TRYB DANEJ": RETURN
850 C$=C$+" -----": RETURN
860 C$=C$+" WYPROWADZ ZNAK": RETURN
870 C$=C$+" ZGLOS PRZERWANIE": RETURN
890 PROCEDURE
900 B=3: T=PEEK(AD+2): H$=H$(T/16)+H$(T AND15): GOTO 920
910 B=2: H$=""
920 S=PEEK(AD+1): H$=H$+H$(S/16)+H$(S AND15)+"H"
: RETURN
950 B=2: S=PEEK(AD+1): DI=S: IF S>127 LET DI=S-256
960 A!=AD+2+DI
: IF A!<0 THEN AS=A!: A!=A!+65536
    ELSE IF A!>32767 THEN AS=A!-65536 ELSE AS=A!
970 AD$=H$(A!/4096)+H$(A!/256 AND15)
+ H$(AS/16 AND15)+H$(AS AND15)+"H": RETURN
1000 B=2: REM *** CB ***
1010 S=PEEK(AD+1): Z=S/64: D=S AND7: P=S/8 AND7
1020 IF Z LET C$=Z$(Z)+H$(P)+P$ ELSE C$=T$(P)
1030 C$=C$+A$(D)
: RETURN
```

```
1500 REM *** DD I FD ***
1510 B=2: S=PEEK(AD+1): D=S AND7: P=S/8 AND7: C#=N$
1520 IF R=221 LET I$="IX" ELSE IF R=253 LET I$="IY"
1530 ON S/64 GOTO 1700,1750,1800
1540 ON D GOTO 1560,1590,1610,1630,1630,1630: RETURN
1560 IF P=5 LET C$="ADD "+I$+P$+I$: RETURN
1570 IF P AND1 LET C$="ADD "+I$+P$+B$(P/2): RETURN
1580 IF P=4 GOSUB 2500: C#=L$+I$+P$+HL$: RETURN
1590 IF S=34 GOSUB 2500: C#=L$+"(+HL$+)", "+I$: RETURN
1600 IF S=42 GOSUB 2500: C#=L$+I$+"(+HL$+)": RETURN
1610 IF S=35 LET C$="INC "+I$: RETURN
1620 IF S=43 LET C$="DEC "+I$: RETURN
1630 IF S=52 GOSUB 2700: C$="INC (+I$+D$+)": RETURN
1640 IF S=53 GOSUB 2700: C$="DEC (+I$+D$+)": RETURN
1650 IF S=54 GOSUB 2700: GOSUB 2800
: C#=L$+"(+I$+D$+)", "+H$: RETURN
1660 RETURN
1700 IF (D=6 OR P=6) AND (D<>P) GOSUB 2700 ELSE RETURN
1720 I$="(+I$+D$+)"
1730 IF D=6 LET C#=L$+A$(P)+P$+I$ ELSE C#=L$+I$+P$+A$(D)
1740 RETURN
1750 IF D=6 GOSUB 2700: C#=M$(P)+"(+I$+D$+)"
1760 RETURN
1800 IF S=203 GOTO 1900
1810 IF S=225 LET C$="POP "+I$: RETURN
1820 IF S=227 LET C$="EX (SP)", "+I$: RETURN
1830 IF S=229 LET C$="PUSH "+I$: RETURN
1840 IF S=233 LET C$="J$+"(+I$+)": RETURN
1850 IF S=235 LET C$="EX DE", "+I$: RETURN
1860 IF S=249 LET C#=L$+"SP", "+I$: RETURN
1870 RETURN
1900 REM CZTEROBAJTOWE * DD CB ***
1910 B=4: T=PEEK(AD+2): U=PEEK(AD+3)
: Z=U/64: D=U AND7: P=U/8 AND7
1920 IF D<>6 RETURN
1930 DI=T: IF T>127 LET DI=T-256
1940 D$=STR$(DI): IF DI>=0 LET D$="+"MID$(D$,2)
1950 IF Z LET C#=Z$(Z)+H$(P)+P$ ELSE C#=T$(P)
1960 C#=C$+"(+I$+D$+)": RETURN
2000 REM KODY *** ED ***
2010 B=2: S=PEEK(AD+1): D=S AND7: P=S/8 AND7: C#=N$
2020 IF (S AND192)<>64 GOTO 2200
2050 IF D>3 GOTO 2130 ELSE ON D GOTO 2060,2090,2110
2060 IF D=1 LET C$="OUT (C)", "+A$(P)
ELSE C$="IN "+A$(P)+", (C)"
2070 IF P=6 LET C#=N$
2080 RETURN
2090 IF P AND1 LET C$="ADC" ELSE C$="SBC"
2100 C#=C$+" HL", "+B$(P/2): RETURN
2110 IF (P OR1)=5 RETURN ELSE GOSUB 2500: C$="(+HL$+)"
2120 IF P AND1 THEN C#=L$+B$(P/2)+P$+C$: RETURN
ELSE C#=L$+C$+P$+B$(P/2): RETURN
2130 IF S=68 LET C$="NEG" ELSE IF S=70 LET C$="IM0"
2140 IF S=69 LET C$="RETN" ELSE IF S=77 LET C$="RETI"
2150 IF S=71 LET C#=L$+"I,A" ELSE IF S=87 LET C#=L$+"A,I"
2160 IF S=79 LET C#=L$+"R,A" ELSE IF S=95 LET C#=L$+"A,R"
2170 IF S=86 LET C$="IM1" ELSE IF S=94 LET C$="IM2"
2180 IF S=103 LET C$="RRD" ELSE IF S=111 LET C$="RLD"
2190 RETURN
2200 IF (S OR 27)<>187 RETURN
2210 IF (S OR 8)=187 LET C$="OT"+E$(P): RETURN
ELSE C#=E$(D)+E$(P): RETURN
```

```
2500 B=4: T=PEEK(AD+2): U=PEEK(AD+3)
2510 HL#=H$(U/16)+H$(UAND15)+H$(T/16)+H$(T AND15)+"H"
: RETURN
2700 B=3: T=PEEK(AD+2): DI=T: IF T>127 LET DI=T-256
2710 D$=STR$(DI): IF DI>=0 LET D$="+"+MID$(D$,2)
2720 RETURN
2800 B=4: U=PEEK(AD+3): D=U AND7: P=U/8 AND7
: H#=H$(U/16)+H$(U AND15)
: RETURN
2810 H#=H$(U/16)+H$(U AND 15)
: RETURN
2830 KONIEC DROBNYCH PROCEDUR
3000 AD$="": INPUT "ADRES HEX/DEC";AD$: GOSUB 90
3010 IF AD$="" THEN 3080
ELSE IF AD$="D" THEN 100
ELSE IF AD$="M" THEN 4000
3020 IF RIGHT$(AD$,1)="H" THEN AD!=0: GOTO 3050
3030 A=ASC(AD$)
: IF A=43 OR A=45 THEN AD!=AD!+VAL(AD$): GOTO 3080
ELSE IF A>47 AND A<58 LET AD!=VAL(AD$): GOTO 3080
3040 PRINT"ADRESY HEX ZAPISUJ NP. 01AFH":GOTO 3000
3050 FOR I=1 TO LEN(AD$)-1: L=0: A=ASC(MID$(AD$,I))
3060 IF 47<A AND A<58 THEN L=A-48.
ELSE IF 64<A AND A<71 THEN L=A-55 ELSE 3040
3070 AD!=16*AD!+L: NEXT
3080 GOSUB 4200
3100 AD$= H$(AD!/4096) + H$(AD!/256 AND 15)
+ H$(AD/16 AND 15) + H$(AD AND 15)
3110 PRINT @AT-64,"ADRES ";AD$+"H /";AD%;"DEC"
: RETURN
4000 PRINT,"MONITOR PAMIECI"
4010 GOSUB 3000: B=1
4020 FOR J=0 TO 7: I=AD AND15: D=3*I+AT: E=47+I+AT
: FOR I=I TO15: P=PEEK(AD)
4030 PRINT@D,H$(P/16)+H$(P AND 15);: D=D+3
4040 IF P<32 OR P=127 OR P>191 THEN A$=""
ELSE A$=CHR$(P)
4050 PRINT@E,A$;:E=E+1: AD!=AD!+B: AD=AD+B: NEXT
: PRINT
4060 AT=(AT AND1984)+64: IF AT>1023 LET AT=AT-64
4070 A$=INKEY$: IF A$="" GOTO 4100
4080 IF A$="D" GOTO 100 ELSE IF A$="M" GOTO 4000
4090 IF A$="A" PRINT CHR$(94);AD
4100 NEXT J: GOTO 4010
4110 KONIEC MONITORA
4130 :
4200 IF AD!<0 THEN AD!=AD!+65536: GOTO 4200
ELSE IF AD!>65535 THEN AD!=AD!-65536: GOTO 4200
4210 IF AD!<32768 THEN AD=AD! ELSE AD=AD!-65536
4220 RETURN
4300 IF ERR=8 OR ERR=10 OR ERR=16 GOSUB 4200
ELSE ON ERROR GOTO 0
4310 RESUME
4444 KONIEC PROGRAMU 30.7.1985
```

LINIE W WERSJI PROGRAMU WYKORZYSTUJACEJ
PROCEDURE WYDRUKU W FORMACIE HEKSADECYMALNYM

```
20 POKE 16526,240: POKE 16527,127:
FOR I=0 TO 10: READ B: POKE 32752+I,B: NEXT:
DATA 205,127,10,124,183,192,125,205,17,50,209:
REM WYDRUK HEX
CALL 0A7FH
LD A,H
OR A
RET NZ
LD A,L
CALL 3211H
RET
```

```
140 REM WYDRUKUJ ADRESY I KODY W HEX
150 IFUSR(AD!/256) OR USR(AD AND 255)
160 PRINT "H"; TAB(10);
170 IFUSR(R) OR USR(S) OR USR(T) OR USR(U)
180 PRINT "H"; TAB(20); C$
```

```
4030 PRINT@D, :: P=USR(P): D=D+3:
IF P<32 OR P=127 OR P>191 THEN A$=" "
ELSE A$=CHR$(P)
```

Przykład działania programu DISASSEMBLER

0000H	F3H	DI	
0001H	AFH	XOR	A
0002H	C3A937H	JP	37A9H
0005H	C30040H	JP	4000H
0008H	C30040H	JP	4000H
000BH	E1H	POP	HL
000CH	E9H	JP	(HL)
000DH	C39F06H	JP	069FH
0010H	C30340H	JP	4003H
0013H	C5H	PUSH	BC
0014H	0601H	LD	B,01H
0016H	182EH	JR	0046H
0018H	C30640H	JP	4006H
001BH	C5H	PUSH	BC
001CH	0602H	LD	B,02H
001EH	1826H	JR	0046H
0020H	C30940H	JP	4009H
0023H	C5H	PUSH	BC
0024H	0604H	LD	B,04H
0026H	181EH	JR	0046H
0028H	C30C40H	JP	400CH
002BH	111540H	LD	DE,4015H
002EH	18E3H	JR	0013H
0030H	C30F40H	JP	400FH
0033H	111D40H	LD	DE,401DH
0036H	18E3H	JR	001BH
0038H	C31240H	JP	4012H
003BH	112540H	LD	DE,4025H
003EH	18DBH	JR	001BH
0040H	C31905H	JP	05D9H
0043H	C9H	RET	
0044H	00H	NOP	
0045H	00H	NOP	
0046H	C3C203H	JP	03C2H
0049H	CD2B00H	CALL	002BH
004CH	B7H	OR	A
004DH	C0H	RET	NZ
004EH	18F9H	JR	0049H
0050H	0DH	DEC	C
0051H	0DH	DEC	C
0052H	1FH	RRA	
0053H	1FH	RRA	
0054H	01015BH	LD	BC,5B01H
0057H	1BH	DEC	DE
0058H	0AH	LD	A,(BC)
0059H	1AH	LD	A,(DE)
005AH	08H	EX	AF,AF'
005BH	1809H	JR	0066H
005DH	19H	ADD	HL,DE
005EH	2020H	JR	NZ,0080H
0060H	0BH	DEC	BC
0061H	78H	LD	A,B
0062H	B1H	OR	C
0063H	20FBH	JR	NZ,0060H
0065H	C9H	RET	
0066H	C30E40H	JP	400EH
0069H	3AEC37H	LD	A,(37ECh)
006CH	3CH	INC	A
006DH	FE02H	CP	02H
006FH	00H	NOP	
0070H	00H	NOP	
0071H	00H	NOP	
0072H	C3CC06H	JP	06CCH
0075H	118040H	LD	DE,4080H
0078H	21F718H	LD	HL,18F7H
007BH	012700H	LD	BC,0027H

BOGUSŁAW PIOTROWSKI
ZUK MERA-ELZAB

OPIS PROGRAMU "DEBIT"

1. WSTĘP

W mikrokomputerach MERITUM produkowanych do sierpnia tego roku, w EPROM—ie nr 6 znajdował się program "DEBIT" umożliwiający uruchamianie i analizowanie programów napisanych w języku wewnętrznym mikroprocesora Z-80.

2. Wywoływanie programu "DEBIT"

Po wprowadzeniu zlecenia SYSTEM i po pojawieniu się zaproszenia "*"?" należy wpisać:

/12288 ENTER

Program DEBIT zgłasza się znakiem "***". Dane niezrozumiałe dla komputera są ignorowane, wyświetlany jest znak "?" i ponownie zgłasza się DEBIT.

Przykład:

* 454 U ?
*

3. Zlecenia DEBIT-u

D — WYPROWADZENIE OBSZARU PAMIĘCI W POSTACI LICZB SZESNASTKOWYCH

Postać zlecenia:

* D ADRP ADRK ENTER

Oznaczenia:

ADRP — adres początkowy obszaru wyświetlanego

ADRK — adres końcowy obszaru wyświetlanego

Zawartość pamięci wyprowadzana jest od adresu początkowego do adresu końcowego włącznie. W jednej linii ekranu wyświetlony jest bieżący adres i szesnaście wartości w kodach szesnastkowych. Po wyświetleniu obszaru pamięci ponownie zgłasza się DEBIT.

Uwaga!

- Debit przyjmuje dane tylko w postaci szesnastkowej
- Dane wprowadzane w zleceniu mogą być oddzielone od siebie zarówno przecinkiem "," jak i spacją " "
- Znaki podkreślone pisze operator, znaki niepodkreślone — komputer.

Powyższe uwagi dotyczą zlecenia D oraz zleceń opisanych poniżej.

Przykład:

* D 1000, 102 C

```
1000 20 28 07 78 B9 0E 2A 20 01 41 71 D7 28 14 FE 45
1010 28 10 FE 44 28 0C FE 30 28 F0 FE 2C 28 EC FE 2E
1020 20 03 2B 36 30 7B E6 10 28 03 2B 36 24
```

*

E - ZAINICJOWANIE WYKONANIA PROGRAMU W JĘZYKU WEWNĘTRZNYM OD PODANEGO ADRESU

Postać zlecenia:

* E ADRA, ADRS

Oznaczenia:

- ADRA - aktualny adres licznika rozkazów
- ADRS - adres startu (nowy adres licznika rozkazów)

Zlecenie powoduje skok do programu rozpoczynającego się od ADRS. Po wykonaniu programu (o ile program nie powoduje zawieszenia się systemu) ponownie zgłasza się DEBIT.

M - ZMIANA ZAWARTOŚCI NIEKTÓRYCH KOMÓREK PAMIĘCI LUB WPISANIE PROGRAMU DO PAMIĘCI

Postać zlecenia:

* M ADR

ZK -

Oznaczenia:

- ADR - adres komórki, której wartość chcemy zmienić lub adres komórki, od której chcemy wpisać program do pamięci
- ZK - aktualna zawartość komórki pamięci.

Zlecenie umożliwia wpisywanie wartości do komórek pamięci. Po wprowadzeniu zlecenia komputer wyświetli aktualną zawartość komórki pamięci, której adres został podany w zleceniu. Możemy wówczas wprowadzić nową wartość do komórki pamięci lub nie. Przejście do następnej komórki następuje po naciśnięciu klawisza spacji lub . W przypadku naciśnięcia spacji wartości poszczególnych komórek są wyświetlane w jednej linii, a w przypadku klawisza - jedna pod drugą. Aby wyjść ze zlecenia należy nacisnąć klawisz i DEBIT zgłasza się ponownie.

Przykład:

* M 6000

FF - 21
FF - 00
00 - 68
00 - 11
FF - ?

*

P - CZYTANIE LUB PISANIE SŁOWA DO PORTU

Postać zlecenia:

* P R AP

Oznaczenia:

R - czytanie portu
AP - adres portu

Zlecenie wyprowadza na ekran zawartość portu o adresie AP, po czym zgłasza się DEBIT.

Przykład:

* P R FF
7 F

*

Postać zlecenia:

* P W AP, BD

Oznaczenia:

W - pisanie do portu
AP - adres portu
BD - bajt danych wysyłanych do portu

Zlecenie wprowadza bajt danych BD do portu o adresie AP, po czym zgłasza się DEBIT.

Przykład:

* P W FE, 01

*

Po naciśnięciu powinien być słyszalny stuk, świadczący o "zapaleniu" bitu na porcie głośnikowym.

T – TESTOWANIE OBSZARU PAMIĘCI

Postać zlecenia:

* I ADRP, ADRK, KT **ENTER**

Oznaczenia:

- ADRP – adres początkowy obszaru testowanego
- ADRK – adres końcowy obszaru testowanego
- KT – krotność testowania obszaru

Zlecenie testuje wybrany obszar pamięci w ten sposób, że wpisuje i odczytuje do i z każdej komórki wartość 55, AA i młodszy bajt adresu. Jeżeli testowany obszar jest dobry, zgłasza się DEBIT, jeżeli nie jest dobry – z lewej strony ekranu pojawia się adres komórki, jej wartość i różnica w stosunku do wartości wpisywanej. Po wypisaniu na ekran wszystkich niesprawnych komórek ponownie zgłasza się DEBIT.

Przykład:

Aby naocznie przekonać się jak przebiega test pamięci można przetestować obszar pamięci ekranu.

* 3C00, 3FFF, 10 **ENTER**

X – WYŚWIETLENIE I ZMIANA ZAWARTOŚCI REJESTRÓW PROCESORA

Postać zlecenia:

* X **ENTER**

Zlecenie wyświetla zawartość akumulatora (A), rejestrów bocznych (B, C, D, E, H, L), rejestru flag (F), adres pamięci wskazywany przez rejestr HL (M), wskaźnik stosu (S) i licznik rozkazów (P).

Przykład:

* X **ENTER**

A=C9 B=00 C=00 D=33 E=6E F=42 H=7F L=D1 M=7FD1 S=7FB1
P=336E

*

Postać zlecenia:

* X R ZR –

Oznaczenia:

- R – symbol rejestru
- ZR – aktualna zawartość rejestru

Zlecenie umożliwia zmianę zawartości rejestrów począwszy od rejestru R (oznaczenie symboliczne rejestru) do rejestru P. Zawartość rejestru można modyfikować lub nie. Przejście do następnego rejestru odbywa się przez

naciśnięcie klawisza spacji lub **ENTER** . W przypadku naciśnięcia spacji wartości poszczególnych rejestrów są wyświetlane w jednej linii, a w przypadku klawisza **ENTER** – jedna pod drugą. Aby ponownie wejść do DEBIT-u należy przejść przez wszystkie rejestry lub nacisnąć klawisz **BREAK**

Przykład:

```
* X D 33-00 ENTER  
6E-00 ENTER  
4Z- ENTER  
7F- ENTER  
D1- ENTER  
7F D1 - ENTER  
7F B1 - ENTER  
336E - ENTER
```

*

4. Wyjście z programu "DEBIT"

Aby wyjść z DEBIT-u należy nacisnąć klawisz **NMI** .

Produkowane od sierpnia 1985 roku mikrokomputery MERITUM przystosowane są do współpracy ze stacją dysków elastycznych. W związku z tym w EPROM-ie nr 6 w miejscu programu DEBIT znajduje się program komunikacji ze stacją dysków.

STANISŁAW PŁACZEK
OETO POLITECHNIKI ŚLĄSKIEJ

ZASTOSOWANIE MERITUM-I W CHARAKTERZE STEROWNIKA DIGIGRAFU

WSTĘP

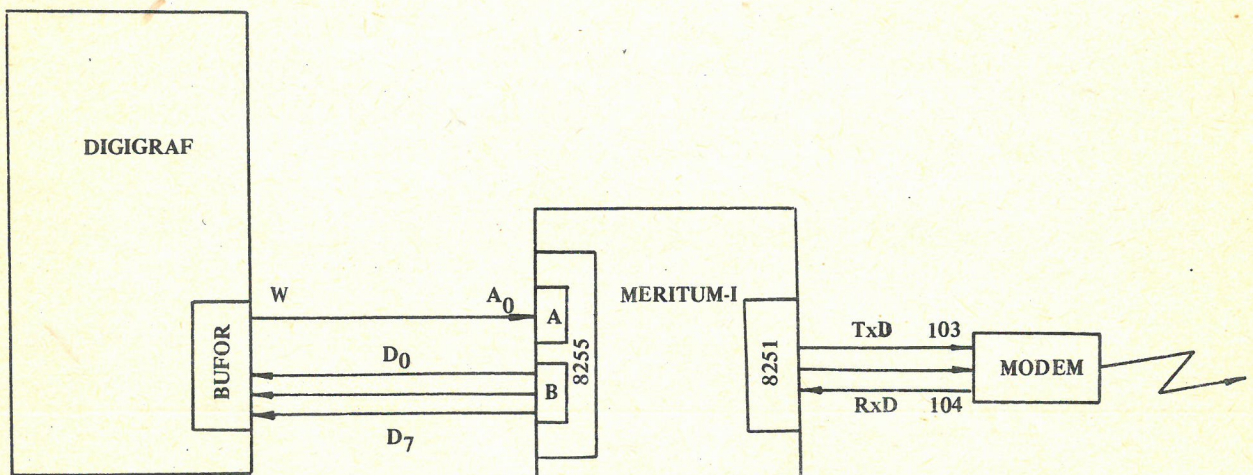
Analizując uważnie schemat ideowy komputera osobistego MERITUM-I można zauważyć, iż jest on wyposażony w standardowe moduły sprzęgające:

- moduł transmisji szeregowej typu INTEL-8251 wraz z wbudowanymi nadajnikami i odbiornikami spełniającymi standard interfejsu szeregowego V-24;
- równoległy programowalny moduł typu INTEL-8255. Uwzględniając zajęcie części bitów portu C, port A można wykorzystać jako wejściowy, a port B jako wyjściowy. Porty te mogą pracować zarówno w trybie 0 jak i w trybie 1;
- programowalny dzielnik częstotliwości typu INTEL-8253.

Reasumując, wszystkie wyżej wymienione moduły mogą być wykorzystywane w dowolny sposób, np.:

- element – 8251 w połączeniu z programowalnym dzielnikiem częstotliwości – 8253 może współpracować nie tylko z drukarką, ale również z urządzeniem do przesyłania sygnałów na odległość – modemem,
- element – 8255 może służyć do podłączenia czytnika taśmy papierowej, drukarki (DZM-180, D-100, ROBOTRON) jak również porty A i B mogą razem sterować bardziej złożonym obiektem.

Na rys. 1, pokazano sposób podłączenia komputera osobistego MERITUM-I do urządzenia kreślącego typu DIGIGRAF – z jednej strony, a z drugiej do modemu, poprzez który realizuje się komunikację z dużym systemem komputerowym, typu ODRA 1305, pracującym pod systemem operacyjnym GEORGE-3.



Rys. 1. Podłączenie MERITUM-I do DIGIGRAF-u i MODEM-u

2. CHARAKTERYSTYKA DIGIGRAFU

DIGIGRAF – nazwa automatycznego urządzenia kreślącego, składa się ze stołu kreślącego we współrzędnych X, Y oraz elektroniki sterującej. Urządzenie to może pracować automatycznie lub też może być podłączone do systemu komputerowego. Zintegrowana głowica kreśląca pozwala na zamocowanie czterech różnokolorowych pisaków, lub urządzenia frezującego, które umożliwia bezpośrednie modelowanie różnych form w materiale.

Programowanie DIGIGRAFU odbywa się przy pomocy instrukcji dla interpolatora. Zestaw instrukcji tworzy blok, który musi być zakończony nową linią (LF).

Dla przykładu:

- N001 – instrukcja numeru bloku 001
- GO1 – instrukcja liniowej interpolacji
- X + 000100 – instrukcja przyrostu po osi X o + 100 jednostek
- Y – 000500 – instrukcja przyrostu po osi Y o – 500 jednostek
- I + 001000 – instrukcja centrum okręgu po osi X
- J + 000600 – instrukcja centrum okręgu po osi Y
- F8 – instrukcja szybkości przesuwu głowicy
- Q01 – instrukcja wyboru pisaka i rodzaju linii

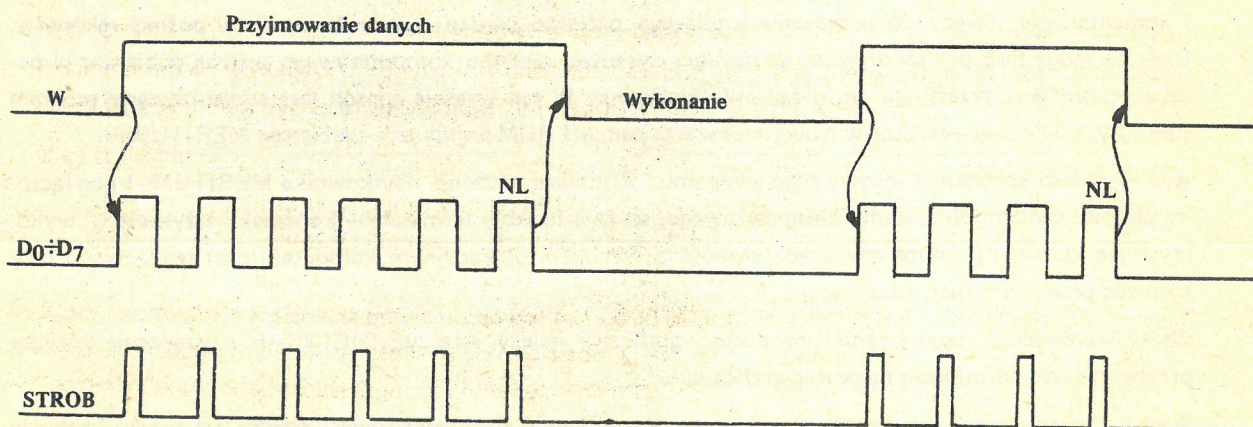
Łącząc wyżej wymienione instrukcje otrzymujemy blok sterujący. Przykładowy format bloku: N001GO 1X + 001000Y + 000100 F8Q01 (LF).

Standardowo DIGIGRAF wyposażony jest w czytnik taśmy papierowej, z której dane z zakodowanym rysunkiem, są wczytywane do urządzenia kreślącego. Chcąc jednak wyeliminować ten bardzo kłopotliwy nośnik informacji sięgnięto po MERITUM–I.

3. TECHNICZNA REALIZACJA PODŁĄCZENIA MERITUM–I DO DIGIGRAFU

Adaptując MERITUM–I do spełnienia funkcji sterownika urządzenia kreślącego, należy zapewnić całkowitą identyczność starego i nowego sposobu przesyłania danych do urządzenia.

Na rys. 2 pokazano w jaki sposób DIGIGRAF pobiera dane do swojego bufora i steruje głowicą. Sygnał "W" jest żądaniem przesłania bloku sterującego. W tym stanie MERITUM–I wysyła poszczególne bajty informacji wraz z sygnałem strobu, wpisując je do bufora DIGIGRAFU. Po przesłaniu bajtu nowej linii (LF), DIGIGRAF zdejmuję linię "W", tj. W=0 i przechodzi do fazy wykonania. MERITUM–I czeka z wysłaniem kolejnego bajtu do czasu wysterowania przez DIGIGRAF linii "W".



Rys. 2. Blok sterujący

W MERITUM–I wykorzystuje się dwa porty A i B w programowalnym module sprzęgającym 8255. Port A pracuje w trybie 0 – jako wejście (na Ao podaje się sygnał W), a port B – w trybie 1 – jako wyjście. Żądane tryby pracy realizuje się przez wpisanie do rejestru sterującego RS modułu 8255 słowa sterującego:

LD A, 149 – do akumulatora wpisujemy binarnie 10010101
OUT (247), A – wysłanie zawartości akumulatora do rejestru RS

Badanie stanu logicznego sygnału W (tj. bitu Ao) oraz wysyłanie kolejnych bajtów realizuje się programowo. Wykorzystując w podobny sposób moduł transmisji szeregowej 8251 oraz programowalny dzielnik częstotliwości 8253, można zrealizować techniczne podłączenie komputera osobistego MERITUM–I do modemu. Sekwencja programująca może być następująca dla portu 8251:

LD A,64 – wykonanie zerowania
OUT (253),A

LD A,251 – programowanie rejestru RS
OUT (253),A

LD A,7 – odblokowanie nadajnika i odbiornika układu 8251
OUT (253),A

4. STRUKTURA PROGRAMU STERUJĄCEGO

Jak już podkreśliłem, MERITUM–I można wykorzystać do wysyłania sekwencji sterujących – danych do DIGIGRAFU. Dane do kreślenia można otrzymać z trzech źródeł, a mianowicie:

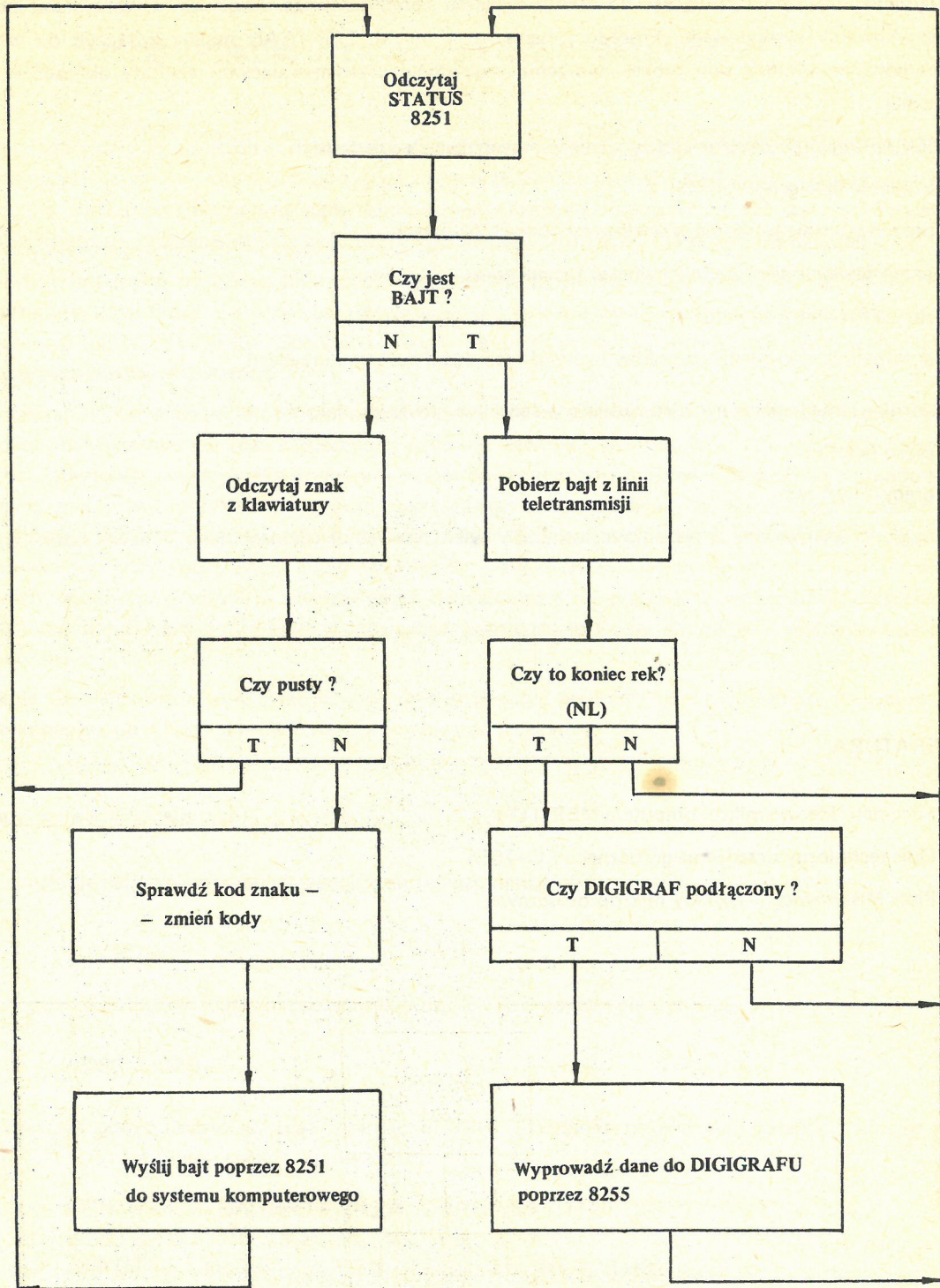
- a) programu, który może być napisany w BASIC-u lub assamblerze Z–80. Przyjmując dane bezpośrednio z klawiatury można bezpośrednio kreślić różne funkcje czy też krzywe,
- b) taśmy magnetofonowej. Dane te można przy pomocy prostego programu wysyłać do urządzenia kreślącego
- c) Bazy Danych dużego systemu komputerowego, poprzez łącza teletransmisji danych.

Pierwsze dwa sposoby można wykorzystać do kreślenia prostych figur i zależności ze względu na ograniczoną pamięć operacyjną komputera. Chcąc kreślić duże, skomplikowane rysunki z odpowiednimi napisami i komentarzami, należy do urządzenia kreślącego przestać bardzo dużo informacji w postaci rekordów. Dane te mogą być przechowywane w pamięci dyskowej systemu komputerowego skąd są pobierane w postaci rekordów i przesyłane do urządzenia kreślącego. W ten właśnie sposób jest zorganizowany program sterujący, który stale rezyduje w wolnym obszarze pamięci ROM komputera osobistego MERITUM–I.

Rys. 3 pokazuje schemat ideowy tego programu. Z punktu widzenia użytkownika MERITUM–I podłączony poprzez modem do systemu komputerowego, spełnia funkcję terminala – końcówki. Użytkownik wykorzystując klawiaturę, może nawiązać łączność z systemem operacyjnym komputera oraz realizować i nadzorować przebieg swoich zadań.

Chcąc wprowadzić dane (wyniki) w postaci graficznej należy włączyć DIGIGRAF i urządzenie kreślące przetwarza nam informację na postać graficzną.

Proces kreślenia można w każdej chwili przerwać, poprawić dane wejściowe i powtórzyć proces kreślenia od początku.



Rys. 3. Schemat blokowy programu sterującego

5. ZASTOSOWANIE

Podłączenie urządzenia kreślącego do dużego systemu komputerowego umożliwia wykorzystanie bogatego oprogramowania systemowego. Procedury napisane w języku FORTRAN można dołączyć do własnych programów, co ułatwia sam proces tworzenia programów przedstawiających rezultaty obliczeń w formie graficznej.

DIGIGRAF znajduje szerokie zastosowanie w następujących dziedzinach:

- a) kreślenie map geodezyjnych
- b) opracowywanie połączeń w wielowarstwowych pakietach
- c) przedstawienie sieci wentylacyjnej w formie grafu
- d) rozwiązywanie kratownic
- e) automatyczne kreślenie rysunków technicznych detali maszyn i urządzeń.

Zaopatrując zintegrowaną głowicę rysującą w specjalne przyrządy, jak:

- pióro świetlne
- precyzyjny frez

można wyżej wymienione urządzenie zastosować w wielu nowych dziedzinach nauki, a nawet sztuki.

6. LITERATURA

1. Schematy ideowe mikrokomputera MERITUM-I
2. Opis techniczny urządzenia graficznego EC-7054
3. Piotr Misiurewicz – Układy mikroprocesorowe.

KRYSTIAN BARTNICZEK
ZUK MERA—ELZAB

MIKROKOMPUTER NA LEKCJACH MATEMATYKI

Coraz częściej spotykany w szkołach mikrokomputer może być wykorzystany jako doskonała pomoc naukowa na lekcjach matematyki. Czytelnik, który zetknął się już z tym urządzeniem zapewne miał okazję poznać m.in. programy dydaktyczne, gry bądź programy rozwiązujące skomplikowane zadania matematyczne. Do celów szkolnych bardziej przydatne byłyby jednak programy mające związek z zagadnieniami poruszonymi w programie nauczania.

Niniejszy artykuł jest próbą pokazania kilku problemów, które można doskonale rozwiązać bądź zilustrować przy pomocy mikrokomputera. Programy, które przytoczono w tekście, napisane są w języku BASIC w wersji używanej na mikrokomputerach MERITUM lub ZX—Spectrum. Po nieznacznych przeróbkach mogą one być również uruchomione na innych typach mikrokomputerów.

Dodatkową zaletą tych programów jest ich zwięzłość. Pozwala to na bezpośrednie wprowadzenie programu z klawiatury do pamięci mikrokomputera (bez konieczności używania zewnętrznych nośników informacji np. kasety). Może to być dla początkującego użytkownika doskonałą okazją do przełamania oporów w zetknięciu z komputerem. Ponadto programy te stanowią przykład zwięzłego i oszczędnego programowania, co jest szczególnie ważne dla urządzeń o małej pojemności pamięci operacyjnej. Zwięzłość tę osiąga się m.in. poprzez rezygnację z nadmiaru komentarzy i komunikatów, odpowiedni dobór nazw zmiennych, używanie podprogramów itp. W związku z niepełnym zabezpieczeniem przed wprowadzaniem niewłaściwych danych, może się zdarzyć, iż wprowadzając "złośliwie" dobrane wartości, np. ułamki lub liczby ujemne tam, gdzie wymagane są liczby naturalne — wykonywanie programu zostanie przerwane.

Jako ćwiczenie Czytelnik może spróbować samodzielnie wyposażyć programy w pełną diagnostykę wprowadzanych danych i przekonać się do jakiego stopnia zwiększa to objętość programu.

1. NAJMNIEJSZA WSPÓLNA WIELOKROTNOŚĆ

Program znajduje najmniejszą wspólną wielokrotność liczb naturalnych:

$$L_1, L_2, L_3, \dots, L_n \quad (n \geq 2)$$

Po uruchomieniu zleceniem RUN i naciśnięciu klawisza ENTER pojawia się pytanie:

ILOŚĆ LICZB ?

należy wówczas podać liczbę "n" oraz nacisnąć ENTER. Następnie pojawią się pytania o kolejne liczby L_i ($i = 1, 2, \dots, n$), które należy kolejno wprowadzać:

```
10 REM ----- NAJMNIEJSZA WSPOLNA WIELOKROTNOŚĆ -----
15 DIM L(200): DEFDBL W,L,A
20 INPUT " ILOSC LICZB ";N: IF N<2 THEN 20
25 FOR I=1 TO N
30 PRINT USING "   ###   LICZBA ";I: INPUT L(I): NEXT I
40 A=L(1): FOR I=2 TO N: GOSUB 1000: A=W: NEXT I
50 PRINT " NAJMNIEJSZA WSPOLNA WIELOKROTNOŚĆ = ";W
60 GOTO 20
1000 IF A<L(I) THEN 1030
1010 W=A: A=L(I): L(I)=W
1030 W=L(I): FOR J=1 TO A
1040 IF W=A*INT(W/A) THEN 1100
1050 W=W+L(I): NEXT J
1100 RETURN
```

Program ten można wykorzystać dla obliczenia wspólnego mianownika przy dodawaniu i odejmowaniu ułamków.

2. DZIELENIE WIELOMIANÓW

Po uruchomieniu programu (RUN < ENTER >) pojawia się pytanie o stopień dzielnej i dzielnika. Należy podać dwie liczby:

n, m

oddzielone przecinkiem i nacisnąć ENTER. Liczby te muszą spełniać nierówność $n \geq m \geq 0$. Następnie pojawia się pytania o kolejne współczynniki dzielnej i dzielnika:

```
10 REM ----- DZIELENIE WIELOMIANOW -----
15 REM      K.BARTNICZEK 1985.09.17
20 CLEAR 2000: DIM A(100),D(100),Q(100),R(100)
80 CLS: INPUT " PODAJ STOPNIEN DZIELNEJ I DZIELNIKA ";N,M
85 IF M>N OR M<0 THEN 80
90 E$="EJ": GOSUB 500: E$="A(###)=": K=1: P0=N: GOSUB 600
100 E$="IKA": GOSUB 500: E$="D(###)=": K=2: P0=M: GOSUB 600
110 REM ----- DZIELENIE -----
150 FOR I=0 TO N:S=0:Z=SGN(INT((I/(N-M+1)))): G=I-1+Z*(N-M+1-I)
160 IF G<0 THEN 175 ELSE FOR J=0 TO G
170 S=S+Q(J)*D(I-J): NEXT J
175 ON Z+1 GOTO 180,190
180 Q(I)=(A(I)-S)/D(0): GOTO 200
190 R(N-I)=A(I)-S
200 NEXT I
220 REM ----- W Y N I K I -----
350 CLS: PRINT " I L O R A Z = ";
360 FOR I=0 TO N-M: PRINT Q(I);: NEXT I: PRINT
370 PRINT " R E S Z T A = ";
380 IF M<1 THEN 400 ELSE FOR I=M-1 TO 0 STEP-1: PRINT R(I);
390 NEXT I: PRINT
400 IF INKEY$="" THEN 400 ELSE GOTO 20
490 REM ----- PODPROGRAMY -----
500 PRINT "WPROWADZ KOLEJNO WSPOLCZYNNIKI DZIELN"+E$
510 PRINT "POCZAWSZY OD NAJWIEKSZEJ POTEGI": RETURN
600 FOR I=0 TO P0: PRINT USING E$;P0-I;
610 IF K=2 THEN 650 ELSE INPUT A(I)
630 IF A(0)=0 THEN 700 ELSE GOTO 670
650 INPUT D(I): IF D(0)=0 THEN 700 ELSE GOTO 670
670 NEXT I: RETURN
700 PRINT " MUSI BYC ROZNY OD ZERA !": GOTO 600
```

Po zakończeniu pracy programu otrzymuje się współczynniki ilorazu oraz reszty z dzielenia. Naciśnięcie klawisza ENTER powoduje powrót do pytania o nowe dane.

Jako ćwiczenie proponuje się Czytelnikowi napisać analogiczny program na mnożenie dwóch wielomianów.

3. PIERWIĄSTEK RÓWNANIA $f(x) = 0$

Przed uruchomieniem programu należy w linii 1000 wpisać funkcję $f(x)$, której pierwiastek chcemy wyliczyć:

1000 Y (I) = wyrażenie reprezentujące funkcję $f(x)$

Po uruchomieniu należy podać końce przedziału, wewnątrz którego znajduje się szukany pierwiastek. W pewnych przypadkach (np. jeśli wartości funkcji na końcach przedziału są tych samych znaków) program ponawia pytanie o końce przedziału.

W poniższym programie w linii 1000 znajduje się wyrażenie reprezentujące funkcję: $x^3 - 5x^2 - 3x + 15$

```

10 REM ----- MIEJSCA ZEROWE FUNKCJI -----
15 DEFDBL A,X,Y
20 PRINT: INPUT "PODAJ KONCE PRZEDZIALU";A(0),A(1)
30 A(2)=(A(0)+A(1))/2: FOR I=0 TO 2: X=A(I)
40 GOSUB 1000: IF ABS(Y(I))<2E-14 THEN 1200 ELSE NEXT I
45 FOR I=0 TO 1: IF Y(I)*Y(2)<0 THEN 60
50 NEXT I: GOTO 20
60 A(1-I)=A(2): GOTO 30
1000 Y(I)=X*X*X-5*X*X-3*X+15
1010 RETURN
1200 PRINT USING " X = ##.#####E" X: GOTO 20
    
```

Po zakończeniu pracy programu wyświetla się wartość pierwiastka równania $f(x) = 0$ oraz pojawia się zapytanie o nowe końce przedziału. Można w ten sposób policzyć inny pierwiastek tego samego równania.

Warto zwrócić uwagę na zwięzłą budowę programu, którą osiąga się dzięki użyciu zmiennych tablicowych A (I), Y (I) (linie programu 40, 45, 60) oraz wywoływaniu podprogramu (linia 1000).

4. PROGRAM KOMBINATORYCZNY GENERUJĄCY WARIACJE WZGLĘDNI PERMUTACJE

Po uruchomieniu programu pojawia się pytanie o liczby n, k ($n \geq k \geq 0$), gdzie n oznacza ilość elementów zbioru, k – ilość elementów podzbioru. Program wyświetla na ekranie wszystkie permutacje k -elementowych podzbiorów zbioru n -elementowego (tzw. wariacje). W przypadku gdy $k = n$ otrzymuje się n -elementowe permutacje. Po zakończeniu wyświetla się ilość wszystkich wariacji, która wynosi: $\binom{n}{k} k!$

```

5 REM ----- GENERATOR WARIACJI ---- REKURSJA ----
10 REM          K.BARTNICZEK, 07.08.1985
15 CLEAR 1000: DIM X(100)
20 INPUT "PODAJ N,K";N,K: K1=K
25 IF K<0 OR K>N THEN 20
30 L=0: GOSUB 100
40 PRINT " ILOSC WARIACJI (PERMUTACJI) = ";L
50 GOTO 15
90 REM ----- PODPROGRAM -----
100 IF K=0 THEN 300
110 X(K)=1
120 IF X(K)>N THEN 350
130 IF N<K+1 THEN 160 ELSE FOR I=N TO K+1 STEP-1
140 IF X(I)=X(K) THEN 200
150 NEXT I
160 K=K-1: GOSUB 100: K=K+1
200 X(K)=X(K)+1
210 GOTO 120
300 IF K1<1 THEN 320 ELSE FOR I=K1 TO 1 STEP-1
310 PRINT " "+CHR$(64+X(I));:NEXT I: PRINT
320 L=L+1
350 RETURN
    
```

Jako ćwiczenie dla Czytelnika proponuje się napisanie analogicznego programu generującego wszystkie k-elementowe podzbiory zbioru n-elementowego (tzw. kombinację).

Warto jeszcze zwrócić uwagę na konstrukcję tego programu. Zastosowano tu tzw. rekursywne odwołanie do podprogramu, tzn. odwołanie się wewnątrz treści podprogramu do niego samego. Co to znaczy?

Wiadomo, że w programie głównym można odwołać się do podprogramu instrukcją GOSUB nr – linii. Fragment podprogramu musi być zakończony instrukcją RETURN. Taką technikę zastosowano w programie 1, gdzie w "pętli":

```
FOR I = 2 TO N          (linia 40)
```

następuje wielokrotne wywoływanie podprogramu (zawartego w liniach 1000 – 1100).

Również z każdego podprogramu można wywołać inne podprogramy (jest to tzw. zagnieżdżenie). Należy tylko przestrzegać zasady, aby zmienne używane w (pod)programie wywołującym nie kolidowały ze zmiennymi używanymi w podprogramie wywoływanym.

Z rekursją mamy do czynienia wówczas, gdy podprogram wywoływany jest identyczny z podprogramem wywołującym (lub mówiąc prościej: podprogram wywołuje sam siebie). W naszym przykładzie podprogramem są linie 100 – 350. Wewnątrz treści podprogramu w linii 160 następuje rekursywne wywołanie podprogramu (GOSUB 100).

Istnieją pewne zagadnienia programistyczne, które bez użycia tej techniki nie da się rozwiązać. Czytelnik może spróbować podać taki przykład.

5. ROZKŁAD DWUMIANOWY (BERNOULLIEGO)

Program generuje sumy:

$$X_1 + X_2 + \dots + X_n \quad (1)$$

zmiennych losowych, gdzie X_i ($i = 1, 2, \dots, n$) przyjmuje wartość 1 z prawdopodobieństwem p oraz wartość 0 z prawdopodobieństwem $1-p$ ($0 < p < 1$).

Po uruchomieniu programu należy podać liczbę $n \leq 62$ (długość serii) oraz podać prawdopodobieństwo sukcesu p .

Kropki w dolnej poziomej linii reprezentują liczby całkowite z przedziału $< 0, n >$ (bo tylko takie wartości może przyjmować suma (1)). W prawym górnym rogu ekranu wyświetla się aktualna wartość licznika rejestrującego ilość wykonanych eksperymentów losowych polegających na losowaniu zmiennych tworzących sumę (1). Pionowy "słupek" stojący nad kropką wskazuje ile razy suma (1) przyjęła wartość reprezentowaną przez tę kropkę.

```
10 REM ----- BERNOULLI - POISSON -----
15 L=0: CLEAR 500: DIM Y(127)
20 CLS:INPUT " PODAJ DLUGOSC SERII "N: IF N>62 THEN 20
30 INPUT " PRAWDOPODOBNIENSTWO SUKCESU "P
35 W=10: REM - WSPOLCZYNNIK NORMUJACY
40 PRINT@896,"": FOR I=0 TO N: PRINT CHR$(46);: NEXT I
50 S=0: FOR I=1 TO N: S=S-(RND(O)<P): NEXT I
60 Y(S)=Y(S)+1: SET(2*S,47-Y(S)/W)
100 L=L+1: PRINT@58,USING"L=####"L;: GOTO 50
```

Program może służyć jako ilustracja , ojęcia rozkładu prawdopodobieństwa.

6. KRZYWA GAUSSA

Podobnie jak w poprzednim przykładzie program generuje sumy zmiennych losowych:

$$X_1 + X_2 + \dots + X_n \quad (2)$$

Zmienna X_i ($i = 1, 2, \dots, n$) przyjmuje wartości 1, 2, 3, 4, 5, 6 z prawdopodobieństwem 1/6. Eksperyment losowy można interpretować jako rzut n kostkami do gry i badanie sumy wylosowanych oczek. Kropki w dolnej linii poziomej oznaczają liczby naturalne z przedziału $\langle n, 6n \rangle$. Z uwagi na wymiary ekranu mikrokomputera musi być $n \leq 12$.

Po uruchomieniu programu po kilkunastu minutach otrzymuje się na ekranie wyraźny zarys krzywej Gaussa.

```
10 REM ----- KRZYWA GAUSSA -----
15 L=0: CLEAR 500: DIM T(127)
20 CLS: INPUT "ILOSC KOSTEK": N: IF N>12 THEN 20
25 W=SQR(N*17.5/6): REM - WSPOLCZYNNIK NORMUJACY
35 PRINT@896,"":FOR I=0 TO 5*N: PRINT CHR$(46);: NEXT I
40 S=0: FOR I=1 TO N: S=S+RND(6): NEXT I
50 T(S-N)=T(S-N)+1: SET(2*(S-N),47-T(S-N)/W)
60 L=L+1: PRINT@58,USING"L=####"L;: GOTO 40
```

Program 6 jest ilustracją znanego faktu, iż suma dużej ilości zmiennych losowych niezależnych ma w przybliżeniu rozkład Gaussa. Podobny rozkład ma np. wzrost obserwowany w dużych populacjach osobników tego samego gatunku lub wielkość błędu pomiaru obserwowana w trakcie dokonywania długiej serii pomiarów tej samej wielkości.

Przytoczone przykłady sugerują sposób wykorzystania mikrokomputera jako narzędzia służącego do ilustracji pewnych pojęć matematycznych wymagających od ucznia sporej dozy wyobraźni, a czasem także wykonania dużej ilości manipulacji na rekwizytach i na kartce papieru (por. programy 4 i 6). Mikrokomputer umożliwia w skróconym czasie komfortowe przeprowadzenie operacji oraz obserwację wyników na ekranie, a także wielokrotne powtarzanie tych samych eksperymentów przy zmieniających się danych wejściowych. Umożliwia to lepsze zrozumienie danego zagadnienia bez rozpraszania uwagi i energii na sprawach drugorzędnych i nieistotnych. Mając takie narzędzie można położyć większy nacisk na projektowanie eksperymentów i algorytmów, co posiada nowe, rzadko spotykane w dotychczasowej praktyce szkolnej walory dydaktyczne.

JÓZEF WAJSMAN
MIECZYŚLAW JAGODZIŃSKI
INSTYTUT AUTOMATYKI
POLITECHNIKA ŚLĄSKA

PROGRAM "WIELOMIAN"

Przedstawiony poniżej program WIELOMIAN wykonuje operacje dodawania, odejmowania, mnożenia i dzielenia dwóch wielomianów. Dodawanie i odejmowanie realizowane jest poprzez odpowiednie sumowanie współczynników wielomianów. Mnożenie i dzielenie oparte jest na wzorach rekurencyjnych zasięgniętych z podręcznika Romana Zuberka "Metody numeryczne i programowanie".

MNOŻENIE

Współczynniki iloczynu oblicza się według wzoru:

$$A_K = \sum_{J=\max(0, K-Q)}^{\min(P, K)} B_J C_{K-J}, \quad K = 0, 1, \dots, N \quad \text{gdzie: } N = P + Q$$

P — stopień pierwszego wielomianu

Q — stopień drugiego wielomianu

DZIELENIE

Współczynniki ilorazu i reszty oblicza się według wzoru:

ILORAZ —

$$A_K = \frac{1}{C_0} \left(B_K - \sum_{J=\max(0, K-Q)}^{K-1} A_J C_{K-J} \right), \quad K = 0, 1, \dots, N$$

RESZTA —

$$A_K = B_K - \sum_{J=\max(0, K-Q)}^n A_J C_{K-J}, \quad K = N + 1, N + 2, \dots, P$$

przy czym $N = P - Q$, $C = 0$

Program posiada pełne zabezpieczenie przed wprowadzeniem niewłaściwych danych oraz opisy błędów, które mogą wystąpić w trakcie wykonywania programu. Błędy sygnalizowane są na ekranie monitora poprzez wyświetlenie odpowiednich napisów informujących o rodzaju i miejscu błędu.

Wyniki operacji otrzymywane na ekranie, można uzyskać w formie wydruku zastępując instrukcję PRINT w zaznaczonych na listingu liniach programu instrukcją LPRINT.

OPIS ZMIENNYCH WEJŚCIOWYCH

P — stopień pierwszego wielomianu postaci:

$$B(0) \cdot x^P + B(1) \cdot x^{P-1} + \dots + B(P)$$

Q — stopień drugiego wielomianu postaci:

$$C(0) \cdot x^Q + C(1) \cdot x^{Q-1} + \dots + C(Q)$$

N — numer wariantu:

N = 1	DODAWANIE
N = 2	ODEJMOWANIE
N = 3	MNOŻENIE
N = 4	DZIELENIE

PO WYKONANIU OBLICZEŃ:

Wielomian wynikowy ma postać:

$$A(0) \cdot x^M + A(1) \cdot x^{M-1} + \dots + A(M)$$

DLA DODAWANIA M = MAX (P, Q)

DLA ODEJMOWANIA M = MAX (P, Q)

DLA MNOŻENIA M = P + Q

DLA DZIELENIA M = P - Q

oraz postać RESZTY:

$$A(M+1) \cdot x^{Q-1} + A(M+2) \cdot x^{Q-2} + \dots + A(P)$$

PRZYKŁADY

DODAWANIE

Dane:

P	=	3
Q	=	4
B(0)	=	1
B(1)	=	2
B(2)	=	3
B(3)	=	4
C(0)	=	1
C(1)	=	-1
C(2)	=	-2
C(3)	=	-3
C(4)	=	-4

Wyniki:

$$\begin{aligned} M &= 4 \\ A(0) &= 1 \\ A(1) &= 0 \\ A(2) &= 0 \\ A(3) &= 0 \\ A(4) &= 0 \end{aligned}$$

MNOŻENIE

Dane:

$$\begin{aligned} P &= 3 \\ B(0) &= 1 \\ B(1) &= 3 \\ B(2) &= -1 \\ B(3) &= 0 \\ Q &= 2 \\ C(0) &= 1 \\ C(1) &= 0 \\ C(2) &= -2 \end{aligned}$$

Wyniki:

$$\begin{aligned} M &= 5 \\ A(0) &= 1 \\ A(1) &= 3 \\ A(2) &= -3 \\ A(3) &= -6 \\ A(4) &= 2 \\ A(5) &= 0 \end{aligned}$$

DZIELENIE:

Dane:

$$\begin{aligned} P &= 3 \\ B(0) &= 4 \\ B(1) &= -2 \\ B(2) &= 1 \\ B(3) &= -1 \\ Q &= 2 \\ C(0) &= 1 \\ C(1) &= 2 \\ C(2) &= -2 \end{aligned}$$

Wyniki:

$$\begin{aligned} M &= 1 \\ A(0) &= 4 \\ A(1) &= -10 \end{aligned}$$

Reszta:

$$\begin{aligned} A(2) &= 29 \\ A(3) &= -21 \end{aligned}$$

LISTING PROGRAMU "WIELOMIAN"

```
2 CLS:PRINTCHR*(23):PRINT@468,"WIELOMIAN":FOR T=1 TO 900 :
  NEXT :CLS
10 CLS:PRINT"*****
  *****":PRINT
20 PRINT"          DODAWANIE, ODEJMOWANIE, MNOZENIE          "
21 PRINT"          LUB  DZIELENIE DWOCH WIELOMIANOW":PRINT
23 PRINT"*****
  *****"
24 DIM A(300),B(300),C(300),L(300),G(300)
25 ON ERROR GOTO 1700
40 INPUT "CZY PODAC DOKLADNY OPIS ZMIENNYCH(TAK/NIE)";
  PYT*
50 IF LEFT*(PYT*,1)="T" THEN PYT*=" ":GOSUB 740 ELSE PYT*=" "
55 CLS:PRINT"*****
  *****"
60 PRINT"DODAWANIE NUMER WARIANTU N=1"
70 PRINT"ODEJMOWANIE NUMER WARIANTU N=2"
80 PRINT"MNOZENIE NUMER WARIANTU N=3"
81 PRINT"DZIELENIE NUMER WARIANTU N=4"
86 PRINT"*****
  *****"
87 PRINT"PODAJ NUMER WARIANTU N=?"
88 INPUT N
89 IF N<1 OR N>4 THEN PRINT"WPROWADZILES NIEPOPRAWNA WARTOSC":
  GOTO 87
90 ON N GOSUB 99,1299,349,489
92 END
99 CLS:PRINT"*****
  *****"
100 PRINT"PROGRAM REALIZUJE DODAWANIE DWOCH WIELOMIANOW"
101 PRINT"*****
  *****"
130 GOSUB 910
140 PRINT"*****WYNIKI*****
  *****"
150 IF P=Q THEN 230
160 IF P>Q THEN 290
165 GOSUB 2300
170 FOR K=0 TO Q
180 A(K)=C(K)+B(Q-K)
190 PRINT"A(";K;")=";A(K)
191 IF CINT((K+1)/10)*10=K THEN GOSUB 1600
200 NEXT K
210 PRINT"*****
  *****"
220 RETURN
230 FOR K=0 TO P
240 A(K)=B(K)+C(K)
250 PRINT"A(";K;")=";A(K)
251 GOSUB 1600
260 NEXT K
270 PRINT"*****
  *****"
280 RETURN
290 GOSUB 2400
291 FOR K=0 TO P
300 A(K)=B(K)+C(P-K)
310 PRINT"A(";K;")=";A(K)
```

```
311 IF CINT((K+1)/10)*10=K THEN GOSUB 1600
320 NEXT K
330 PRINT"*****
*****"
340 RETURN
349 CLS:PRINT"*****
*****"
350 PRINT"*****
*****"
360 PRINT"PROGRAM REALIZUJE MNOZENIE DWOCH WIELOMIANOW"
361 PRINT"*****
*****"
370 GOSUB 910
380 PRINT"*****WYNIKI*****
*****"
390 FOR K=0 TO P+Q
400 A(K)=0
405 IF K<P THEN MIN=K ELSE MIN=P
410 IF K-Q>0 THEN MAX=K-Q ELSE MAX=0
420 FOR J=MAX TO MIN
430 A(K)=A(K)+B(J)*C(K-J)
440 NEXT J
450 PRINT "A(";K;")=";A(K)
451 IF CINT((K+1)/10)*10=K THEN GOSUB 1600
460 NEXT K
480 RETURN
489 CLS:PRINT"*****
*****"
490 PRINT"*****
*****"
499 CLS:PRINT"*****
*****"
500 PRINT"PROGRAM REALIZUJE DZIELENIE DWOCH WIELOMIANOW"
510 PRINT"*****
*****"
520 GOSUB 910
555 PRINT"*****WYNIKI*****
*****"
556 N=P-Q
560 FOR K=0 TO P-Q
570 A(K)=0
580 IF K-Q>0 THEN MAX=K-Q ELSE MAX=0
585 A1K=0
590 FOR J=MAX TO K-1
600 A1K=A(J)*C(K-J)+A1K
610 NEXT J
611 A(K)=(B(K)-A1K)/C(0)
620 PRINT"A(";K;")=";A(K)
621 IF CINT((K+1)/10)*10=K THEN GOSUB 1600
630 NEXT K
640 PRINT"*****RESZTA*****WIELOMIANU*****
*****"
650 FOR K=N+1 TO P
655 A(K)=0
660 IF K-Q>0 THEN MAX=K-Q ELSE MAX=0
665 A1=0
670 FOR J=MAX TO N
680 A1=A(J)*C(K-J)+A1
```

```
690 NEXT J
691 A(K)=B(K)-A1
700 PRINT "A(" ; K ; ")=" ; A(K)
701 IF CINT((K+1)/10)*10=K THEN GOSUB 1600
710 NEXT K
720 PRINT "*****"
    *****
730 RETURN
740 CLS:PRINT "*****"
    *****
745 PRINT "          OPIS ZMIENNYCH WEJSCIOWYCH:"
750 PRINT "P-STOPIEN PIERWSZEGO WIELOMIANU POSTACI:
    B(0)*XC^P+B(1)*XC^(P-1)+...+B(P)"
770 PRINT "Q-STOPIEN DRUGIEGO WIELOMIANU POSTACI:
    C(0)*XC^Q+C(1)*XC^(Q-1)+...+C(Q)"
790 PRINT "*****"
    *****
800 PRINT "WIELOMIAN WYNIKOWY MA POSTAC:
    A(0)*XC^M+A(1)*XC^(M-1)+...+A(M)"
801 PRINT "DLA DODAWANIA M=MAX(P,Q)"
802 PRINT "DLA ODEJMOWANIA M=MAX(P,Q)"
803 PRINT "DLA MNOZENIA M=P+Q"
804 PRINT "DLA DZIELENIA M=R-Q, POSTAC RESZTY JEST NASTEPUJACA:
    A(M+1)*XC^(Q-1)+A(M+2)*XC^(Q-2)+...+A(P)"
805 PRINT "*****"
870 INPUT "CZY POWTORZYC OPIS ZMIENNYCH (TAK/NIE)";PYT%
880 IF LEFT$(PYT%,1)="T" THEN PYT%=" ":GOTO 740 ELSE PYT%=" ":
    RETURN
881 PRINT "*****"
890 PRINT "WPROWADZANIE DANYCH"
910 PRINT "P STOPIEN PIERWSZEGO WIELOMIANU"
920 INPUT "P=?";P
935 IF P>0 THEN 938
936 PRINT "WPROWADZILES NIEPOPRAWNA WARTOSC, P MUSI BYC
    WIEKSZE OD 0"
937 GOTO 1234
938 PRINT "Q STOPIEN DRUGIEGO WIELOMIANU"
939 INPUT "Q=?";Q
940 IF Q>0 THEN 943
941 PRINT "WPROWADZILES NIEPOPRAWNA WARTOSC, Q MUSI BYC WIEKSZE OD 0
942 GOTO 1242
943 IF N<>4 THEN 949
944 IF P>=Q THEN 949
945 PRINT "WPROWADZILES NIEPOPRAWNA WARTOSC, P MUSI BYC
    WIEKSZE LUB ROWNE Q"
946 GOTO 920
949 FOR K=0 TO P
950 PRINT "B(" ; K ; ")=" ;
960 INPUT B(K)
968 NEXT K
969 PRINT "WPROWADZ WSPOLCZYNNIKI DRUGIEGO WIELOMIANU"
970 FOR K=0 TO Q
971 PRINT "C(" ; K ; ")=" ;
972 INPUT C(K)
973 NEXT K
1183 PRINT "*****DANE*****WPROWADZILES*****POPRAWNIE***
    *****"
1184 PRINT "*****"
    *****
1185 INPUT "CZY CHCESZ SPRAWDZIC WPROWADZONE WSPOLCZYNNIKI
    (TAK/NIE)";PYT%
```

```
1186 IF LEFT$(PYT$,1)="N" THEN RETURN
1187 FOR K=0 TO P
1189 PRINT"B(";K;")=";B(K)
1191 PRINT"CZY CHCESZ POPRAWIC (TAK/NIE)";
1193 INPUT PYT$
1195 IF LEFT$(PYT$,1)="N" THEN GOTO 1199
1197 INPUT"NOWA WARTOSC=";B(K)
1199 NEXTK
1200 FORK=0 TO Q
1202 PRINT"C(";K;")=";C(K)
1204 PRINT"CZY CHCESZ POPRAWIC (TAK/NIE)";
1206 INPUTPYT$
1208 IFLEFT$(PYT$,1)="N" THEN GOTO 1216
1210 INPUT"NOWA WARTOSC=";C(K)
1216 NEXT
1229 GOTO1184
1234 INPUT"NOWA WARTOSC=";P
1235 GOTO 935
1242 INPUT"NOWA WARTOSC=";Q
1243 GOTO 940
1299 CLS:PRINT"*****
*****"
1300 PRINT"*****
*****"
1310 PRINT"PROGRAM REALIZUJE ODEJMOWANIE DWOCH WIELOMIANOW"
1320 PRINT"*****
*****"
1330 GOSUB 910
1340 PRINT"*****WYNIKI*****
*****"
1350 IF P=Q THEN GOTO 1490
1360 IF P>Q THEN GOTO 1430
1365 GOSUB 2300
1370 FORK=0 TO Q
1380 A(K)=B(Q-K)-C(K)
1390 PRINT"A(";K;")=";A(K)
1391 IF CINT((K+1)/10)*10=K THEN GOSUB1600
1400 NEXT K
1410 PRINT"*****
*****"
1420 RETURN
1430 GOSUB 2400
1431 FORK=0 TO P
1440 A(K)=B(K)-C(P-K)
1450 PRINT"A(";K;")=";A(K)
1451 IF CINT((K+1)/10)*10=K THEN GOSUB 1600
1460 NEXT K
1470 PRINT"*****
*****"
1480 RETURN
1490 FORK=0 TO P
1500 A(K)=B(K)-C(K)
1510 PRINT"A(";K;")=";A(K)
1511 IF CINT((K+1)/10)*10=K THEN GOSUB 1600
```

```
1520 NEXT K
1530 PRINT"*****
*****"
1550 RETURN
1600 U*=INKEY*:U*=INKEY*
1610 PRINT@1014,"<<ENTER>>";:FOR T=1 TO 200:NEXT
1620 U*=INKEY*:IF U*("<") THEN PRINT@1014," "
PRINT@959,:RETURN
1630 PRINT@1014," " :;FOR T=1 TO 200:NEXT
1640 U*=INKEY*:IF U*("<") THEN PRINT@959,:RETURN ELSE 1600
1700 CLS:PRINT:PRINT:PRINT
1701 PRINT"!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!"
1702 PRINT"          BLAD WYKONANIA NR"ERR/2+1
1703 PRINT"          W LINII NR"ERL
1704 ON ERR/2+1 GOTO 1710,1720,1730,1740,1750,1760,1770,1780,
1790,1800,1810,1820,1830,1840,1850,1860,1870,1880,1890,
1900,1910,1920
1710 PRINT"NF-INSTRUKCJA NEXT UZYTA BEZ FOR":GOTO2000
1720 PRINT"SN-BLAD SKLADNI":GOTO 2000
1730 PRINT"RG-RETURN BEZ GOSUB":GOTO 2000
1740 PRINT"OD-BRAK DANYCH":GOTO 2000
1750 PRINT"FC-NIEDOZWOLONE WYWOLANIE FUNKCJI":GOTO 2000
1760 PRINT"OV-LICZBA ZA DUZA LUB ZA MALA":GOTO 2000
1770 PRINT"OM-BRAK PAMIECI":GOTO 2000
1780 PRINT"UL-BRAK LINII":GOTO 2000
1790 PRINT"BS-ZLY WSKAZNIK":GOTO 2000
1800 PRINT"DD-PONOWNA DEKLARACJA":GOTO 2000
1810 PRINT"/0-DZIELENIE PRZEZ ZERO":GOTO 2000
1820 PRINT"ID-ZLE UZYCIE INPUT":GOTO 2000
1830 PRINT"TM-WARTOSC LANCUCHOWA POROWNYWANA Z WARTOSCIA
NIELANCUCHOWA":GOTO 2000
1840 PRINT"OS-BRAK PAMIECI DLA LANCUCHA":GOTO 2000
1850 PRINT"LS-ZA DLUGI LANCUCH":GOTO 2000
1860 PRINT"ST-SKOMPLIKOWANA OPERACJA":GOTO 2000
1870 PRINT"CN-KONIEC PROGRAMU":GOTO 2000
1880 PRINT"NR-BRAK RESUME":GOTO 2000
1890 PRINT"RW-RESUME BEZ ON ERROR GOTO":GOTO 2000
1900 PRINT"UE-ZLY PARAMETR ERROR":GOTO 2000
1910 PRINT"MO-BRAK PARAMETRU":GOTO 2000
1920 PRINT"FE-ZLE DANE Z TASMY":GOTO 2000
2000 PRINT"          PROGRAM PRZERWANY"
2100 PRINT:PRINT"!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!":END
2300 FOR K=0 TO P
2310 L(K)=B(P-K)
2320 NEXT K
2330 FOR K=0 TO P
2340 B(K)=L(K)
2350 NEXT K
2360 RETURN
2400 FOR K=0 TO Q
2410 G(K)=C(Q-K)
2420 NEXT K
2460 RETURN
```

MACIEJ DEPTUCH
ZUK MERA-ELZAB

SPROSTOWANIE DO ARTYKUŁU Z Nr 2 BIULETYNU

Wylimitowanie rozbłysków ekranu w mikrokomputerze Meritum--I

Z przykrością informuję, że do mojego artykułu zamieszczonego w nr 2 Biuletynu, opisującego przeróbki eliminujące rozbłyski ekranu, wkradły się błędy. Poniżej przedstawiono prawidłowy opis koniecznych modyfikacji.

W celu wykonania modyfikacji należy:

- rozciąć połączenia
 - M 63/6 – M 63/10 (cięcie po stronie lutowania pod elementem M 63)
 - M 40/4 – M 68/6, M 77/2 (jedno cięcie po stronie lutowania pod elementem M68)

- wykonać połączenia dodatkowe
 - M 63/6 – M 48/1
 - M 50/1 – M 48/2
 - M 48/3 – M 63/10
 - M 33/8 – M 68/6, M 77/2

Za dopuszczenie do druku mylnych informacji serdecznie przepraszam.

PROPOZYCJE I OFERTY

Pierwotnym założeniem Zespołu Redakcyjnego było, aby Biuletyn był otwarty dla wszystkich jego czytelników i dla każdej sprawy związanej z rozwojem komputerów osobistych produkowanych przez ZUK MERA–ELZAB. Realizując to zamierzenie, począwszy już od tego numeru otwieramy nowy dział pt. "PROPOZYCJE I OFERTY".

W dziale tym zamieszczać będziemy nadsyłane do nas propozycje, oferty oraz ogłoszenia związane z podanym na wstępie tematem.

Czytelników zapraszamy do redagowania tego działu.

REDAKCJA

x x x

BOGUSŁAW TURKO
pracownik WBP
Wrocław, ul. Hercena
tel. 323–11

Zawiadamiam użytkowników mikrokomputera MERITUM, że na zamówienie mogę wykonać w krótkim czasie DISASSEMBLER mikroprocesora Z80 napisany w języku BASIC (prędkość disassemblacji poniżej 1 s/instrukcję) lub w języku wewnętrznym.

Ponadto informuję, że żadna z tych dwóch wersji programu nie przekroczy 4 k pamięci (!), a zatem nadaje się na wersję MERITUM I 16 k.

x x x

REDAKCJA BIULETYNU MERITUM
ZUK MERA–ELZAB
ul. Kruczkowskiego 39, tel. 72-20-21 w. 294
41-808 Z a b r z e

Autora nadesłanego do nas artykułu "PIERWIASTKI WIELOMIANU I, II, III i IV STOPNIA" prosimy o listowny lub telefoniczny kontakt z Redakcją Biuletynu.

GRAŻYNA CIOSEK—GZUBICKA
RESORTOWY OŚRODEK DOKSZTAŁCANIA
I DOSKONALENIA KADR MGİE, BYTOM

Jeszcze kilka lat temu słowo mikrokomputer wymawiano nieśmiało, a nawet rzecz można z zażenowaniem. Tylko niewiele osób wiedziało, co kryje się pod tym tajemniczym słowem. Dzisiaj mikrokomputer przestał być tematem tabu, a stał się wręcz niezbędnym sprzętem w biurze, szkole itp.

Jednak do tej pory nazwy MERITUM, ZX SPECTRUM, czy inne dla wielu wciąż są hasłem. Chcąc chociaż częściowo zaspokoić "głód" komputerowy nasz Ośrodek zorganizował serię kursów, na których słuchacz poznaje sam komputer, a przede wszystkim jego możliwości. Uczestnikom kursu zostają również podane podstawowe wiadomości z programowania w języku BASIC, co pozwala kursantom na samodzielną pracę z mikrokomputerem.

W ostatnim dniu zajęć padają pytania: Kiedy ciąg dalszy? Słyszac to pytanie po raz setny (bo tyle osób przeszkoliliśmy na 5 kursach, które odbyły się do tej pory) postanowiliśmy zorganizować ten "ciąg dalszy".

Od stycznia 1986 roku w naszym Ośrodku rozpoczyna się kolejna seria kursów. Tematyka tych kursów została dobrana pod kątem zainteresowań słuchaczy. Będzie to programowanie w języku BASIC dla tych, którzy znają już podstawy tego języka, a także dla tych, którzy mają zamiar go poznać.

Organizujemy kilkudniowe szkolenie nauczycieli, na którym będziemy prowadzić naukę języka BASIC—MERITUM. Chcemy pokazać jak można wykorzystać mikrokomputer do celów dydaktycznych w myśl:

- mikrokomputer wykłada
- mikrokomputer uczy
- mikrokomputer egzaminuje.

W dalszym ciągu będzie kontynuowany cykl kursów "Obsługa mikro— i minikomputerów", na których słuchacz może się zapoznać z możliwościami różnych systemów mikrokomputerowych, a także przyswoić sobie podstawowe wiadomości z programowania w języku BASIC. O ile zajdzie taka potrzeba jesteśmy w stanie zorganizować kursy w godzinach popołudniowych.

Do tej pory szkoliliśmy pracowników zakładów podległych Ministerstwu Górnictwa i Energetyki. Ze względu na duże zainteresowanie osób pracujących w innych resortach postanowiliśmy udostępnić kursy wszystkim pragnącym poznać tajemnice mikrokomputera.

Nasza dewiza to:

"Słowo mikrokomputer, to nie hasło lecz pomocnik w każdej dziedzinie naszej pracy zwiększający jej komfort".

WYKAZ KURSÓW PROWADZONYCH PRZEZ NASZ OŚRODEK

1. Zastosowanie mikro— i minikomputerów:

- kursokonferencja

Przeznaczona jest dla kadry kierowniczej zakładów i przedsiębiorstw.

Czas trwania 3 dni.

Zadaniem jej jest zapoznanie uczestników z zastosowaniem mikro— i minikomputerów w przedsiębiorstwie na różnych stanowiskach pracy.

2. Obsługa mikro— i minikomputerów:

Kurs przeznaczony jest dla kadry inżynieryjno—technicznej i pracowników administracji.

Czas trwania 10 dni.

Celem kursu jest zapoznanie słuchaczy z obsługą mikro— i minikomputerów, wykorzystaniem programów użytkowych, a także przekazanie wstępnych wiadomości o programowaniu w języku BASIC.

3. Programowanie w języku BASIC

SESJA I

Przeznaczona jest dla pracowników inżyniersko–technicznych i administracyjnych bez przygotowania specjalistycznego z informatyki.

SESJA II – 6 dni

Przeznaczona dla osób, które ukończyły sesję I lub 10–dniowy kurs "Obsługi mikro– i minikomputerów" organizowany w naszym Ośrodku.

Celem kursu jest nauka programowania w języku BASIC.

4. Obsługa i programowanie mikrokomputera MERITUM – kurs dla nauczycieli

Zadaniem kursu jest nauka programowania w języku BASIC ze specjalnym ukierunkowaniem wykorzystania komputera w celach dydaktycznych.

Kurs składa się z trzech sesji odbywających się w odstępach 2 do 6 tygodni.

Zainteresowanych prosimy o kontakt z naszym Ośrodkiem

telefon 81-87-17

lub

telefon 81-14-11 wewn. 153.

Adres:

RESORTOWY OŚRODEK DOKSZTAŁCANIA I DOSKONALENIA KADR MGİE
ul. Kosynierów, 41–907 B Y T O M

ZAKŁAD PROJEKTOWO–KONSTRUKCYJNY
SYSTEMÓW KOMPUTEROWYCH "AGAPOL"
KATOWICE, ul. DROZDÓW 17b, tel. 518-409

Zakład nasz zajmuje się projektowaniem i konstrukcją części sprzętowej i programowej systemów komputerowych. Główną naszą specjalnością są prace związane z mikrokomputerem MERITUM I oraz MERITUM II. Ponadto wykonujemy prace konstrukcyjne i programowe związane z innymi mikrokomputerami takimi jak ZX SPECTRUM, IBM PC itd.

O F E R U J E M Y :

- Inteligentny terminal graficzny. Możliwy do połączenia z dowolnym komputerem poprzez łącze równoległe bądź szeregowe w standardzie V24. Programowny listą zleceń podobną do listy zleceń plotera. Daje rozdzielczość obrazu 512 x 256. Współpracuje z dowolnym monitorem np. NEPTUN 156.
- Programator pamięci PROM/EPROM. Poprzez łącze równoległe można go połączyć z dowolnym mikrokomputerem. Umożliwia programowanie następujących typów pamięci: 2716, 2732A, 2732, 74S188, 74S288, 74S287, 74S387, 2764. Dla mikrokomputera ZX SPECTRUM oferujemy interfejs aplikacyjny i oprogramowanie programatora.
- Emulator mikroprocesora Z–80. Umożliwia wykorzystanie mikrokomputera MERITUM, jako systemu uruchomieniowego dla nowo konstruowanych mikrokomputerów opartych o mikroprocesor Z–80.
- Interfejs równoległy do mikrokomputera ZX SPECTRUM. Umożliwia podłączenie do mikrokomputera programatora, drukarki równoległej oraz innych urządzeń pracujących w równoległym reżimie transmisji. Daje również możliwość podłączenia do mikrokomputera monitora (np. NEPTUN 156) w miejsce niszczącego wzrok telewizora. Zawiera również przycisk zerowania ogólnego mikrokomputera.
- Karta interfejsu równoległego do komputera IBM PC lub IBM PC XT. Umożliwia podłączenie programatora, drukarki równoległej (np. DZM 180) lub innych urządzeń pracujących w trybie równoległym.
- Karty wejść/wyjść analogowych i cyfrowych. Przystosowane do współpracy z komputerem MERITUM I w wersji przemysłowej (w kasecie CAMAC). Umożliwiają zastosowanie tego mikrokomputera do sterowania obiektami przemysłowymi.
- Czytnik ROM–u. Umożliwia szybkie ładowanie programów z zewnętrznej pamięci ROM. Bardzo wygodne narzędzie dla często używanych nakładek.
- Przystosowanie mikrokomputera MERITUM II do pracy pod systemem operacyjnym CP/M wraz z pełnym oprogramowaniem użytkowym.

W ZAKRESIE OPROGRAMOWANIA OFERUJEMY :

- Oprogramowanie dla mikrokomputerów MERITUM I oraz MERITUM II. W oprogramowaniu tym oferujemy:
 - a) dla mikrokomputera MERITUM I

NODOS80 – nakładka BASIC-a umożliwiająca jego rozszerzenie o dodatkowe zlecenia takie jak re-

numeracja programu, kompresja, operacja na liczbach ósemkowych i szesnastkowych, jednoznakowe skróty instrukcji BASIC-a. W sumie 14 dodatkowych zleceń.

- SCRIPC** – Procesor tekstowy. Z MERITUM I robi inteligentną maszynę do pisania. Ma możliwość zapisu wprowadzonego tekstu na kasetę magnetofonową i wczytywanie tekstu z kasy. Wyrównywanie drukowanego tekstu do lewego lub prawego marginesu, centrowanie poziome lub pionowe i wiele innych możliwości.
- DEBIT** – Debugger dla mikrokomputera MERITUM I. Umożliwia testowanie i uruchamianie programów napisanych w języku wewnętrznym.
- INVERT** – Disassembler. Umożliwia symboliczne zdekodowanie obszaru pamięci zajmowanego przez program napisany w języku wewnętrznym. Pomocny przy uruchamianiu programów.
- EDTASM** – Edytor/Assembler. Daje możliwość tworzenia źródłowych programów assemblerowych, zapisu ich na kasetę magnetofonową lub wczytywanie z kasy magnetofonowej. Program wynikowy (po assemblacji) jest zapisywany na kasetę magnetofonową.
- AGAPOL FT** – Nakładka BASIC-a umożliwiająca wykorzystanie zleceń dyskowych (nieдоступnych w MERITUM I) jako zleceń realizujących szybki odczyt i zapis na kasetę magnetofonową w zmodyfikowanym standardzie. Ten nowy standard daje czterokrotne przyspieszenie operacji zapisu i odczytu.
- CROM** – Nakładka umożliwiająca zastosowanie przystawki zwanej czytnikiem romów. Rozszerza BASIC o dwa zlecenia: PDIR (listowanie kartoteki ROM-u) i PLOAD (ładowanie programu z ROM-u). Przystawka czytnika ROM-ów eliminuje konieczność posługiwania się magnetofonem kasetowym dla często wykorzystywanych programów.
- BIBMAT** – Biblioteka programów matematycznych i statystycznych napisana w BASIC-u.
- b) dla mikrokomputera MERITUM II
- MERDOS** – System operacyjny. Sprzedawny dla formatów dysku: 1 ÷ 35 ścieżek i pojedyncza gęstość, 2 ÷ 40 ścieżek i podwójna gęstość. Kompatybilny z systemem operacyjnym TRSDOS V. 2.3.
- F80** – Kompilator fortranu. Dostarczany łącznie z makroassemblerem i programem łączącym. Zgodny z F80 firmy Microsoft.
- L80** – Program łączący dla F80 i makroassemblera M80. Zgodny z linkerem L80 firmy Microsoft pracujący pod systemem CP/M.
- M80** – Makroassembler. Zgodny z makroassemblerem M80 firmy Microsoft.
- VC-VISICALC** – Program sporządzania tabel (ang.: spreadsheet).
- muSIMP i muMATH** – Interpretery języków opartych na Lisp-ie. Pomocne w zagadnieniach arytmetyki komputerowej i sztucznej inteligencji. Dostarczane łącznie z interakcyjnymi lekcjami dyskowymi.

- TASMON – Rewelacyjny debugger dyskowy z symbolicznym invertassemblerem i możliwością krokowego wykonywania procedur zapisanych w Rom-ie.
- SCRIPD – Procesor tekstowy. Kompatybilny z procesorem tekstowym SCRIPSIT dla mikrokomputera TRS.80.
- MAGAZYN – System zarządzający magazynem.
- CREATOR – Program tworzący programy obsługujące bazy danych (tzw. generator programów). Dostarczany łącznie z programem REPORTOR produkującym raporty dla wygenerowanej bazy danych.
- CERNLIB – Biblioteka programów matematycznych i statystycznych dla fortranu F80 (sprawdzona w IBJ w CERNIE – 50 p).
- BIBMAT – Biblioteka programów matematycznych i statystycznych w BASIC-u. Sprzedawana również dla mikrokomputera MERITUM I.
- FOURIER – Program edukacyjny do nauki teorii związanej z szeregami Fouriera.
- FLEXSORT – Obsługa tekstowej bazy danych.
- BASDEB – Debugger dla języka BASIC. Wygodny zastępca dla złeceń TRON i TROFF.
- ALCOR–PASCAL
- BASIC–COMPILATOR.

OFERUJEMY BOGATY ZESTAW GIER DLA MIKROKOMPUTERÓW MERITUM I i II

1. VODOO – gra typu adventure
2. DATESTONE – gra typu adventure
3. ATLANTIS – gra typu adventure z grafiką
4. PYRAMID – gra typu adventure
5. SARGON2d – gra w szachy
6. FBPLANET – gra typu adventure z grafiką
7. JUNGLBOY – gra "Tarzan"
8. GOMOKU – inteligentna gra w kółko i krzyżyk
9. ELIZA – zabawa w psychoterapię
10. BISMARCK – gra w okręty
11. ALLSTAR – gra w baseball
12. PIRATE – gra typu adventure
13. BATTLEFIELD – gra wojenna
14. NAPOLEON – gra wojenna, zdobywanie Europy przez Napoleona
15. BOUNCE – zwalczanie obiektów kosmicznych
16. ROBOT – walka ze zbuntowanymi robotami
17. HOLE – przeprawa przez czarne dziury
18. GALAXY – niszczenie najeźdźców z kosmosu
19. DEFEND – lot przez asteroidy
20. NIGHTDRIVE – rajd samochodowy

- | | |
|----------------|--|
| 21. FLIPPER | – bilard elektroniczny |
| 22. OTELLO | – gra starożytna |
| 23. DRAGON | – wyprawa w głąbiny |
| 24. ASSYLUM | – gra typu adventure |
| 25. KONG | – Koing-Kong ratuje swoich przyjaciół |
| 26. COLUMBIA | – symulator lotu promu kosmicznego |
| 27. TENIS | – gra w tenisa |
| 28. MUSICMAKER | – program komponujący |
| 29. STARWORS | – gwiazdne wojny |
| 30. HANGMAN | – matematyczny program edukacyjny dla dzieci |
| 31. TESTIQ | – program edukacyjny sprawdzający inteligencję |
| 32. ANDROID | – walka z robotami. |

KONKURS

Napisać program zamieniający liczby zapisane w numeracji rzymskiej na liczby dziesiętne. Program powinien sprawdzać poprawność składniową wprowadzanych danych i być możliwie krótki.

Uwaga:

Praca nad tym programem może dać pewne wyobrażenie o problemach związanych z pisaniem kompilatorów.

Spośród nadesłanych rozwiązań zespół oceniający w składzie:

- przedstawiciele redakcji Biuletynu
- Główny Konstruktor ZUK MERA–ELZAB
- programista OETO Politechniki Śląskiej

wyberze 3 (trzy) najciekawsze programy i przyzna nagrody w następującej wysokości:

- I nagroda – 5000 zł
- II nagroda – 3000 zł
- III nagroda – 2000 zł

Rozwiązanie w postaci kasyety z nagrany programem (do zwrotu) należy nadsyłać w terminie do dnia ~~30~~ 31 stycznia 1986 roku na adres:

Zakłady Urządzeń Komputerowych
MERA–ELZAB
ul. Kruczkowskiego 39, 41-808 Zabrze
"Redakcja Biuletynu MERITUM"

Redakcja zastrzega sobie prawo do odstąpienia od rozstrzygnięcia konkursu w przypadku ilości rozwiązań mniejszej od dziesięciu. Życzymy miłego spędzenia czasu przy rozwiązywaniu konkursu.

REDAKCJA

A oto publikujemy nagrodzone rozwiązanie konkursu z numeru 1 nadesłane przez Andrzeja Adamskiego z Grudziądza.

ANDRZEJ ADAMSKI
UL. SOBIESKIEGO 9/5
86-300 GRUDZIADZ

PROGRAM "LITERKI"

```
1 CLS:FOR G=15360 TO 15423:POKE G,191:POKE G+960,191:NEXT G
  :FOR G=15360 TO 16320 STEP 64:POKE G,191:POKE G+63,191:NEXT G
  :PRINT@400,"R Y S O W A N I E   K R Z Y Z A M I";
  :PRINT@942,"PROG. A.ADAMSKI";
2 FOR G=0 TO 200:NEXT G:PRINT@463,"*":FOR G=464 TO 499 STEP 2
  :PRINT@G,"+*":NEXT G:FOR G=0 TO 850:NEXT G:CLS
  :FOR G=15360 TO 15423:POKE G,166:POKE G+960,153:NEXT G
3 FOR G=15360 TO 16320 STEP 64:POKE G,153:POKE G+63,166:NEXT G
  :PRINT@391,"NACISNIENCIE KLAWISZA RYSUJE KRZYZ O SRODKU < LITERY >";
  :PRINT@729,"JESTES GOTOWY ? - NACISNIJ < ENTER >";
  :PRINT@455,"< ENTER >-- PRZYWOLANIE LANCUCHA LITER";
4 IF INKEY*( )CHR*(13) THEN 4
30 CLEAR 101
40 DIM A*(100):R1=100
50 CLS:DIM P(25),L(25),Q(5):FOR A=15362 TO 15392 STEP 6
  :FOR B=A TO 960+A STEP 64:POKE B,191:NEXT B:NEXT A
  :FOR A=15362 TO 16322 STEP 192:FOR B=A TO 30+A:POKE B,191
  :NEXT B:NEXT A
55 C=65:I=1:FOR A=15427 TO 16195 STEP 192:FOR B=A TO 24+A STEP 6
  :POKE B,C:POKE B-1,149:P(I)=B+1:C=C+1:I=I+1:NEXT B:NEXT A
  :FOR I=1 TO 25:READ L1:L(I)=L1:NEXT I
60 DATA 1,2,2,2,3,4,5,5,5,6,4,5,5,5,6,4,5,5,5,6,7,8,8,8,9
80 POKE 15604,42:POKE 15608,42
90 A*=INKEY*:IF A*=CHR*(13) THEN 200 ELSE IF A*="" THEN GOTO 90
  ELSE IF ASC(A*)<65 OR ASC(A*)>89 THEN PRINT@116,"BLAD";
  :GOTO 90
94 R1=R1-1:A*(R1)=A*:PRINT@167,"ZOSTALO";R1;"KROKOW ";
  :IF R1=0 THEN 200
95 PRINT@231,"AKTU.LITERA";:PRINT@116," ";:PRINT@246,A*;
  :I=ASC(A*)-64:P=P(I):L=L(I)
  :ON L GOTO 110,115,120,125,130,135,140,145,150
110 G=3:Q(2)=6:Q(3)=192:GOTO 155
115 G=4:Q(2)=-6:Q(3)=6:Q(4)=192:GOTO 155
120 G=3:Q(2)=-6:Q(3)=192:GOTO 155
125 G=4:Q(2)=6:Q(3)=192:Q(4)=-192:GOTO 155
130 G=5:Q(2)=6:Q(3)=-6:Q(4)=192:Q(5)=-192:GOTO 155
135 G=4:Q(2)=-6:Q(3)=192:Q(4)=-192:GOTO 155
140 G=3:Q(2)=6:Q(3)=-192:GOTO 155
145 G=4:Q(2)=-6:Q(3)=6:Q(4)=-192:GOTO 155
150 G=3:Q(2)=-6:Q(3)=-192:GOTO 155
155 Q(1)=0:FOR J=1 TO G:R=P+Q(J):Z=PEEK(R):IF Z=32 THEN H=191
  ELSE H=32
160 FOR W=R TO R+3 :POKE W,H:NEXT W
165 FOR W=R+63 TO R+67 :POKE W,H:NEXT W
170 NEXT J:GOTO 90
200 CLS:FOR E1=100 TO R1 STEP -1:PRINT@E1*3+84,A*(E1);:NEXT E1
210 IF INKEY*( )CHR*(13) THEN 210 ELSE RUN
```

ZUK MERA–ELZAB OFERUJE:

**I. ZESTAW PROGRAMÓW MATEMATYCZNYCH
(TAŚMA Z PROGRAMEM + BROSZURA)**

Cena ... 9800

Zestaw zawiera:

Strona 1.

1. OPIS
2. OATR – obliczanie całki oznaczonej z funkcji $f(x)$ z zadaną dokładnością (metoda trapezów).
3. CAŁKA – obliczanie całki oznaczonej z funkcji $f(x)$ (uproszczona) metoda trapezów
4. MRKG – rozwiązywanie układu równań różniczkowych zwyczajnych pierwszego rzędu (metoda Milne'a)
5. HIPER – obliczanie wartości funkcji hiperbolicznych i do nich odwrotnych
6. ALI – interpolacja wartości funkcji dla danego argumentu x . Jeżeli funkcja dana jest w postaci tablicy argumentów i wartości
7. INK – interpolacja wartości funkcji M zmiennych (metoda najmniejszych kwadratów)
8. RNI – rozwiązywanie równania nieliniowego $f(x)=0$ (metoda Newtona)

Strona 2.

1. OPIS
2. MACIERZ – procedury macierzowe: dodawanie, odejmowanie, mnożenie przez użytkownika funkcji dla wybranej kolumny macierzy
3. INUM – odwracanie macierzy i obliczanie jej wyznacznika
4. WAWM – obliczanie wartości własnych macierzy rzeczywistej symetrycznej
5. URL – rozwiązywanie układu równań liniowych $A * X=B$ (metoda eliminacji)

II. ZESTAW PROGRAMÓW DO OBLICZEŃ STATYSTYCZNYCH
(2 TAŚMY Z PROGRAMEM + BROSZURA)

Cena 12100

Zestaw zawiera:

Taśma 1 Strona 1

1. CLEAN — program oblicza średnia arytmetyczna i zmienność całkowita z eliminacją odstających obserwacji
2. REGREMP — dla n obserwacji dwucechowych, program dzieli przedział zmienności cechy x na zadaną liczbę klas, a następnie przydziela każdą z wartości Y_i do jednej z utworzonych klas. Dla każdej klasy obliczane są wartości średnie, sumy kwadratów oraz odchylenie standardowe wartości Y_i znajdujących się w danej klasie.
3. DETRATIO — wybór cech o największej sile dyskryminacji na podstawie kryterium Lambda Wilksa

Strona 2.

1. OSTN — obliczanie szeregów rozdzielczych oraz testowanie normalności rozkładu
2. IDEP1 — wybór reprezentacji cech metodą analizy regresji
3. EOUCOV — testowanie równości k macierzy kowariancji

Taśma 2 Strona 1

1. MAXTEPREGR — obliczanie regresji wielokrotnej z wyborem zmiennych
2. EORTEST — testy równoległości i identyczności płaszczyzn regresji

Strona 2

1. POLTRENDDEC — aproksymacja zaobserwowanych wartości zmiennej y wartościami oczekiwanymi według zależności wielomianowej
2. MANOVA — wielozmienna analiza wariancji z jednym kierunkiem klasyfikacji

III. SZYBKI LOADER – XLOAD 16k
(1 TAŚMA Z PROGRAMEM + OPIS)

Cena .. 3100

Taśma zawiera:

XLOAD 16k – program umożliwia szybki zapis, weryfikację i odczyt programów z taśmy magneto-
tofonowej. Prędkość zapisu około 2000 bitów/s.

IV. PROGRAM KARTOTEKA 16k
(1 TAŚMA + OPIS)

Cena .. 13900

Taśma zawiera:

KARTOTEKA 16k – program obsługi danych typu kartoteka

Program KARTOTEKA umożliwia:

- dowolną organizację kartoteki na ekranie komputera
- automatyzację wypełniania kartoteki
- aktualizację istniejącej kartoteki
- sortowanie według dowolnego klucza całej kartoteki
- wyszukiwanie określonej zawartości według określonego klucza
- drukowanie całej lub części kartoteki
- przeglądanie kartoteki

Program KARTOTEKA pracuje z szybkim loaderem XLOAD.

**V. PROGRAM OPTYMALIZACJA PODSTAWOWYCH
PARAMETRÓW EKONOMICZNYCH PRZEDSIĘBIORSTWA
Z PUNKTU WIDZENIA MAKSYMALIZACJI ZYSKU – PFAZ
(1 TAŚMA + OPIS)**

Cena **35000**

Taśma zawiera:

PFAZ – program optymalizacji parametrów ekonomicznych przedsiębiorstwa.

Program dokonuje:

- najkorzystniejszego (z punktu widzenia maksymalizacji zysku) wyboru parametrów ekonomicznych: sprzedaży, wynagrodzeń, zatrudnienia
- wyświetlenia na trzech planszach uzyskanych wyników obliczeń.

Adresy firm prowadzących serwis mikrokomputerów MERITUM z wyposażeniem

Lp.	Firma	Miejscowość	Ulica	Nr telef.	Zakres napraw
1.	PTB Predom Org Meraserw	01 - 244 Warszawa 00 - 871 Warszawa	ul. Bema 57a ul. Żelazna 67	32-12-31 24-39-99	mikrokomputer MERITUM, zasilacz, monitor Neptun 156
2.	ZTB Predom Org Meraserw	80 - 341 Gdańsk 80 - 557 Gdańsk	ul. Bałtycka 5 ul. Marynarki Polskiej 59	53-01-82 43-12-71	
3.	ZTB Predom Org Meraserw Meraserw	40 - 064 Katowice 41 - 200 Sosnowiec 45 - 037 Opole	ul. Kopernika 28 ul. Ostrogórska 9 ul. Sienkiewicza 23	51-80-11 66-65-89 367 - 52	
4.	ZTB Predom Org Meraserw	30 - 114 Kraków 30 - 114 Kraków	ul. Kościuszki 43 ul. Kościuszki 56	26-68-44 22-60-43	
5.	ZTB Predom Org	20 - 804 Lublin	ul. Wojciechowska 3a	55-00-17	
6.	ZTB Predom Org Meraserw	93 - 176 Łódź 90 - 720 Łódź	ul. Suwalska 16 ul. Wólczańska 4	84-27-30 33-45-68	
7.	ZTB Predom Org Meraserw	61 - 813 Poznań 60 - 203 Poznań	ul. Ratajczaka 2 ul. Krawthofera 36	513-73 656-14	
8.	ZTB Predom Org Meraserw	51 - 630 Wrocław 50 - 224 Wrocław	ul. Chełmońskiego 10 ul. K. Marksa 20	48-42-93 21-00-05	
9.	Meraserw	70 - 312 Szczecin	ul. Bema 5	801-55	
10.	Meraserw	85 - 016 Bydgoszcz	ul. 3-go Maja 3	21-03-99	
11.	Meraserw	25 - 361 Kielce	ul. Szymanowskiego 3	232-31	
12.	Meraserw	87 - 100 Toruń	ul. Koniuchy 75	207-11	
13.	Unitra Unimor	80 - 822 Gdańsk	ul. Rzeźnicza 54/56		
14.	Unitra Serwis	na terenie całego kraju			magnetofon
15.	ZUK Mera-Elzab	41 - 800 Zabrze	ul. 3-go Maja 82	71-80-00	mikrokomputer MERITUM zasilacz

Warunki płatności Biuletynu użytkowników komputerów osobistych MERITUM

cena 1 egz. – 200 zł

płatne za ZALICZENIEM POCZTOWYM na konto Zakładów Urządzeń Komputerowych MERA-ELZAB
ul. Kruczkowskiego 39, 41 - 808 Zabrze, NBP II O/M Zabrze nr 27397 – 723

.....
.....
.....
(nazwa i adres zamawiającego)

**Zakłady Urządzeń Komputerowych
MERA-ELZAB
ul. Kruczkowskiego 39
41-808 ZABRZE**

Zamawiam numer 4 Biuletynu użytkowników komputerów osobistych MERITUM w ilości egzemplarzy.

Należną z tego tytułu kwotę zobowiązuję się przekazać za zaliczeniem pocztowym na konto
Zakładów Urządzeń Komputerowych MERA-ELZAB – NBP II O/M Zabrze nr 27397 – 723.

.....
.....
.....
(nazwa i adres zamawiającego)

**Zakłady Urządzeń Komputerowych
MERA-ELZAB
ul. Kruczkowskiego 39
41-808 ZABRZE**

Zamawiam numer 5 Biuletynu użytkowników komputerów osobistych MERITUM w ilości egzemplarzy.

Należną z tego tytułu kwotę zobowiązuję się przekazać za zaliczeniem pocztowym na konto
Zakładów Urządzeń Komputerowych MERA-ELZAB – NBP II O/M Zabrze nr 27397 – 723.

