

# 1.6. Реализация вещественной арифметики на машинном языке

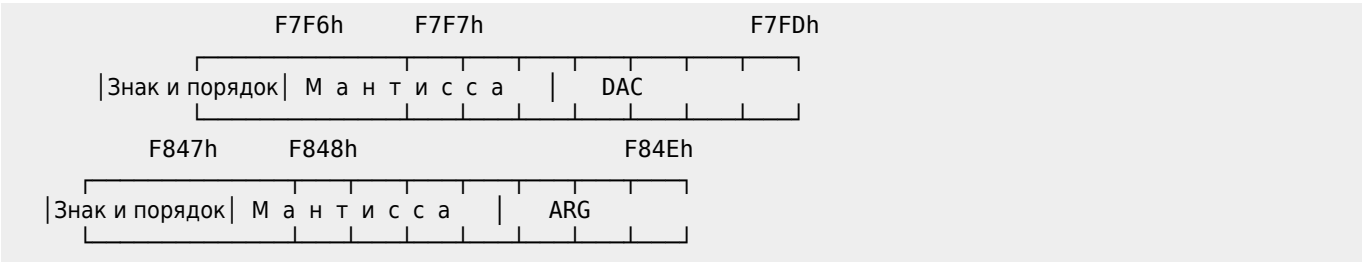
Изучай все не из тщеславия, а ради практической пользы.

—Г.Лихтенберг. Афоризмы

В системной области имеются два вещественных «регистра» — *аккумулятора*, при помощи которых компьютер осуществляет все арифметические операции: \*

- DAC («Deci**ma**l ACcu**m**ulator«-»десятичный аккумулятор») — F7F6h, 16 байтов,
- ARG («ARGu**m**ent«-»аргумент») — F847h, 16 байтов.

Вещественное число в аккумуляторах размещается следующим образом:



Целое число в аккумуляторе располагается следующим образом:



Подробнее о хранении чисел см. в [разделе X.4.1](#).

В ячейке VALTYP системной области (по адресу F663h) хранится тип числа, находящегося в аккумуляторе DAC, причем тип закодирован следующим образом:

- 2 — целое число;
- 4 — вещественное число одинарной точности;
- 8 — вещественное число двойной точности.

В следующих табличках приведен список входных точек очень полезных подпрограмм ROM.

## 1.6.1. Пересылки

| Имя подпрограммы | Адрес | Выполняемые действия | Тип    | Изменяемые регистры |
|------------------|-------|----------------------|--------|---------------------|
| MAF              | 2C4Dh | ARG := DAC           | Double | A,B,D,E,H,L,ARG     |
| AM               | 2C50h | ARG := (HL)          | Double | A,B,D,E,H,L,ARG     |
| MOV8DH           | 2C53h | (DE):= (HL)          | Double | A,B,D,E,H,L         |
| MFA              | 2C59h | DAC := ARG           | Double | A,B,D,E,H,L,DAC     |
| MFM              | 2C5Ch | DAC := (HL)          | Double | A,B,D,E,H,L,DAC     |

| Имя подпрограммы | Адрес | Выполняемые действия | Тип    | Изменяемые регистры |
|------------------|-------|----------------------|--------|---------------------|
| MMF              | 2C67h | (HL):= DAC           | Double | A,B,D,E,H,L         |
| MOV8HD           | 2C6Ah | (HL):= (DE)          | Double | A,B,D,E,H,L         |
| XTF              | 2C6Fh | (SP):= DAC           | Double | A,B,D,E,H,L         |
| PHA              | 2CC7h | ARG := (SP)          | Double | A,B,D,E,H,L,ARG     |
| PHF              | 2CCCh | DAC := (SP)          | Double | A,B,D,E,H,L,DAC     |
| PPA              | 2CDCh | (SP):= ARG           | Double | A,B,D,E,H,L         |
| PPF              | 2CE1h | (SP):= DAC           | Double | A,B,D,E,H,L         |
| PUSHF            | 2EB1h | DAC := (SP)          | Single | D,E,DAC             |
| MOVFM            | 2EBEh | DAC := (HL)          | Single | B,C,D,E,H,L,DAC     |
| MOVFR            | 2EC1h | DAC := (CBED)        | Single | DAC,D,E             |
| MOVRF            | 2ECCh | (CBED):= DAC         | Single | B,C,D,E,H,L         |
| MOVRMI           | 2ED6h | (CBED):= (HL)        | Single | B,C,D,E,H,L         |
| MOVRLM           | 2EDFh | (BCDE):= (HL)        | Single | B,C,D,E,H,L         |
| MOVMF            | 2EE8h | (HL):= DAC           | Single | B,C,D,E,H,L         |
| MOVE             | 2EEBh | (HL):= (DE)          | Single | B,C,D,E,H,L         |
| VMOVAM           | 2EEFh | ARG := (HL)          | VALTYP | B,C,D,E,H,L,ARG     |
| MOVVFM           | 2EF2h | (DE):= (HL)          | VALTYP | B,C,D,E,H,L         |
| VMOVE            | 2EF3h | (HL):= (DE)          | VALTYP | B,C,D,E,H,L         |
| VMOVFA           | 2F05h | DAC := ARG           | VALTYP | B,C,D,E,H,L,DAC     |
| VMOVFM           | 2F08h | DAC := (HL)          | VALTYP | B,C,D,E,H,L,DAC     |
| VMOVAF           | 2F0Dh | ARG := DAC           | VALTYP | B,C,D,E,H,L,ARG     |
| VMOVMF           | 2F10h | (HL):= DAC           | VALTYP | B,C,D,E,H,L         |

**Пример 1.** Записать число в аккумулятор DAC,а затем прочесть его.

[1060-01.bas](#)

 [1060-01.bas](#)

```

10 CLEAR 200,&HF000:DEFUSR=&HF000
20 INPUT"Ваше число";N:A$=HEX$(VARPTR(N))
30 POKE &HF100,VAL("&h"+RIGHT$(A$,2)):POKE&HF101,VAL("&h"+LEFT$(A$,2))
40 DATA 2A,00,F1      :'LD    HL,(F100h)
50 DATA CD,5C,2C      :'CALL 2C5Ch
70 DATA 21,F6,F7      :'LD    HL,F7F6h
80 DATA 11,00,F2      :'LD    DE,F200h
90 DATA 01,10,00      :'LD    BC,10h
100 DATA ED,B0         :'LDIR
130 DATA C9            :'RET
140 FOR T=0 TO 17:READ Z$:POKE &HF000+T,VAL("&h"+Z$):NEXT
150 X=USR(X)
160 FOR T=0 TO 15:PRINT HEX$(PEEK(VARPTR(N)+T)),HEX$(PEEK(&HF200+T))
170 NEXT

```

**Пример 2.** Записать число в аккумулятор ARG,а затем прочесть его.

[1060-02.bas](#)

 [1060-02.bas](#)

```

10 CLEAR 200,&HF000:DEFUSR=&HF000
20 INPUT"Ваше число";N:A$=HEX$(VARPTR(N))
30 POKE &HF100,VAL("&h"+RIGHT$(A$,2)):POKE&HF101,VAL("&h"+LEFT$(A$,2))
40 DATA 2A,00,F1      :'LD    HL,(F100h)
50 DATA CD,50,2C      :'CALL 2C50h

```

```

70 DATA 21,47,F8      : 'LD    HL,F847h
80 DATA 11,00,F2      : 'LD    DE,F200h
90 DATA 01,10,00      : 'LD    BC,10h
100 DATA ED,B0        : 'LDIR
130 DATA C9           : 'RET
140 FOR T=0 TO 17:READ Z$:POKE &HF000+T,VAL("&h"+Z$):NEXT
150 X=USR(X)
160 FOR T=0 TO 15:PRINT HEX$(PEEK(VARPTR(N)+T)),HEX$(PEEK(&HF200+T))
170 NEXT

```

## 1.6.2. Арифметические операции над целыми числами

Наиболее полезны те советы, которым легко следовать.

—Л.Вовенарг

| Имя подпрограммы | Адрес | Выполняемые действия | Тип           | Изменяемые регистры |
|------------------|-------|----------------------|---------------|---------------------|
| UMULT            | 314Ah | DE := BC·DE          | A,B,C,D,E     |                     |
| ISUB             | 3167h | HL := DE-HL          | A,B,C,D,E,H,L |                     |
| IADD             | 3172h | HL := DE+HL          | A,B,C,D,E,H,L |                     |
| IMULT            | 3193h | HL := DE·HL          | A,B,C,D,E,H,L |                     |
| IDIV             | 31E6h | HL := DE/HL          | A,B,C,D,E,H,L |                     |
| IMOD             | 323Ah | HL := DE mod HL      | A,B,C,D,E,H,L |                     |

Пример 3. Умножение целых чисел (операнды находятся в регистрах HL и DE).

[1060-03.bas](#)

 [1060-03.bas](#)

```

10 CLEAR 200,&HD000
20 DEFUSR=&HD000
30 I=&HD000
40 READ A$:IF A$="z" THEN A=USR(0):PRINT PEEK(&H9000):END
60 POKE I,VAL("&h"+A$):I=I+1:GOTO 40
90 DATA 21,0A,00:' LD    HL,000Ah ; Загрузка регистра HL константой
100 DATA 11,04,00:' LD    DE,0004h ; Загрузка регистра DE константой
110 DATA CD,93,31:' CALL 3193h ; Вызов нужной подпрограммы
120 DATA 22,00,90:' LD    (9000h),HL
130 DATA C9        : ' RET
140 DATA "z"       : '

```

## 1.6.3. Арифметические операции над вещественными числами

| Имя подпрограммы | Адрес | Выполняемые действия | Изменяемые регистры   |
|------------------|-------|----------------------|-----------------------|
| DECSUB           | 268Ch | DAC := DAC-ARG       | A,B,C,D,E,H,L,DAC     |
| DECADD           | 269Ah | DAC := DAC+ARG       | A,B,C,D,E,H,L,DAC     |
| DECNRM           | 26FAh | Нормализовать DAC    | A,B,C,D,E,H,L,DAC     |
| DECROU           | 273Ch | Округлить DAC        | A,B,C,D,E,H,L,DAC     |
| DECMUL           | 27E6h | DAC := DAC·ARG       | A,B,C,D,E,H,L,DAC     |
| DECDIV           | 289Fh | DAC := DAC/ARG       | A,B,C,D,E,H,L,DAC     |
| COS              | 2993h | DAC := COS(DAC)      | A,B,C,D,E,H,L,DAC,ARG |

| Имя подпрограммы | Адрес | Выполняемые действия | Изменяемые регистры   |
|------------------|-------|----------------------|-----------------------|
| SIN              | 29ACh | DAC := SIN(DAC)      | A,B,C,D,E,H,L,DAC,ARG |
| TAN              | 29FBh | DAC := TAN(DAC)      | A,B,C,D,E,H,L,DAC,ARG |
| ATN              | 2A14h | DAC := ATN(DAC)      | A,B,C,D,E,H,L,DAC,ARG |
| LOG              | 2A72h | DAC := LOG(DAC)      | A,B,C,D,E,H,L,DAC,ARG |
| SQR              | 2AFFh | DAC := SQR(DAC)      | A,B,C,D,E,H,L,DAC,ARG |
| EXP              | 2B4Ah | DAC := EXP(DAC)      | A,B,C,D,E,H,L,DAC,ARG |
| RND              | 2BDFh | DAC := RND(DAC)      | A,B,C,D,E,H,L,DAC,ARG |
| SIGN             | 2E71h | A := SIGN(DAC)       | A                     |
| ABSFN            | 2E82h | DAC := ABS(DAC)      | A,B,C,D,E,H,L,DAC,ARG |
| NEG              | 2E8Dh | DAC := - DAC         | A,H,L,DAC             |
| SGN              | 2E97h | DAC := SGN(DAC)      | A,H,L,DAC             |

Пример 4. Сумма двух вещественных чисел.

[1060-04.bas](#)

 [1060-04.bas](#)

```

10 CLEAR 200,&HF000:DEFUSR=&HF000
20 DATA 21,00,00      : 'LD  HL,address(N1) ;F000h
30 DATA CD,5C,2C      : 'LD  DAC,(HL)      ;F003h
40 DATA 21,00,00      : 'LD  HL,address(N2) ;F006h
50 DATA CD,50,2C      : 'LD  ARG,(HL)       ;F009h
60 DATA CD,9A,26      : 'DAC:=DAC+ARG       ;F00Ch
70 DATA 21,00,00      : 'LD  HL,address(N1) ;F00Fh
80 DATA CD,67,2C      : 'LD  (HL),DAC       ;F012h
90 DATA C9             : 'RET                ;F015h
100 FOR T=0 TO &H15:READ Z$:POKE &HF000+T,VAL("&h"+Z$):NEXT
110 INPUT"Первое слагаемое";N1:INPUT"Второе слагаемое";N2
120 A$=HEX$(VARPTR(N1)):B$=HEX$(VARPTR(N2))
130 L1$=RIGHT$(A$,2):L2$=RIGHT$(B$,2)
140 H1$=LEFT$(A$,2):H2$=LEFT$(B$,2)
150 POKE &HF001,VAL("&h"+L1$):POKE &HF002,VAL("&h"+H1$)
160 POKE &HF007,VAL("&h"+L2$):POKE &HF008,VAL("&h"+H2$)
170 POKE &HF010,VAL("&h"+L1$):POKE &HF011,VAL("&h"+H1$)
180 H=USR(H)
190 PRINT"Сумма: ";N1

```

Пример 5. Возведение в степень.

[1060-05.bas](#)

 [1060-05.bas](#)

```

10 CLEAR 200,&HF000:DEFUSR=&HF000
20 DATA 21,00,00      : 'LD  HL,address(N1) ;F000h
30 DATA CD,5C,2C      : 'LD  DAC,(HL)      ;F003h
40 DATA 21,00,00      : 'LD  HL,address(N2) ;F006h
50 DATA CD,50,2C      : 'LD  ARG,(HL)       ;F009h
60 DATA CD,D7,37      : 'DAC:=DAC^ARG       ;F00Ch
70 DATA 21,00,00      : 'LD  HL,address(N1) ;F00Fh
80 DATA CD,67,2C      : 'LD  (HL),DAC       ;F012h
90 DATA C9             : 'RET                ;F015h
100 FOR T=0 TO &H15:READ Z$:POKE &HF000+T,VAL("&h"+Z$):NEXT
110 INPUT"N1";N1:INPUT"N2";N2
120 A$=HEX$(VARPTR(N1)):B$=HEX$(VARPTR(N2))
130 L1$=RIGHT$(A$,2):L2$=RIGHT$(B$,2)
140 H1$=LEFT$(A$,2):H2$=LEFT$(B$,2)
150 POKE &HF001,VAL("&h"+L1$):POKE &HF002,VAL("&h"+H1$)
160 POKE &HF007,VAL("&h"+L2$):POKE &HF008,VAL("&h"+H2$)

```

```

170 POKE &HF010,VAL("&h"+L1$):POKE &HF011,VAL("&h"+H1$)
180 H=USR(H)
190 PRINT"N1^N2=";N1

```

*Пример 5.* Вычисление значения функции синус вещественного аргумента.

[1060-05.bas](#)

 [1060-05.bas](#)

```

10 CLEAR 200,&HF000:DEFUSR=&HF000
20 DATA 21,00,00      : 'LD  HL,address(N1) ;F000h
30 DATA CD,5C,2C      : 'LD  DAC,(HL)      ;F003h
60 DATA CD,AC,29      : 'DAC:=SIN(DAC)      ;F006h
70 DATA 21,00,00      : 'LD  HL,address(N1) ;F009h
80 DATA CD,67,2C      : 'LD  (HL),DAC       ;F00Ch
90 DATA C9            : 'RET                ;F00Fh
100 FOR T=0 TO 15:READ Z$:POKE &HF000+T,VAL("&h"+Z$):NEXT
110 INPUT"Число";N1:A$=HEX$(VARPTR(N1))
130 L1$=RIGHT$(A$,2):H1$=LEFT$(A$,2)
150 POKE &HF001,VAL("&h"+L1$):POKE &HF002,VAL("&h"+H1$)
170 POKE &HF00A,VAL("&h"+L1$):POKE &HF00B,VAL("&h"+H1$)
180 H=USR(H):PRINT"Синус:";N1

```

*Пример 7.* Генерация псевдослучайного числа.

[1060-07.bas](#)

 [1060-07.bas](#)

```

10 CLEAR 200,&HF000:DEFUSR=&HF000
20 DATA 21,00,00      : 'LD  HL,address(N1) ;F000h
30 DATA CD,5C,2C      : 'LD  DAC,(HL)      ;F003h
60 DATA CD,DF,2B      : 'DAC:=RND(DAC)      ;F006h
70 DATA 21,00,00      : 'LD  HL,address(N1) ;F009h
80 DATA CD,67,2C      : 'LD  (HL),DAC       ;F00Ch
90 DATA C9            : 'RET                ;F00Fh
100 FOR T=0 TO 15:READ Z$:POKE &HF000+T,VAL("&h"+Z$):NEXT
110 INPUT"Число";N1
120 A$=HEX$(VARPTR(N1))
130 L1$=RIGHT$(A$,2)
140 H1$=LEFT$(A$,2)
150 POKE &HF001,VAL("&h"+L1$):POKE &HF002,VAL("&h"+H1$)
170 POKE &HF00A,VAL("&h"+L1$):POKE &HF00B,VAL("&h"+H1$)
180 H=USR(H):PRINT"RND(n1)=";N1

```

*Пример 8.* Вычислить значение функции двух переменных  $z = \cos(x) + \sin(y)$

[1060-08.bas](#)

 [1060-08.bas](#)

```

10 CLEAR 200,&HF000:DEFUSR=&HF000:T=0
20 INPUT"X=";X:INPUT"Y=";Y:A1$=HEX$(VARPTR(X)):A2$=HEX$(VARPTR(Y))
30 L1=VAL("&h"+RIGHT$(A1$,2)):R1=VAL("&h"+LEFT$(A1$,2)) 'Адрес числа X
40 L2=VAL("&h"+RIGHT$(A2$,2)):R2=VAL("&h"+LEFT$(A2$,2)) 'Адрес числа Y
50 POKE &HF301,L1:POKE &HF302,R1:POKE &HF303,L2:POKE &HF304,R2
60 READ Z$:IF Z$<>"RET" THEN POKE &HF000+T,VAL("&h"+Z$):T=T+1:GOTO 60
70 DATA 2A,01,F3 : 'LD  HL,(F301h); HL содержит адрес переменной X
75 DATA E5       : 'PUSH HL
80 DATA CD,5C,2C : 'LD  DAC,(HL) ; DAC:=X
90 DATA CD,93,29 : 'DAC:=COS(DAC) ; DAC:=COS(X)
95 DATA E1       : 'POP  HL
97 DATA E5       : 'PUSH HL
100 DATA CD,67,2C : 'LD  (HL),DAC ; Z:=COS(X) ← "Прячем" результат в
; любую заранее зарезервированную
; группу ячеек памяти
; В нашем случае: Z ≡ X
110 DATA 2A,03,F3 : 'LD  HL,(F303h); HL содержит адрес переменной Y

```

```

120 DATA CD,5C,2C : 'LD DAC,(HL) ; DAC:=Y
130 DATA CD,AC,29 : 'DAC:=SIN(DAC) ; DAC:=SIN(Y)
140 DATA CD,4D,2C : 'LD ARG,DAC ; ARG:=SIN(Y)
145 DATA E1 : 'POP HL ;
147 DATA E5 : 'PUSH HL ;
150 DATA CD,5C,2C : 'LD DAC,(HL) ; DAC:=Z
160 DATA CD,9A,26 : 'ADD DAC,ARG ; DAC:=Z+SIN(Y)
165 DATA E1 : 'POP HL ;
170 DATA CD,67,2C : 'LD (HL),DAC ; X:=DAC
180 DATA C9,"RET" : 'RET
190 PRINT COS(X)+SIN(Y)
200 A=USR(A):PRINT X 'Результат

```

### 1.6.4. Возведение в степень

| Имя подпрограммы | Адрес | Выполняемые действия | Тип                |
|------------------|-------|----------------------|--------------------|
| SNGEXP           | 37C8h | DAC := DAC^ARG       | Одинарная точность |
| DBLEXP           | 37D7h | DAC := DAC^ARG       | Двойная точность   |
| INTEXP           | 383Fh | DAC := DE^HL         | Целая              |

*Пример 9.* Вычислить  $3^2$ .

[1060-09.bas](#)

 [1060-09.bas](#)

```

10 CLEAR 200,&HF000:DEFUSR=&HF000
30 DATA 11,03,00 : 'LD DE,0003h
40 DATA 21,02,00 : 'LD HL,0002h
50 DATA CD,3F,38 : 'CALL 383Fh
60 DATA 21,F6,F7 : 'LD HL,F7F6h
70 DATA 11,00,F2 : 'LD DE,F200h
80 DATA 01,10,00 : 'LD BC,10h
90 DATA ED,B0 : 'LDIR
130 DATA C9 : 'RET
140 FOR T=0 TO 19:READ Z$:POKE &HF000+T,VAL("&h"+Z$):NEXT
150 X=USR(X)
160 FOR T=0 TO 15:PRINT HEX$(PEEK(&HF200+T));" ";:NEXT

```

### 1.6.5. Сравнение


| Имя подпрограммы | Адрес | Тип     | Левая часть условия | Правая часть условия | Изменяемые регистры |
|------------------|-------|---------|---------------------|----------------------|---------------------|
| FCOMP            | 2F21h | Single  | CBED                | DAC                  | HL                  |
| ICOMP            | 2F4Dh | Integer | DE                  | HL                   | HL                  |
| XDCOMP           | 2F5Ch | Double  | ARG                 | DAC                  | A,B,C,D,E,H,L       |

Подпрограмма возвращает следующее содержимое регистра A

- -1, если левая часть условия < правой части условия,
- 0, если левая часть условия = правой части условия,
- 1, если левая часть условия > правой части условия.

*Пример 10.* Сравнение двух вещественных чисел.

[1060-10.bas](#)

 [1060-10.bas](#)

```

10 CLEAR 200,&HD000:DEFUSR=&HD000
20 DATA 21,00,00 : 'LD HL,address(N1)
30 DATA CD,5C,2C : 'LD DAC,(HL)

```

```

40 DATA 21,00,00      : 'LD HL,address(N2)
50 DATA CD,50,2C      : 'LD ARG,(HL)
60 DATA CD,5C,2F      : 'Сравнить DAC и ARG
70 DATA 32,00,F1      : 'LD (F100h),A
90 DATA C9             : 'RET
100 FOR T=0 TO &H12:READ Z$:POKE &HD000+T,VAL("&h"+Z$):NEXT
110 INPUT"Left";N1:INPUT"Right";N2
120 A$=HEX$(VARPTR(N1)):B$=HEX$(VARPTR(N2))
130 L1$=RIGHT$(A$,2):L2$=RIGHT$(B$,2)
140 H1$=LEFT$(A$,2):H2$=LEFT$(B$,2)
150 POKE &HD001,VAL("&h"+L1$):POKE &HD002,VAL("&h"+H1$)
160 POKE &HD007,VAL("&h"+L2$):POKE &HD008,VAL("&h"+H2$)
180 H=USR(H):A=PEEK(&HF100)
190 IF A=255 THEN PRINT"LEFT < RIGHT"
200 IF A=1 THEN PRINT"LEFT > RIGHT"
210 IF A=0 THEN PRINT"LEFT = RIGHT"

```

### 1.6.6. Преобразование типов

| Имя подпрограммы | Адрес | Выполняемые действия                                   |
|------------------|-------|--|
| FRCINT           | 2F8Ah | Содержимое DAC преобразуется к целому типу             |
| FRCSNG           | 2FB2h | Содержимое DAC преобразуется к типу одинарной точности |
| FRCDBL           | 303Ah | Содержимое DAC преобразуется к типу двойной точности   |
| FIXER            | 30BEh | DAC:=SGN(DAC)*INT(ABS(DAC))                            |

После выполнения подпрограммы в ячейке с именем VALTYP будет находиться код типа числа, находящегося в DAC.

### 1.6.7. Преобразование чисел для вывода на печать

Для преобразования из строки в число используется подпрограмма с именем

FIN (3299h)

- Аргументы:
  - HL — адрес строки символов,
  - A — первый символ строки.
- Результаты:
  - DAC — вещественное число,
    - C —
      - 0 — была десятичная точка,
      - FFh — десятичной точки не было,
    - B — количество цифр после десятичной точки,
    - D — количество цифр в числе.

Для вывода имеются две подпрограммы:

- FOUT (3425h) — неформатный вывод,
- PUFOUT (3426h) — форматный вывод.

Эти подпрограммы преобразуют число, находящееся в DAC, в строку символов.

Аргументы:

- A - формат; содержимое его битов может быть следующим:
  - bit7: если 1, то вывод осуществляется по формату;
  - bit6: если 1, то через каждые 3 цифры вставляются запятые;
  - bit5: если 1, то первые нули заменить на символ «\*»;
  - bit4: если 1, то перед числом вставить символ «\$»;
  - bit3: если 1, то число выводится всегда со знаком;
  - bit2: если 1, то вставить знак после числа;
  - bit1: не используется;

- bit0:
  - если 0, то число выводится с фиксированной точкой;
  - если 1, то число выводится с плавающей точкой;
- В — количество цифр перед точкой;
- С — количество цифр после точки + 1.

**Результаты:** HL — начальный адрес строки символов.

**Пример 11.** Вычисление косинуса вещественного числа.

[1060-11.bas](#)

 [1060-11.bas](#)

```

10 CLEAR 200,&HD000:DEFUSR=&HD000:A=&HD000
20 A=A+1:READ R$:IF R$="z" THEN 40
30 POKE A,VAL("&h"+R$):GOTO 20
40 M=USR(0)
50 DATA CD,B4,00 : 'CALL 00B4h      ;Ввод строки с клавиатуры
60 DATA 23       : 'INC HL
70 DATA 7E       : 'LD A,(HL)
80 DATA CD,99,32 : 'CALL 3299h      ;Преобразование из строки в число
90 DATA 3E,08    : 'LD A,08h
100 DATA 32,63,F6 : 'LD (F663h),A ;VALTYP:=8
110 DATA CD,93,29 : 'CALL 2993h     ;Нахождение функции COS( )
120 DATA CD,25,34 : 'CALL 3425h     ;Преобразование числа в строку
130 DATA 06,11   : 'LD B,11h
140 DATA 7E      : 'LD A,(HL)
150 DATA 23      : 'INC HL
160 DATA CD,A2,00 : 'CALL 00A2h     ;Вывод символа на монитор |
170 DATA 10,F9   : 'DJNZ $-5       ;Конец цикла      └──┘
180 DATA C9      : 'RET
190 DATA "z"

```

Остальные подпрограммы перечислены ниже:

| Имя подпрограммы | Адрес | Выполняемые действия                  |
|------------------|-------|---------------------------------------|
| FOUTB            | 371Ah | Целое число → в двоичный вид          |
| FOUTO            | 371Eh | Целое число → в восьмеричный вид      |
| FOUTH            | 3722h | Целое число → в шестнадцатеричный вид |

Для этих подпрограмм:

**Аргументы:**

- DAC — целое число,
- VALTYP = 2.

**Результаты:**

- HL — начальный адрес строки.



[https://sysadminmosaic.ru/msx/basic\\_dialogue\\_programming\\_language/106](https://sysadminmosaic.ru/msx/basic_dialogue_programming_language/106)

2023-02-19 16:27

