**Московский государственный технический университет им.Н.Э.Баумана**
**Калужский филиал**
**Приборостроительный факультет**
**Кафедра П2-КФ**

# Отчет

по лабораторной работе № 1

***«Реализация полиномиальной функции на языке ассемблера с использованием команд сопроцессора»***

по курсу

***«Теория и проектирование ЭВМ и ВС»***

Студент : Комиссаров А.В., гр.ЭВМ-102

Преподаватель : доцент Максимов А.В.

Калуга, 1999 г.

# 1. Задание на лабораторную работу

Вариант по списку - **№25** :

$f(x) = \csc x = 1 / \sin x;$
$|x_{max}| = 3\pi / 4.$

Дополнительные указания :

- число разрядов АЦП и ЦАП при моделировании N = 16;
- необходимо взять первые четыре ненулевых члена разложения функции в ряд Маклорена.

Требуется :

- написать и отладить программу на языке ассемблера с использованием команд процессора и математического сопроцессора;
- сравнить оба варианта по точности (построить графики результата и ошибки), по скорости выполнения, по объему используемой памяти. Определить фактическую максимальную глубину регистрового стека при использовании команд FPU.

# 2. Математический алгоритм и масштабные соотношения

Разложение исходной функции косеканса в ряд Маклорена, в котором взяты только первые четыре члена, имеет вид :

$$f(x) \approx \frac{1}{x} + \frac{1}{6}x + \frac{7}{360}x^3 + \frac{31}{15120}x^5. \qquad (2.1)$$

Разложение функции в ряд сходится на всем рассматриваемом интервале $x \in \left[-\frac{3\pi}{4}; \frac{3\pi}{4}\right]$ за исключением точки $x = 0$, в которой данная функция является неограниченной.

При выполнении лабораторной работы в предыдущем семестре (разрабатывался вариант реализации функции с использованием обычных команд процессора) был выбран следующий математический алгоритм :

$$f(x) \approx \frac{1}{x} + \frac{x}{6} \cdot \left[1 + \frac{x^2}{60} \cdot \left(7 + \frac{31}{42} \cdot x^2\right)\right]. \qquad (2.1.1)$$

$u_1 = \frac{1}{x}; \quad u_2 = \frac{x}{6}; \quad u_3 = x \cdot x; \quad u_4 = \frac{u_3}{60}; \quad u_5 = \frac{31}{42} \cdot u_3; \quad u_6 = 7 + u_5; \quad u_7 = u_4 \cdot u_6; \quad u_8 = 1 + u_7;$
$u_9 = u_2 \cdot u_8; \; y = u_{10} = u_1 + u_9.$

Машинный алгоритм для случая реализации с использованием обычных команд процессора, а также масштабные соотношения были получены при выполнении лабораторной работы №2 предыдущего семестра (см. отчет по упомянутой лабораторной работе). Здесь приведем лишь масштабные соотношения для аргумента и результата :

$$\mathcal{M}_X = 2^{-2};\quad M_X = 2^{-2}\cdot 2^{15} = 2^{13};$$

$$\mathcal{M}_Y = 2^{-14};\quad M_Y = 2^{-14}\cdot 2^{15} = 2^{1}.$$

Машинное представление констант (с учетом соответствующих константам двоичных масштабов) имеет следующий вид :

$$\overline{1} = 4000h; \overline{6} = 6000h; \overline{60} = 7800h; \overline{\left(\frac{31}{42}\right)} = 5E79h; \overline{7} = 7000h.$$

## 3. Листинг программы с использованием обычных команд процессора

```
1                               dosseg
2       0000                    .model  tiny
3       0000                    .code
4                               .486
5       0000                    SEGMENT1:
6       0000  B8 0000s                       mov     ax, @code
7       0003  8E D8                         mov     ds, ax
8       0005                    PROG:
9                                           ;u1 = 1/x
10      0005  B8 4000                       mov     ax, const1
11      0008  99                            cwd
12      0009  8B 1E    00A2r                mov     bx, X
13      000D  F7 FB                         idiv    bx
14      000F  A3 00A6r                      mov     u1, ax
15
16                                          ;u2 = x/6
17      0012  33 C0                         xor     ax, ax
18      0014  8B 16    00A2r                mov     dx, X
19      0018  BB 6000 90                    mov     bx, const6
20      001C  D1 FA                         sar     dx, 1
21      001E  D1 D8                         rcr     ax, 1
22      0020  F7 FB                         idiv    bx
23      0022  A3 00A8r                      mov     u2, ax
24
25                                          ;u3 = x*x
26      0025  A1 00A2r                      mov     ax, X
27      0028  8B 1E    00A2r                mov     bx, X
28      002C  F7 EB                         imul    bx
29      002E  D1 E0                         sal     ax, 1
30      0030  D1 D2                         rcl     dx, 1
31      0032  D1 E0                         sal     ax, 1
32      0034  D1 D2                         rcl     dx, 1
33      0036  89 16    00AAr                mov     u3, dx
34
35                                          ;u4(cx)=u3/60
36      003A  33 C0                         xor     ax, ax
37      003C  8B 16    00AAr                mov     dx, u3
38      0040  BB 7800 90                    mov     bx, const60
39      0044  D1 FA                         sar     dx, 1
40      0046  D1 D8                         rcr     ax, 1
41      0048  F7 FB                         idiv    bx
42      004A  8B C8                         mov     cx, ax
43
44                                          ;u5=(31/42)*u3
45      004C  B8 5E79                       mov     ax, const31_42
46      004F  8B 1E    00AAr                mov     bx, u3
47      0053  F7 EB                         imul    bx
```

```
48      0055  D1 E0                           sal     ax, 1
49      0057  D1 D2                           rcl     dx, 1
50
51
52                                            ;u6(ax) = 7+u5=7+[dx]
53      0059  B8 7000                         mov     ax, const7
54      005C  D1 F8                           sar     ax, 1
55      005E  D1 FA                           sar     dx, 1
56      0060  03 C2                           add     ax, dx
57
58                                            ;u7(dx) = u4(cx)*u6
59      0062  F7 E9                           imul    cx
60      0064  D1 E0                           sal     ax, 1
61      0066  D1 D2                           rcl     dx, 1
62
63                                            ;u8(ax) = 1+u7(dx)
64      0068  B8 4000                         mov     ax, const1
65      006B  D1 F8                           sar     ax, 1
66      006D  D1 FA                           sar     dx, 1
67      006F  03 C2                           add     ax, dx
68
69                                            ;u9(dx)=u2*u8(ax)
70      0071  8B 1E    00A8r                  mov     bx, u2
71      0075  F7 EB                           imul    bx
72      0077  D1 E0                           sal     ax, 1
73      0079  D1 D2                           rcl     dx, 1
74      007B  D1 E0                           sal     ax, 1
75      007D  D1 D2                           rcl     dx, 1
76
77                                            ;u10 =   u1 + u9(dx)
78
79      007F  A1 00A6r                        mov     ax, u1
80      0082  83 FA    00                     cmp     dx, 0
81      0085  7D 08    90 90                  jge      AboveZero
82      0089  B9 FFFF                         mov     cx, 0FFFFh
83      008C  EB 03    90                     jmp     Next
84      008F                          AboveZero:
85      008F  33 C9                           xor     cx, cx
86      0091                          Next:
87      0091  D1 E2                           sal     dx, 1
88      0093  D1 D1                           rcl     cx, 1
89      0095  D1 E2                           sal     dx, 1
90      0097  D1 D1                           rcl     cx, 1
91
92
93      0099  03 C1                           add     ax, cx
94      009B  A3 00A4r                        mov     Y, ax
95
96
97      009E                          FINISH:
98      009E  B4 4C                           mov     ah, 4ch
99      00A0  CD 21                           int     21h
100     00A2                          DATA_:
101
102             =4000                         const1          equ     4000h
103             =6000                         const6          equ     6000h
104             =7800                         const60         equ     7800h
105             =5E79                         const31_42      equ     5E79h
106             =7000                         const7          equ     7000h
107     00A2  0001                            X               dw      0001h ;4B33h
108     00A4  ????                            Y               dw      ?
109     00A6  ????                            u1              dw      ?
110     00A8  ????                            u2              dw      ?
111     00AA  ????                            u3              dw      ?
112
113
114                                   END     SEGMENT1
```

# 4. Листинг программы с использованием команд блока FPU

```
1                                       dosseg
2       0000                            .model  tiny
3       0000                            .code
4                                       .486
5       0000                            SEGMENT1:
6       0000  B8 0000s                                mov   ax, @code
7       0003  8E D8                                   mov   ds, ax
8       0005                            PROG:
9       0005  9B DB   E3                              finit
10                                                    ;calc 31/42
11      0008  DB 2E   0100r                           fld       const31
12      000C  DB 2E   010Ar                           fld       const42
13      0010  DE F9                                   fdivp     st(1),st
14      0012  DD D1                                   fst       st(1)
15      0014  DB 3E   0114r                           fstp      const31_42
16                                                    ;calc the arg physval
17      0018  DF 06    008Er                          fild      X
18      001C  DB 2E   00CEr                           fld       Mx
19      0020  DE F9                                   fdivp     st(1),st            ;X/Mx, result  in st
20                                                    ;now calculates the approximation
21                                                    ;store    the X
22      0022  DD D1                                   fst       st(1)
23      0024  DB 3E   009Cr                           fstp      XFP
24                                                    ;u1 = 1/x
25      0028  D9 E8                                   fld1                          ;1 now is in the top, st(1) = x
26      002A  D8 F1                                   fdiv      st,st(1)            ;result in st
27      002C  DD D1                                   fst       st(1)
28      002E  DB 3E   00A6r                           fstp      u1
29                                                    ;u2=x/6
30      0032  DB 2E   00E2r                           fld       Six                 ;6 in the top, st(1)=u1
31      0036  DB 2E   009Cr                           fld       XFP                 ;X=st, 6=st(1), st(2)=u1
32      003A  D8 F1                                   fdiv      st,st(1)            ;st=u2
33      003C  DD D1                                   fst       st(1)
34      003E  DB 3E   00B0r                           fstp      u2
35                                                    ;u3=x*x
36      0042  DB 2E   009Cr                           fld       XFP
37      0046  DB 2E   009Cr                           fld       XFP
38      004A  DE C9                                   fmulp     st(1),st            ;st=u3
39                                                    ;
40      004C  DD D1                                   fst       st(1)                ;u3 in st(0),st(1)
41                                                    ;u4=u3/60
42      004E  DB 2E   00ECr                           fld       Sixty               ;60=st,st(1,2)=u3
43      0052  DE F9                                   fdivp     st(1),st            ;st=u4,st(1)=u3
44      0054  DB 3E   00C4r                           fstp      u4                  ;st=u3
45                                                    ;u5=(31/42)*u3
46      0058  DB 2E   0114r                           fld       const31_42          ;st=31/42,st(1)=u3
47      005C  DE C9                                   fmulp     st(1),st
48                                                    ;u6=7+u5
49      005E  DB 2E   00F6r                           fld       Seven
50      0062  DE C1                                   faddp     st(1),st            ;st=u6
51                                                    ;u7=u4*u6
52      0064  DB 2E   00C4r                           fld       u4
53      0068  DE C9                                   fmulp     st(1),st            ;st=u7
54                                                    ;u8=1+u7
55      006A  D9 E8                                   fld1
56      006C  DE C1                                   faddp     st(1),st            ;st=u8
57                                                    ;u9=u2*u8
```

```
58      006E  DB 2E   00B0r                  fld     u2
59      0072  DE C9                          fmulp   st(1),st        ;st=u9
60                                           ;y=u10=u1+u9
61      0074  DB 2E   00A6r                  fld     u1
62      0078  DE C1                          faddp   st(1),st
63                                           ;calc the mach  res
64      007A  DD D1                          fst     st(1)
65      007C  DB 3E   0092r                  fstp    YFP
66      0080  DB 2E   00D8r                  fld     My
67      0084  DE C9                          fmulp   st(1),st
68      0086  DF 16   0090r                  fist    Y
69      008A                  FINISH:
70      008A  B4 4C                          mov     ah, 4ch
71      008C  CD 21                          int     21h
72      008E                  DATA_:
73                                           ;---variables---
74      008E  B49B                           X               dw      -19301;4B33h 0001h
75      0090  ????                           Y               dw      ?
76      0092  ??????????????????????       YFP             dt      ?
77      009C  ??????????????????????       XFP             dt      ?
78      00A6  ??????????????????????       u1              dt      ?
79      00B0  ??????????????????????       u2              dt      ?
80      00BA  ??????????????????????       u3              dt      ?
81      00C4  ??????????????????????       u4              dt      ?
82                                           ;---scales---
83      00CE  400C8000000000000000           Mx              dt      8192.0
84      00D8  40008000000000000000           My              dt      2.0
85                                           ;---constants---
86      00E2  4001C000000000000000           Six             dt      6.0
87      00EC  4004F000000000000000           Sixty           dt      60.0
88      00F6  4001E000000000000000           Seven           dt      7.0
89      0100  4003F800000000000000           const31  dt     31.0
90      010A  4004A800000000000000           const42  dt     42.0
91      0114  ??????????????????????       const31_42      dt      ?
92
93                                  END     SEGMENT1
```

# 5. Сопоставление вариантов

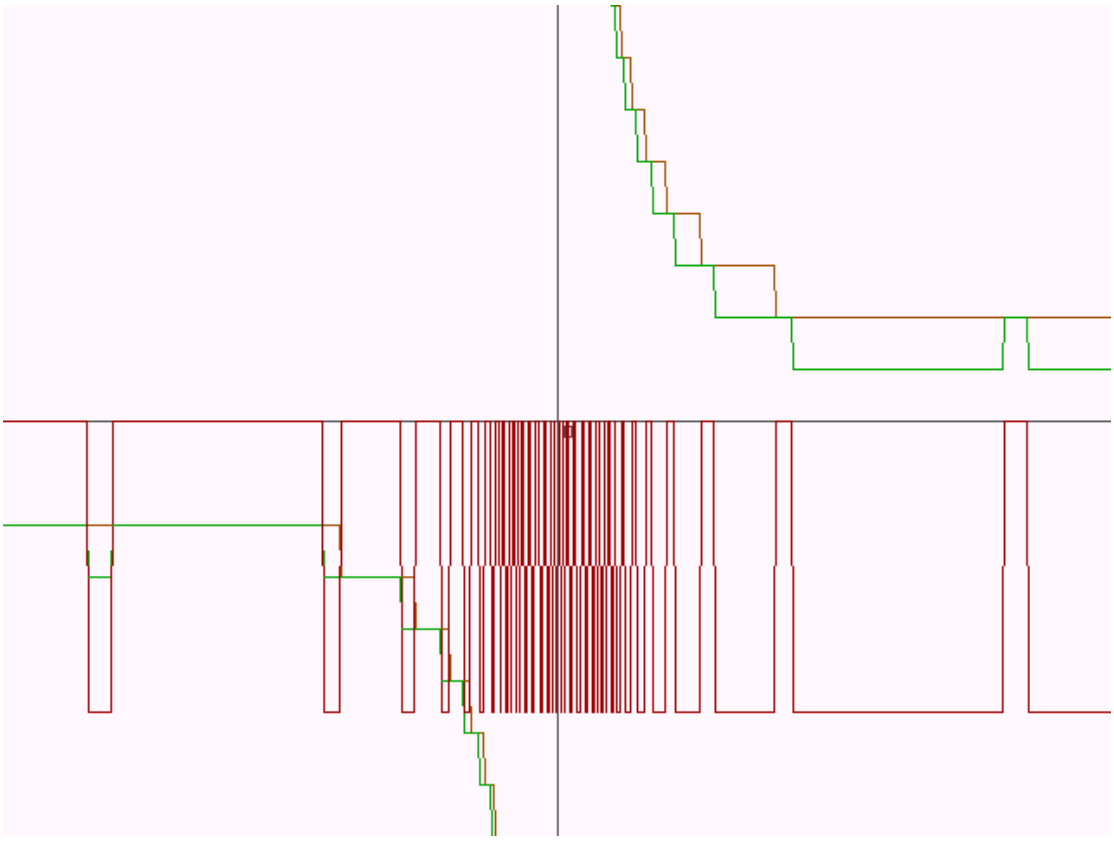
Из анализа приведенных листингов и результатов исполнения программ видно, что :

- Вариант с использованием команд сопроцессора FPU занимает меньше места в памяти машины (соответственно 162 и 142 байт; если не учитывать код, вычисляющий константу 31/42, то объем занимаемой памяти в случае команд FPU составит 126 байт);

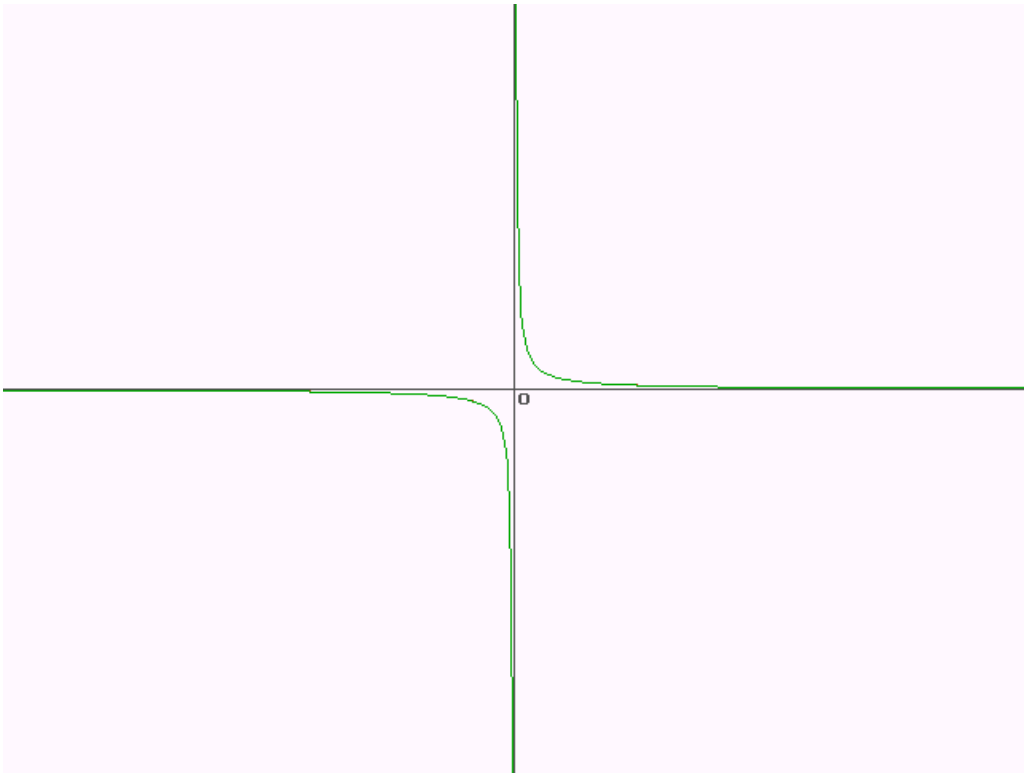
- Вариант с использованием FPU дает более точный результат за счет повышения точности промежуточных вычислений за счет большего числа разрядов, отводимых под представление мантиссы (64 информационных разряда мантиссы в случае использования регистров FPU, 15 информационных разрядов формата с фиксированной запятой - в случае использования регистров CPU). Ошибка вычислялась по формуле

$$E_i = \left| y_i - f_i \right|, \qquad (2.1)$$

где $y_i$ - текущее значение реализуемой с использованием обычных регистров команд CPU, а $f_i$ - текущее значение эталонной функции (график построен для случая машинных значений, а мат.ожидание и дисперсия вычислены для физич.значений). Ниже приводятся графики реализации полинома для обоих случаев, а также график ошибки (в качестве эталонного взято значение реализации функции на FPU) :



**Максимальное по модулю значение функции равно 4. Полный интервал изменения аргумента. В нижней части также показан график ошибки**



**Максимальное по модулю значение функции равно 8193,42. Интервал изменения аргумента сокращен в 96 раз**

Математическое ожидание ошибки реализации можно подсчитать по формуле :

$$M[E] = \frac{\sum_i E_i}{2n}, \qquad (2.2)$$

После прогона программы получаем значение математического ожидания ошибки, приблизительно равное -0,25.

Дисперсию (квадрат среднеквадратического отклонения) ошибки реализации можно подсчитать по формуле :

$$D[E] = M[E^2] - M^2[E]. \qquad (2.3)$$

После прогона программы получаем значение дисперсии ошибки, приблизительно равное 0,0625.

- Вариант с использованием FPU выполняется медленнее, чем вариант с использованием обычных команд CPU. Эксперимент, проведенный с использованием программы, разработанной с средств среды программирования Delphi в системе Windows[*], показал, что при прогоне ассемблерного блока в цикле 2 000 000 раз получаются приблизительно следующие результаты (число экспериментов = 10) :

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| CPU | 4,67 с | 4,67 с | 4,72 с | 4,72 с | 4,66 с | 4,73 с | 4,67 с | 4,67 с | 4,67 с | 4,67 с |
| FPU | 7,69 с | 7,69 с | 7,68 с | 7,69 с | 7,63 с | 7,69 с | 7,69 с | 7,63 с | 7,69 с | 7,69 с |

Средние значения в этом случае соответственно равны 4,69 и 7,68 сек. Следовательно, алгоритм, использующий FPU, работает в среднем в около 1,64 раза медленнее.

- Максимальная глубина стека при реализации на FPU (т.е. количество непустых регистров стека) составляет 4.

[*] Тестирование проводилось на одном и том же значении аргумента