

Университет ИТМО

## **Лабораторная работа №5**

Выполнил:  
студент III курса группы 3125  
Припадчев Артём

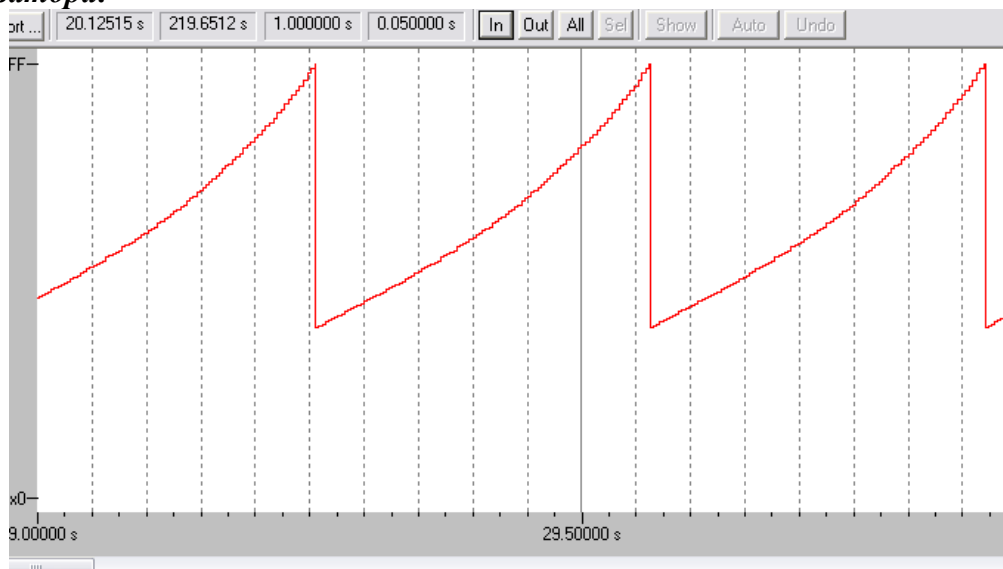
Проверит:  
Тропченко А.А.

Санкт-Петербург  
2014

## Программа вычисления функции тангенса с плавающей точкой на C51:

```
#include <reg51.h>
#include <math.h>
float x;
unsigned char xdata y[100],i;
main()
{
i=0;
for(x=0;x<1;x+=0.01)
y[i++]=P3=tan(x)*100+100;
while(1)
{
i=0;
for(x=0; x<1; x+=0.01)
{tan(x);P3=y[i++];}
}
}
```

### Окно анализатора:



Объем программы: 210

Время вычисления одного значения:  $(0,640047 - 0,460797)/50 = 0,003585 \text{ с} = 3,59 \text{ мс}$

### Листинг программы:

0000 E4		CLR	A				
0001 900000	R	MOV	DPTR,#i			; SOURCE LINE # 9	
0004 F0		MOVX	@DPTR,A		0021 AF00	R	MOV R7,x+03H
					0023 AE00	R	MOV R6,x+02H
					0025 AD00	R	MOV R5,x+01H
					0027 AC00	R	MOV R4,x
					0029 120000	E	LCALL _tan
					002C E4		CLR A
					002D FB		MOV R3,A
					002E FA		MOV R2,A
					002F 79C8		MOV R1,#0C8H
					0031 7842		MOV R0,#042H
					0033 120000	E	LCALL ?C?FPMUL
					0036 E4		CLR A
					0037 FB		MOV R3,A
					0038 FA		MOV R2,A
					0039 79C8		MOV R1,#0C8H
					003B 7842		MOV R0,#042H
					003D 120000	E	LCALL ?C?FPADD
0005 F500	R	MOV	x+03H,A				
0007 F500	R	MOV	x+02H,A				
0009 F500	R	MOV	x+01H,A				
000B F500	R	MOV	x,A				
000D E4		CLR	A				
000E FF		MOV	R7,A				
000F FE		MOV	R6,A				
0010 7D80		MOV	R5,#080H				
0012 7C3F		MOV	R4,#03FH				
0014 AB00	R	MOV	R3,x+03H				
0016 AA00	R	MOV	R2,x+02H				
0018 A900	R	MOV	R1,x+01H				
001A A800	R	MOV	R0,x				
001C 120000	E	LCALL	?C?FPCMP3				

```

0040 120000      E      LCALL   ?C?CASTF          0087 7C3F              MOV      R4,#03FH
0043 8FB0        MOV      P3,R7          0089 AB00              R      MOV      R3,x+03H
0045 900000      R      MOV      DPTR,#i     008B AA00              R      MOV      R2,x+02H
0048 E0          MOVX     A,@DPTR        008D A900              R      MOV      R1,x+01H
0049 FE          MOV      R6,A          008F A800              R      MOV      R0,x
004A 04          INC      A              0091 120000           E      LCALL   ?C?FPCMP3
004B F0          MOVX     @DPTR,A       0094 50DF              JNC     ?C0004
004C 7400      R      MOV      A,#LOW y
004E 2E          ADD      A,R6          SOURCE LINE # 14
004F F582      MOV      DPL,A        0096 AF00              R      MOV      R7,x+03H
0051 E4          CLR      A              0098 AE00              R      MOV      R6,x+02H
0052 3400      R      ADDC     A,#HIGH     009A AD00              R      MOV      R5,x+01H
0054 F583      MOV      DPH,A        009C AC00              R      MOV      R4,x
0056 EF          MOV      A,R7          009E 120000           E      LCALL   _tan
0057 F0          MOVX     @DPTR,A       00A1 900000           R      MOV      DPTR,#i
0058 AF00      R      MOV      R7,x+03H   00A4 E0              MOVX     A,@DPTR
005A AE00      R      MOV      R6,x+02H   00A5 FF              MOV      R7,A
005C AD00      R      MOV      R5,x+01H   00A6 04              INC      A
005E AC00      R      MOV      R4,x       00A7 F0              MOVX     @DPTR,A
0060 7B0A      MOV      R3,#0AH      00A8 7400              R      MOV      A,#LOW y
0062 7AD7      MOV      R2,#0D7H     00AA 2F              ADD      A,R7
0064 7923      MOV      R1,#023H     00AB F582              MOV      DPL,A
0066 783C      MOV      R0,#03CH     00AD E4              CLR      A
0068 120000     E      LCALL   ?C?FPADD     00AE 3400              R      ADDC     A,#HIGH y
006B 8F00      R      MOV      x+03H,R7   00B0 F583              MOV      DPH,A
006D 8E00      R      MOV      x+02H,R6   00B2 E0              MOVX     A,@DPTR
006F 8D00      R      MOV      x+01H,R5   00B3 F5B0              MOV      P3,A
0071 8C00      R      MOV      x,R4        00B5 AF00              R      MOV      R7,x+03H
0073 8098      SJMP    ?C0001        00B7 AE00              R      MOV      R6,x+02H
                                ;          00B9 AD00              R      MOV      R5,x+01H
                                ;          00BB AC00              R      MOV      R4,x
                                ;          00BD 7B0A              MOV      R3,#0AH
                                ;          00BF 7AD7              MOV      R2,#0D7H
                                ;          00C1 7923              MOV      R1,#023H
                                ;          00C3 783C              MOV      R0,#03CH
                                ;          00C5 120000           E      LCALL   ?C?FPADD
0075 E4          CLR      A              00C8 8F00              R      MOV      x+03H,R7
0076 900000     R      MOV      DPTR,#i     00CA 8E00              R      MOV      x+02H,R6
0079 F0          MOVX     @DPTR,A       00CC 8D00              R      MOV      x+01H,R5
                                ;          00CE 8C00              R      MOV      x,R4
                                ;          00D0 80B0              SJMP    ?C0006
                                ;          ; FUNCTION main (END)
                                ;          CODE SIZE      =      210
007A F500      R      MOV      x+03H,A     0085 7D80              MOV      R5,#080H
007C F500      R      MOV      x+02H,A
007E F500      R      MOV      x+01H,A
0080 F500      R      MOV      x,A
0082 E4          CLR      A
0083 FF          MOV      R7,A
0084 FE          MOV      R6,A
0085 7D80      MOV      R5,#080H

```

### Программа вычисления функции тангенса с фиксированной точкой на C51 (от 0 до 1):

```

#include <reg51.h>
char x;
unsigned char xdata y[100],i;
main()
{
while(1)
{
i=0;
for(x=0; x<100; x+=1)
{
y[i]=x*(100+(x*x*((100/3)+(x*x*(2*100/15)/100/100))/100/100))/100;
P3=y[i++];}}
}

```

## Окно анализатора:



Объём программы: 167

Время вычисления: 0,0015399 с = 1,54 мс

## Листинг программы:

; SOURCE LINE # 8				0032 FA	MOV	R2,A	
0000 E4	CLR	A	0033 E500	R	MOV	A,x	
0001 900000	R	MOV	DPTR,#i	0035 FD	MOV	R5,A	
0004 F0	MOVX	@DPTR,A	0036 33	RLC	A		
; SOURCE LINE # 9				0037 95E0	SUBB	A,ACC	
0005 F500	R	MOV	x,A	0039 FC	MOV	R4,A	
; SOURCE LINE # 10				003A AF00	R	MOV	R7,x
; SOURCE LINE # 11				003C EF	MOV	A,R7	
0007 E500	R	MOV	A,x	003D 33	RLC	A	
0009 FD	MOV	R5,A	003E 95E0	SUBB	A,ACC		
000A 33	RLC	A	0040 FE	MOV	R6,A		
000B 95E0	SUBB	A,ACC	0041 120000	E	LCALL	?C?IMUL	
000D FC	MOV	R4,A	0044 AD03	MOV	R5,AR3		
000E AF05	MOV	R7,AR5	0046 AC02	MOV	R4,AR2		
0010 FE	MOV	R6,A	0048 120000	E	LCALL	?C?IMUL	
0011 120000	E	LCALL	?C?IMUL	004B 7C00	MOV	R4,#00H	
0014 7C00	MOV	R4,#00H	004D 7D64	MOV	R5,#064H		
0016 7D0D	MOV	R5,#0DH	004F 120000	E	LCALL	?C?SIDIV	
0018 120000	E	LCALL	?C?IMUL	0052 EF	MOV	A,R7	
001B 7D64	MOV	R5,#064H	0053 75F064	MOV	B,#064H		
001D 120000	E	LCALL	?C?SIDIV	0056 120000	E	LCALL	?C?SCDIV
0020 EF	MOV	A,R7	0059 FF	MOV	R7,A		
0021 75F064	MOV	B,#064H	005A 33	RLC	A		
0024 120000	E	LCALL	?C?SCDIV	005B 95E0	SUBB	A,ACC	
0027 FF	MOV	R7,A	005D FE	MOV	R6,A		
0028 33	RLC	A	005E EF	MOV	A,R7		
0029 95E0	SUBB	A,ACC	005F 2464	ADD	A,#064H		
002B FE	MOV	R6,A	0061 FF	MOV	R7,A		
002C EF	MOV	A,R7	0062 E4	CLR	A		
002D 2421	ADD	A,#021H	0063 3E	ADDC	A,R6		
002F FB	MOV	R3,A	0064 FE	MOV	R6,A		
0030 E4	CLR	A	0065 AD00	R	MOV	R5,x	
0031 3E	ADDC	A,R6	0067 ED	MOV	A,R5		
			0068 33	RLC	A		
			0069 95E0	SUBB	A,ACC		
			006B FC	MOV	R4,A		
			006C 120000	E	LCALL	?C?IMUL	

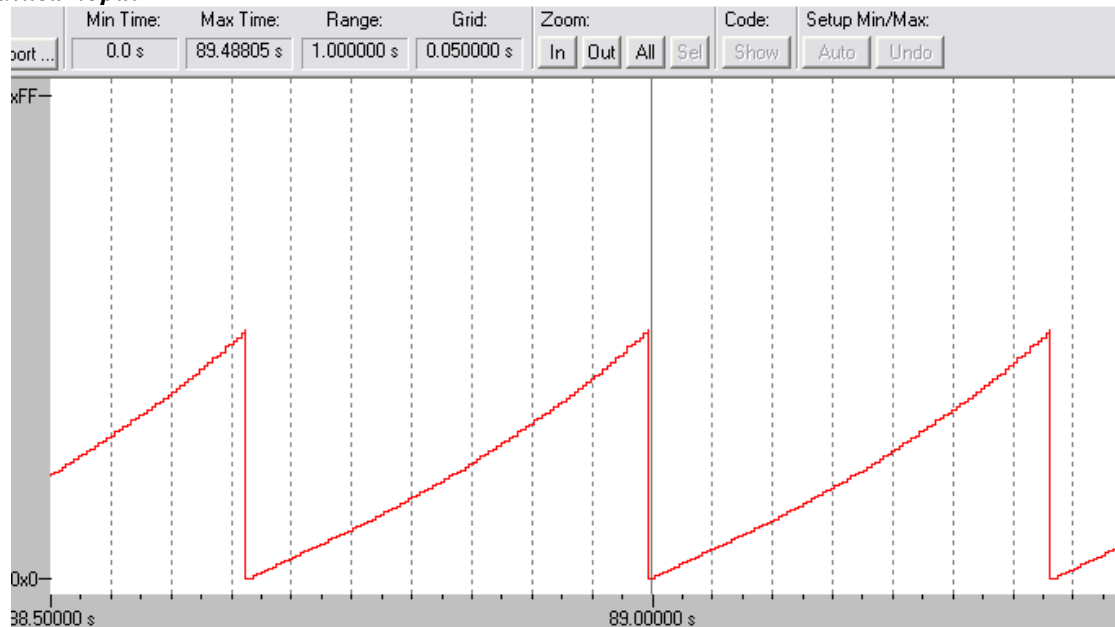
```

006F 7C00      MOV     R4,#00H      008B F0          MOVX   @DPTR,A
0071 7D64      MOV     R5,#064H     008C 7400        R      MOV   A,#LOW y
0073 120000    E      LCALL  ?C?SIDIV   008E 2F          ADD   A,R7
0076 900000    R      MOV   DPTR,#i    008F F582        MOV   DPL,A
0079 E0         MOVX   A,@DPTR      0091 E4          CLR   A
007A 2400      R      ADD   A,#LOW y    0092 3400        R      ADDC  A,#HIGH y
007C F582      MOV   DPL,A         0094 F583        MOV   DPH,A
007E E4        CLR   A             0096 E0          MOVX  A,@DPTR
007F 3400      R      ADDC  A,#HIGH y  0097 F5B0        MOV   P3,A
0081 F583      MOV   DPH,A
0083 EF        MOV   A,R7
0084 F0        MOVX  @DPTR,A
; SOURCE LINE # 12
0085 900000    R      MOV   DPTR,#i
0088 E0        MOVX  A,@DPTR
0089 FF        MOV   R7,A
008A 04        INC   A
; SOURCE LINE # 13
0099 0500      R      INC   x
009B E500      R      MOV   A,x
009D 6464      XRL  A,#064H
009F 6003      JZ   $ + 5H
00A1 020000    R      LJMP  ?C0003
00A4 020000    R      LJMP  ?C0001
; FUNCTION main (END)

```

### Программа на А51 (с фиксированной точкой)

#### Окно анализатора:



Объём программы: 403

Время вычисления одного значения:  $0,339/100=0,00339$  с = 3,35 мс

```

Dseg at 100
Xseg at 0
Cseg at 0
//macro multiplying
multi MACRO
Mov a,r1
Mov b,r4
Mul ab
Mov r1,a           //r1 - Low
Mov r5,b           //r5 - buffer
Mov a,r2
Mov b,r4
Mul ab
clr c
Add a,r5           //r2 <- r1 (perenos)
Mov r2,a           //r2 - Medium
Mov a,r3
Addc a,b
Mov r3,a           //r3 - High
ENDM
//macro summing
sum macro
clr c
Mov a,r1
Add a,r4
Mov r1,a
Mov a,r2
Addc a,#0
Mov r2,a
Mov a,r3
Addc a,#0

```

```

Mov r3,a
ENDM

Jmp start

//dividing podprogramma
division:
Mov r6,#16
Cikl:
clr c
Mov b,r3 //r3 - High delimoe
Mov a,r4
Rr a //делитель делится на 2
для срвнения
Anl a,#0x7f
Xch a,b
Subb a,b
Jc belzero
Abovezero: //((S<<1)-A)>0
clr c
Mov a,r1
Rl a
Mov b,a //Saving C (r2 <- r1)
Anl a,#0xfe
Mov r1,a
Mov a,b
Anl a,#0x01
Mov b,a
Mov a,r2
Rl a
Mov r7,a //Saving C (r3 <- r2)
Anl a,#0xfe
Add a,b
Mov r2,a
Mov a,r7
Anl a,#0x01
Mov b,a
Mov a,r3
Rl a
Anl a,#0xfe
Add a,b
Mov r3,a //Number doubled
Mov b,r4 //Doing [(S<<1)]-A+1
Subb a,b
Mov r3,a
Mov a,r1
Inc a
Mov r1,a
jmp fincikl
belzero: //((S<<1)-A)<0
clr c
Mov a,r1
Rl a
Mov b,a //Saving C (r2 <-
r1)
Anl a,#0xfe
Mov r1,a
Mov a,b
Anl a,#0x01
Mov b,a
Mov a,r2
Rl a

Mov r7,a //Saving C (r3 <- r2)
Anl a,#0xfe
Add a,b
Mov r2,a
Mov a,r7
fincikl:
Djnz r6,cikl
clr a
Mov r3,a
ret

//Program body
start:clr a
clr c
Mov b,a
Mov r1,a
Mov r2,a
Mov r3,a
Mov r4,a
Mov r5,a
Mov r6,a
Mov r7,a
Mov p3,#0
Mov r0,#0 //maincikl counter
maincycle:
Mov r1,#62 //r1, r2, r3 - mnogimoe
Mov a,r0
Mov r4,a //r4 - mnogitel'
multi
multi
Mov r4,#100
lcall division //62*x*x/100
Mov a,r2
Mov r1,a
Mov a,r3
Mov r2,a
clr a
Mov r3,a
mov r4,#11
lcall division //62*x*x/100/2835
Mov r4,#6
sum //((1700/315)[~6]
+ (62*x*x/100/2835)
Mov a,r0
Mov r4,a
multi
multi
Mov r4,#100
lcall division
lcall division
Mov r4,#13
sum
Movx a,@dptr
Mov r4,a

```

```

multi
multi
Mov r4,#100
lcall division
lcall division
//((2*100/15)+(x*x*((17*100/315)+62*
x*x/100/2835))/100/100)
Mov r4,#34
sum
Mov a,r0
Mov r4,a
multi
multi
Mov r4,#100
lcall division
lcall division
Mov r4,#100
sum
Mov a,r0
Mov r4,a
multi
Mov r4,#100
lcall division
Mov P3,r1
Mov a,r0
Inc a
Mov r0,a
clr c
Subb a,#100
jz final
jmp maincycle
final:
jmp start
end

```