

**Санкт-Петербургский национальный
исследовательский университет
информационных технологий,
механики и оптики**

Кафедра информатики и прикладной математики

Организация ЭВМ и систем

Лабораторная работа 5



Старались: **Шкаруба Н.Е.**

Суханова В.А.

Проверил: **Тропченко А.А.**

Группа: **Р3318**

2016 г

Задание:

- 1) Для заданных функций разработать программу вычисления функции с плавающей точкой, вывести график, измерить среднее время вычисления одного значения и объем программы.
- 2) Разработать программы вычисления с фиксированной точкой в C51 и в ассемблере. Вычисление функции - в целых 8-разрядных числах в 29 диапазоне аргумента 0 – 1.0 и с использованием макрокоманд и подпрограмм. Измерить среднее время вычислений и объем программы. Привести график изменения этих параметров для трех рассмотренных способов (две программы в Си и одна в Ассемблере).

Функция:

$1/(1-x) \sim 1 + x + x^2 + x^3 + \dots$ $-1 < x < 1$ (сходимость ряда)

Программа в C51 (плавающая запятая):

```
#include <reg51.h>
#include <math.h>

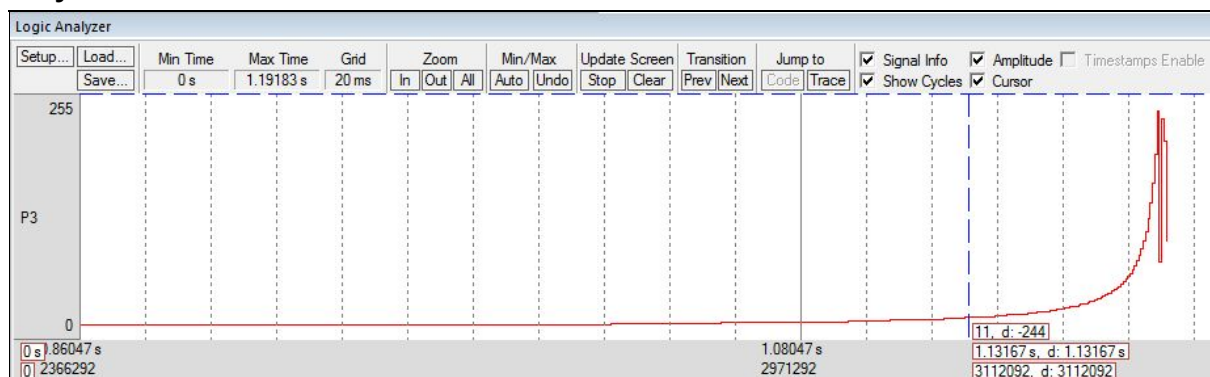
float x;
unsigned char xdata y[100],i;

main() {
    i = 0;
    for(x = -0.999; x < 1; x += 0.001)
        y[i++] = P3 = (1 / (1 - x)) * 100 + 100;

    while(1) {
        i = 0;

        for(x = -0.999; x < 1; x += 0.001) {
            1 / (1 - x);
            P3 = y[i++];
        }
    }
}
```

Результат выполнения:



Объём программы: 127, среднее время вычисления одного значения 1,13167 s.

Файл листинга (файл с расширением *.lst)

C51 COMPILER V9.56.0.0 L

11/06/2016 00:23:59 PAGE 1

C51 COMPILER V9.56.0.0, COMPILATION OF MODULE L

OBJECT MODULE PLACED IN .\Objects\l.obj

COMPILER INVOKED BY: C:\Keil_v5\C51\BIN\C51.EXE l.c OPTIMIZE(8,SIZE) BROWSE DEBUG

OBJECTEXTEND CODE PRINT(.\\Listings\\l.l

-st) TABS(2) OBJECT(.\\Objects\\l.obj)

line level source

```
1      #include <reg51.h>
2      #include <math.h>
3
4      float x;
5      unsigned char xdata y[100],i;
6
7      main() {
8 1      i = 0;
9 1      for(x = -0.999; x < 1; x += 0.001)
10 1      y[i++] = P3 = (1 / (1 - x));
11 1
12 1      /*while(1) {
13 1          i = 0;
14 1          for(x = -0.999; x < 1; x += 0.001) {
15 1              1 / (1 - x);
16 1              P3 = y[i++];}
17 1      }*/
18 1      }
```

*** WARNING C290 IN LINE 18 OF l.c: missing return value

C51 COMPILER V9.56.0.0 L

11/06/2016 00:23:59 PAGE 2

ASSEMBLY LISTING OF GENERATED OBJECT CODE

```
      ; FUNCTION main (BEGIN)
      ; SOURCE LINE # 7
      ; SOURCE LINE # 8
0000 E4      CLR  A
0001 900000  R   MOV  DPTR,#i
0004 F0      MOVX  @DPTR,A
      ; SOURCE LINE # 9
0005 750077  R   MOV  x+03H,#077H
0008 7500BE  R   MOV  x+02H,#0BEH
000B 75007F  R   MOV  x+01H,#07FH
000E 7500BF  R   MOV  x,#0BFH
0011      ?C0001:
0011 E4      CLR  A
0012 FF      MOV  R7,A
0013 FE      MOV  R6,A
0014 7D80      MOV  R5,#080H
0016 7C3F      MOV  R4,#03FH
0018 AB00  R   MOV  R3,x+03H
001A AA00  R   MOV  R2,x+02H
```

```

001C A900    R  MOV   R1,x+01H
001E A800    R  MOV   R0,x
0020 120000  E  LCALL ?C?FPCMP3
0023 5059    JNC   ?C0004
                ; SOURCE LINE # 10
0025 AB00    R  MOV   R3,x+03H
0027 AA00    R  MOV   R2,x+02H
0029 A900    R  MOV   R1,x+01H
002B A800    R  MOV   R0,x
002D E4      CLR   A
002E FF      MOV   R7,A
002F FE      MOV   R6,A
0030 7D80    MOV   R5,#080H
0032 7C3F    MOV   R4,#03FH
0034 120000  E  LCALL ?C?FPSUB
0037 A804    MOV   R0,AR4
0039 A905    MOV   R1,AR5
003B AA06    MOV   R2,AR6
003D AB07    MOV   R3,AR7
003F E4      CLR   A
0040 FF      MOV   R7,A
0041 FE      MOV   R6,A
0042 7D80    MOV   R5,#080H
0044 7C3F    MOV   R4,#03FH
0046 120000  E  LCALL ?C?FPDIV
0049 120000  E  LCALL ?C?CASTF
004C 8FB0    MOV   P3,R7
004E 900000  R  MOV   DPTR,#i
0051 E0      MOVX  A,@DPTR
0052 FE      MOV   R6,A
0053 04      INC   A
0054 F0      MOVX  @DPTR,A
0055 7400    R  MOV   A,#LOW y
0057 2E      ADD   A,R6
0058 F582    MOV   DPL,A
005A E4      CLR   A
005B 3400    R  ADDC  A,#HIGH y
005D F583    MOV   DPH,A
005F EF      MOV   A,R7
0060 F0      MOVX  @DPTR,A
C51 COMPILER V9.56.0.0  L

```

11/06/2016 00:23:59 PAGE 3

```

0061 AF00    R  MOV   R7,x+03H
0063 AE00    R  MOV   R6,x+02H
0065 AD00    R  MOV   R5,x+01H
0067 AC00    R  MOV   R4,x
0069 7B6F    MOV   R3,#06FH
006B 7A12    MOV   R2,#012H
006D 7983    MOV   R1,#083H
006F 783A    MOV   R0,#03AH
0071 120000  E  LCALL ?C?FPADD
0074 8F00    R  MOV   x+03H,R7
0076 8E00    R  MOV   x+02H,R6
0078 8D00    R  MOV   x+01H,R5
007A 8C00    R  MOV   x,R4
007C 8093    SJMP  ?C0001
                ; SOURCE LINE # 18

```

```

007E    ?C0004:
007E 22      RET
        ; FUNCTION main (END)

```

MODULE INFORMATION: STATIC OVERLAYABLE

```

CODE SIZE    = 127  ---
CONSTANT SIZE = ---  ---
XDATA SIZE   = 101  ---
PDATA SIZE   = ---  ---
DATA SIZE    = 4    ---
IDATA SIZE   = ---  ---
BIT SIZE     = ---  ---

```

END OF MODULE INFORMATION.

C51 COMPILATION COMPLETE. 1 WARNING(S), 0 ERROR(S)

Программа в C51 (фиксированная запятая):

```

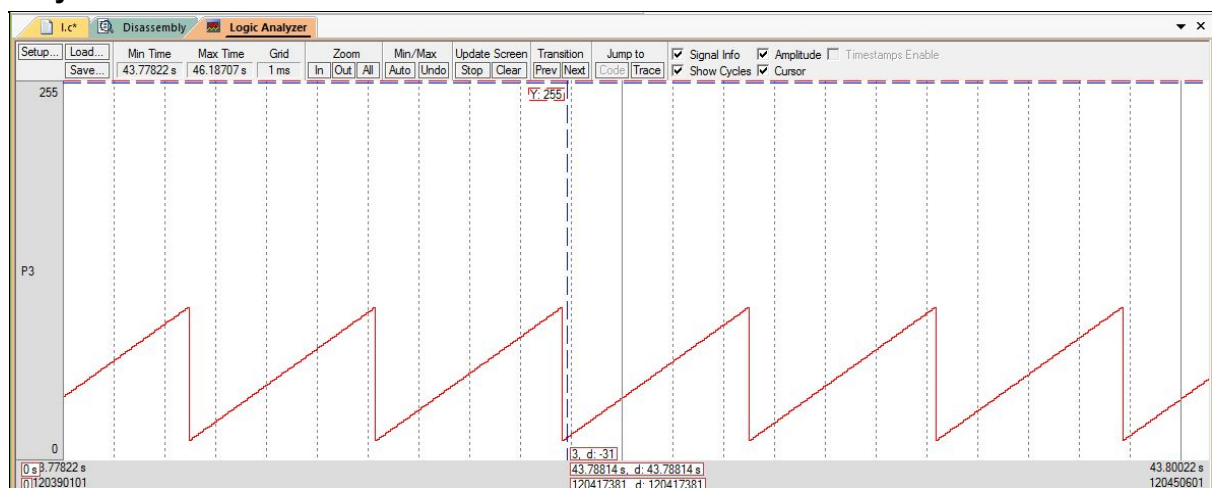
#include <reg51.h>

unsigned char y[100]; // Resulted array
unsigned char x;      // Argument of a function: 1 / (1 - x)
unsigned char i;      // Iterator

int main() {
    for(x = 0, i = 0; x < 100; x++, i++) {
        y[i] = 1 + x*(1+ (x*(1* x*(1* x/100)/100)/100)/100);
        P3 = y[i];
    }
    return 0;
}

```

Результат выполнения:



Объём программы: 127, среднее время вычисления одного значения 1,13511 s.

Файл листинга (файл с расширением *.lst)

C51 COMPILER V9.56.0.0, COMPILATION OF MODULE L3
OBJECT MODULE PLACED IN .\Objects\l3.obj
COMPILER INVOKED BY: C:\Keil_v5\C51\BIN\C51.EXE l3.c OPTIMIZE(8,SPEED)
BROWSE DEBUG OBJECTEXTEND CODE PRINT(.\\Listings\l
-3.lst) TABS(2) OBJECT(.\\Objects\l3.obj)

line	level	source
------	-------	--------

1		#include <reg51.h>
2		
3		char toLower(char letter) {
4	1	if (letter >= 'A' && letter <= 'Z') {
5	2	return letter - ('A' - 'a');
6	2	} else {
7	2	return letter;
8	2	}
9	1	}
10		
11		main() {
12	1	char testString[] = "This programmer";
13	1	//char code testString[] = "This programmer";
14	1	char testStringSize = 17;
15	1	char minChar;
16	1	int i, pivot = 0;
17	1	
18	1	// remove all spaces
19	1	for (i = 0; i < testStringSize; i++) {
20	2	if (testString[i] == ' ') {
21	3	testString[i] = testString[testStringSize - 1];
22	3	testString[testStringSize - 1] = 0;
23	3	testStringSize--;
24	3	}
25	2	}
26	1	
27	1	while(pivot != testStringSize) {
28	2	// search for a min
29	2	minChar = testString[pivot];
30	2	
31	2	for (i = pivot + 1; i < testStringSize; i++) {
32	3	if (toLower(testString[i]) < toLower(minChar)) {
33	4	minChar = testString[i];
34	4	}
35	3	}
36	2	
37	2	// swap min chars with chars at pivot index, increment pivot for each swap
38	2	for (i = pivot; i < testStringSize; i++) {

```

39 3      if (testString[i] == minChar) {
40 4      // swap
41 4      char buffer = testString[i];
42 4      testString[i] = testString[pivot];
43 4      testString[pivot] = buffer;
44 4      pivot++;
45 4      }
46 3      }
47 2      }
48 1      }

```

*** WARNING C290 IN LINE 48 OF I3.c: missing return value

C51 COMPILER V9.56.0.0 L3

10/22/2016

13:52:12 PAGE 2

ASSEMBLY LISTING OF GENERATED OBJECT CODE

```

; FUNCTION _toLower (BEGIN)
; SOURCE LINE # 3
;---- Variable 'letter' assigned to Register 'R7' ----
; SOURCE LINE # 4
0000 C3          CLR    C
0001 EF          MOV    A,R7
0002 6480        XRL    A,#080H
0004 94C1        SUBB   A,#0C1H
0006 400D        JC     ?C0001
0008 D3          SETB   C
0009 EF          MOV    A,R7
000A 6480        XRL    A,#080H
000C 94DA        SUBB   A,#0DAH
000E 5005        JNC    ?C0001
; SOURCE LINE # 5
0010 EF          MOV    A,R7
0011 2420        ADD    A,#020H
0013 FF          MOV    R7,A
0014 22          RET
; SOURCE LINE # 6
0015            ?C0001:
; SOURCE LINE # 7
; SOURCE LINE # 8
; SOURCE LINE # 9
0015            ?C0002:
0015 22          RET
; FUNCTION _toLower (END)

; FUNCTION main (BEGIN)
; SOURCE LINE # 11
; SOURCE LINE # 12
0000 7800    R    MOV    R0,#LOW testString
0002 7C00    R    MOV    R4,#HIGH testString

```

```

0004 7D00      MOV  R5,#00H
0006 7BFF      MOV  R3,#0FFH
0008 7A00  R    MOV  R2,#HIGH_?ix1000
000A 7900  R    MOV  R1,#LOW_?ix1000
000C 7E00      MOV  R6,#00H
000E 7F12      MOV  R7,#012H
0010 120000 E    LCALL ?C?COPY
                        ; SOURCE LINE # 14
;---- Variable 'testStringSize' assigned to Register 'R1' ----
0013 7911      MOV  R1,#011H
                        ; SOURCE LINE # 16
;---- Variable 'pivot' assigned to Register 'R4/R5' ----
0015 E4        CLR  A
0016 FD        MOV  R5,A
0017 FC        MOV  R4,A
                        ; SOURCE LINE # 19
;---- Variable 'i' assigned to Register 'R2/R3' ----
0018 FB        MOV  R3,A
0019 FA        MOV  R2,A
001A           ?C0004:
001A E9        MOV  A,R1
001B FF        MOV  R7,A
001C 33        RLC  A
001D 95E0      SUBB  A,ACC
001F FE        MOV  R6,A
0020 C3        CLR  C
C51 COMPILER V9.56.0.0 L3
13:52:12 PAGE 3

0021 EB        MOV  A,R3
0022 9F        SUBB  A,R7
0023 EE        MOV  A,R6
0024 6480      XRL  A,#080H
0026 F8        MOV  R0,A
0027 EA        MOV  A,R2
0028 6480      XRL  A,#080H
002A 98        SUBB  A,R0
002B 5022      JNC  ?C0008
                        ; SOURCE LINE # 20
002D 7400  R    MOV  A,#LOW testString
002F 2B        ADD  A,R3
0030 F8        MOV  R0,A
0031 E6        MOV  A,@R0
0032 B42013    CJNE  A,#020H,?C0006
                        ; SOURCE LINE # 21
0035 7400  R    MOV  A,#LOW testString+0FFFFH
0037 29        ADD  A,R1
0038 F8        MOV  R0,A
0039 E6        MOV  A,@R0
003A FF        MOV  R7,A

```

10/22/2016


```

003B 7400    R    MOV    A,#LOW testString
003D 2B      ADD    A,R3
003E F8      MOV    R0,A
003F A607    MOV    @R0,AR7
                ; SOURCE LINE # 22
0041 7400    R    MOV    A,#LOW testString+0FFFFH
0043 29      ADD    A,R1
0044 F8      MOV    R0,A
0045 E4      CLR    A
0046 F6      MOV    @R0,A
                ; SOURCE LINE # 23
0047 19      DEC    R1
                ; SOURCE LINE # 24
                ; SOURCE LINE # 25
0048          ?C0006:
0048 0B      INC    R3
0049 BB0001  CJNE   R3,#00H,?C0019
004C 0A      INC    R2
004D          ?C0019:
004D 80CB    SJMP   ?C0004
004F          ?C0008:
                ; SOURCE LINE # 27
004F E9      MOV    A,R1
0050 FF      MOV    R7,A
0051 33      RLC    A
0052 95E0    SUBB   A,ACC
0054 FE      MOV    R6,A
0055 EF      MOV    A,R7
0056 6D      XRL    A,R5
0057 7002    JNZ    ?C0020
0059 EE      MOV    A,R6
005A 6C      XRL    A,R4
005B          ?C0020:
005B 7003    JNZ    $ + 5H
005D 020000  R      LJMP ?C0018
                ; SOURCE LINE # 29
0060 7400    R    MOV    A,#LOW testString
0062 2D      ADD    A,R5
0063 F8      MOV    R0,A
0064 E6      MOV    A,@R0
0065 F500    R    MOV    minChar,A
C51 COMPILER V9.56.0.0 L3
13:52:12 PAGE 4

                ; SOURCE LINE # 31
0067 ED      MOV    A,R5
0068 2401    ADD    A,#01H
006A FB      MOV    R3,A
006B E4      CLR    A
006C 3C      ADDC   A,R4

```

10/22/2016

```

006D FA          MOV  R2,A
006E             ?C0010:
006E E9          MOV  A,R1
006F FF          MOV  R7,A
0070 33          RLC   A
0071 95E0        SUBB  A,ACC
0073 FE          MOV  R6,A
0074 C3          CLR   C
0075 EB          MOV  A,R3
0076 9F          SUBB  A,R7
0077 EE          MOV  A,R6
0078 6480        XRL   A,#080H
007A F8          MOV  R0,A
007B EA          MOV  A,R2
007C 6480        XRL   A,#080H
007E 98          SUBB  A,R0
007F 5029        JNC   ?C0011
                  ; SOURCE LINE # 32
0081 7400  R     MOV  A,#LOW testString
0083 2B          ADD  A,R3
0084 F8          MOV  R0,A
0085 E6          MOV  A,@R0
0086 FF          MOV  R7,A
0087 120000  R   LCALL _toLower
008A AE07        MOV  R6,R7
008C AF00  R     MOV  R7,minChar
008E 120000  R   LCALL _toLower
0091 C3          CLR   C
0092 EF          MOV  A,R7
0093 6480        XRL   A,#080H
0095 F8          MOV  R0,A
0096 EE          MOV  A,R6
0097 6480        XRL   A,#080H
0099 98          SUBB  A,R0
009A 5007        JNC   ?C0012
                  ; SOURCE LINE # 33
009C 7400  R     MOV  A,#LOW testString
009E 2B          ADD  A,R3
009F F8          MOV  R0,A
00A0 E6          MOV  A,@R0
00A1 F500  R     MOV  minChar,A
                  ; SOURCE LINE # 34
                  ; SOURCE LINE # 35
00A3             ?C0012:
00A3 0B          INC   R3
00A4 BB0001      CJNE  R3,#00H,?C0021
00A7 0A          INC   R2
00A8             ?C0021:
00A8 80C4        SJMP  ?C0010
00AA             ?C0011:

```

```

; SOURCE LINE # 38
00AA AA04      MOV  R2,AR4
00AC AB05      MOV  R3,AR5
00AE           ?C0014:
00AE E9        MOV  A,R1
00AF FF        MOV  R7,A
C51 COMPILER V9.56.0.0 L3
13:52:12 PAGE 5
10/22/2016

00B0 33        RLC  A
00B1 95E0      SUBB A,ACC
00B3 FE        MOV  R6,A
00B4 C3        CLR  C
00B5 EB        MOV  A,R3
00B6 9F        SUBB A,R7
00B7 EE        MOV  A,R6
00B8 6480      XRL  A,#080H
00BA F8        MOV  R0,A
00BB EA        MOV  A,R2
00BC 6480      XRL  A,#080H
00BE 98        SUBB A,R0
00BF 508E      JNC  ?C0008
; SOURCE LINE # 39
00C1 7400  R   MOV  A,#LOW testString
00C3 2B        ADD  A,R3
00C4 F8        MOV  R0,A
00C5 E6        MOV  A,@R0
00C6 FF        MOV  R7,A
00C7 B50019    R   CJNE A,minChar,?C0016
; SOURCE LINE # 41
00CA F500  R   MOV  buffer,A
; SOURCE LINE # 42
00CC 7400  R   MOV  A,#LOW testString
00CE 2D        ADD  A,R5
00CF F8        MOV  R0,A
00D0 E6        MOV  A,@R0
00D1 FF        MOV  R7,A
00D2 7400  R   MOV  A,#LOW testString
00D4 2B        ADD  A,R3
00D5 F8        MOV  R0,A
00D6 A607      MOV  @R0,AR7
; SOURCE LINE # 43
00D8 7400  R   MOV  A,#LOW testString
00DA 2D        ADD  A,R5
00DB F8        MOV  R0,A
00DC A600  R   MOV  @R0,buffer
; SOURCE LINE # 44
00DE 0D        INC  R5
00DF BD0001    CJNE R5,#00H,?C0022
00E2 0C        INC  R4

```

```

00E3      ?C0022:
                ; SOURCE LINE # 45
                ; SOURCE LINE # 46
00E3      ?C0016:
00E3 0B      INC    R3
00E4 BB0001  CJNE  R3,#00H,?C0023
00E7 0A      INC    R2
00E8      ?C0023:
00E8 80C4    SJMP  ?C0014
                ; SOURCE LINE # 47
                ; SOURCE LINE # 48
00EA      ?C0018:
00EA 22      RET
            ; FUNCTION main (END)

```

MODULE INFORMATION: STATIC OVERLAYABLE

```

CODE SIZE      =    257    ----
CONSTANT SIZE  =     18    ----
XDATA SIZE     =  ---- ----

```

C51 COMPILER V9.56.0.0 L3

10/22/2016

13:52:12 PAGE 6

```

PDATA SIZE     =  ---- ----
DATA SIZE      =  ----      20
IDATA SIZE     =  ---- ----
BIT SIZE       =  ---- ----

```

END OF MODULE INFORMATION.

C51 COMPILATION COMPLETE. 1 WARNING(S), 0 ERROR(S)

Программа в A51 (фиксированная запятая):

```

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

RI 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

RI a

Mov r7,a

//Saving C (r3 <

-

r2)

Anl a,#0

xfe

Add a,b

Mov r2,a

Mov a,r7

Anl a,#0x01

Mov b,a

Mov a,r3

RI 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 finciki
belzero:
//((S<<1)
-
A)<0
clr c
Mov a,r1
Rl a
Mov b,a
//Sav
ing 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
finciki:
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
//maincikli 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 di
vision
Mov r4,#13
sum
Movx a,@dptr
Mov r4,a

multi
multi

```

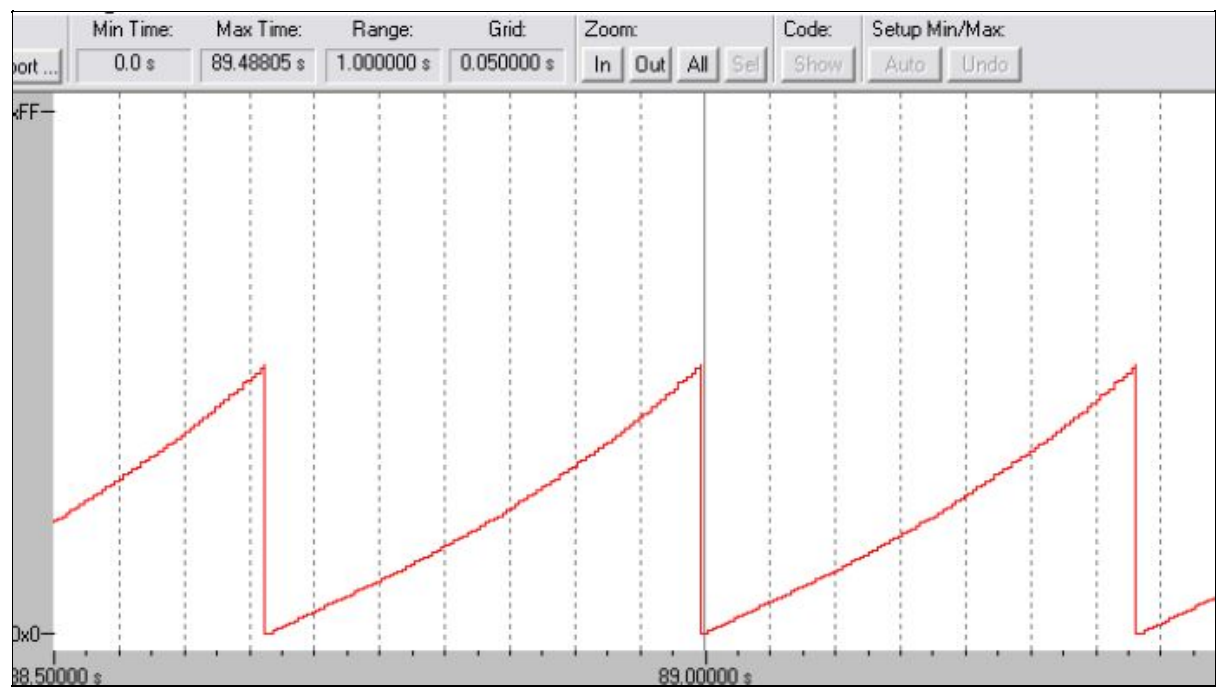


```

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

```

Результат выполнения:



Объём программы:403

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