

Санкт-Петербургский Национальный Исследовательский Университет
Информационных Технологий, Механики и Оптики

ПИиКТ

Лабораторная работа 4
по дисциплине
«Архитектура компьютера»

Выполнили: Студенты группы Р33113

Мансуров Б.Б.

Преподаватель: Тропченко А.Ю.

Санкт-Петербург

2020г

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 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).

Исходный текст программы на C51

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

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

```
double floatingPoint(double x) {  
    return log(x);  
}
```

```
void floatPointingWrapper() {  
    float x;  
    for (x = -1; x <= 1; x += 0.02) {  
        P3 = floatingPoint(x) * 100 + 100;  
    }  
}
```

```
unsigned char fixedPoint(unsigned char x, int accuracy) {  
    int result;  
    int i;  
    for(i = 1; i < accuracy; i++) {  
        result -= pow(x, accuracy) / i;  
    }  
    return result;  
}
```

```
void fixedPointWrapper() {  
    unsigned char x;
```

```

for(x = -100; x <= 100; x++) {

    P3 = fixedPoint(x, 6) + 100;

}

}

void main() {

    while(1) {

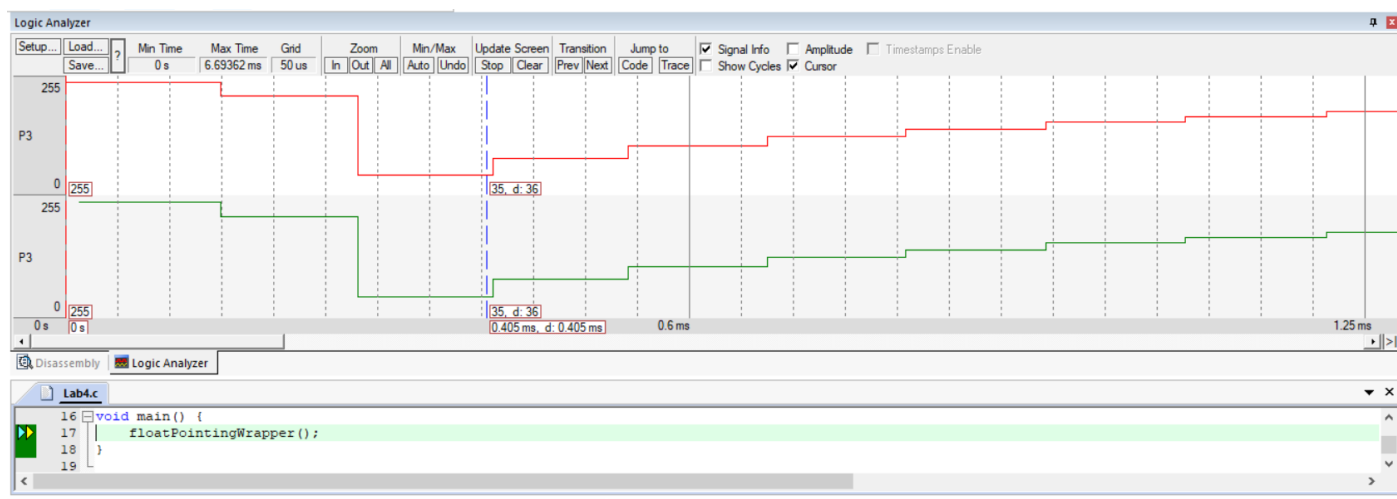
        //floatPointWrapper();

        fixedPointWrapper();

    }

}

```



Исходный текст программы на A51

ASSEMBLY LISTING OF GENERATED OBJECT CODE

```

; FUNCTION _fixedPoint (BEGIN)

; SOURCE LINE # 4

0000 8F00    R   MOV    x,R7

0002 8C00    R   MOV    accuracy,R4

0004 8D00    R   MOV    accuracy+01H,R5

; SOURCE LINE # 7

0006 750000  R   MOV    i,#00H

0009 750001  R   MOV    i+01H,#01H

000C      ?C0001:

000C C3      CLR    C

```

```

000D E500    R   MOV   A,i+01H
000F 9500    R   SUBB  A,accuracy+01H
0011 E500    R   MOV   A,accuracy
0013 6480          XRL  A,#080H
0015 F8          MOV  R0,A
0016 E500    R   MOV   A,i
0018 6480          XRL  A,#080H
001A 98          SUBB  A,R0
001B 5079          JNC  ?C0002
                ; SOURCE LINE # 8
001D AC00    R   MOV   R4,x
001F E4          CLR  A
0020 120000    E   LCALL ?C?FCASTC
0023 C004          PUSH AR4
0025 C005          PUSH AR5
0027 C006          PUSH AR6
0029 C007          PUSH AR7
002B AC00    R   MOV   R4,accuracy
002D AD00    R   MOV   R5,accuracy+01H
002F EC          MOV  A,R4
0030 120000    E   LCALL ?C?FCASTI
0033 8F00    E   MOV   ?_pow?BYTE+07H,R7
0035 8E00    E   MOV   ?_pow?BYTE+06H,R6
0037 8D00    E   MOV   ?_pow?BYTE+05H,R5
0039 8C00    E   MOV   ?_pow?BYTE+04H,R4
003B D007          POP  AR7
003D D006          POP  AR6
003F D005          POP  AR5
0041 D004          POP  AR4
0043 120000    E   LCALL _pow
0046 C004          PUSH AR4
0048 C005          PUSH AR5
004A C006          PUSH AR6
004C C007          PUSH AR7

```

004E AC00	R	MOV	R4,i
0050 AD00	R	MOV	R5,i+01H
0052 EC		MOV	A,R4
0053 120000	E	LCALL	?C?FCASTI
0056 A804		MOV	R0,AR4
0058 A905		MOV	R1,AR5
005A AA06		MOV	R2,AR6
005C AB07		MOV	R3,AR7
005E D007		POP	AR7
0060 D006		POP	AR6
0062 D005		POP	AR5
0064 D004		POP	AR4
0066 120000	E	LCALL	?C?FPDIV
0069 C004		PUSH	AR4
006B C005		PUSH	AR5

```
006D C006      PUSH  AR6
006F C007      PUSH  AR7
0071 AC00    R   MOV   R4,result
0073 AD00    R   MOV   R5,result+01H
0075 EC      MOV   A,R4
0076 120000  E   LCALL ?C?FCASTI
0079 D003      POP   AR3
007B D002      POP   AR2
007D D001      POP   AR1
007F D000      POP   AR0
0081 120000  E   LCALL ?C?FPSUB
0084 120000  E   LCALL ?C?CASTF
0087 8E00    R   MOV   result,R6
0089 8F00    R   MOV   result+01H,R7
                ; SOURCE LINE # 9
008B 0500    R   INC   i+01H
008D E500    R   MOV   A,i+01H
008F 7002      JNZ   ?C0012
0091 0500    R   INC   i
0093      ?C0012:
0093 020000  R   LJMP  ?C0001
0096      ?C0002:
                ; SOURCE LINE # 10
0096 AF00    R   MOV   R7,result+01H
                ; SOURCE LINE # 11
0098 22      RET
                ; FUNCTION _fixedPoint (END)

                ; FUNCTION fixedPointWrapper (BEGIN)
                ; SOURCE LINE # 13
                ; SOURCE LINE # 15
0000 75009C  R   MOV   x,#09CH
```

```

0003      ?C0005:

0003 D3      SETB  C

0004 E500    R   MOV   A,x

0006 9464      SUBB  A,#064H

0008 7480      MOV   A,#080H

000A 9480      SUBB  A,#080H

000C 5012      JNC   ?C0008

                        ; SOURCE LINE # 16

000E 7D06      MOV   R5,#06H

0010 7C00      MOV   R4,#00H

0012 AF00    R   MOV   R7,x

0014 120000   R   LCALL _fixedPoint

0017 EF        MOV   A,R7

0018 2464      ADD   A,#064H

001A F5B0      MOV   P3,A

                        ; SOURCE LINE # 17

001C 0500    R   INC   x

001E 80E3      SJMP  ?C0005

                        ; SOURCE LINE # 18

0020      ?C0008:

0020 22        RET

                        ; FUNCTION fixedPointWrapper (END)


                        ; FUNCTION main (BEGIN)

                        ; SOURCE LINE # 20

0000      ?C0009:

                        ; SOURCE LINE # 21

                        ; SOURCE LINE # 22

0000 120000   R   LCALL fixedPointWrapper

                        ; SOURCE LINE # 23

```

```
0003 80FB      SJMP  ?C0009
```

```
0005 22      RET
```

```
    ; FUNCTION main (END)
```

Вывод

В процессе выполнения лабораторной работы я понял, как работают функции из заголовочного файла `math.h`, а также повторил пройденный материал про чисел с плавающей запятой и фиксированной запятой.