

**ПЛАТФОРМА МУЛЬТИКОР
ПРИКЛАДНАЯ БИБЛИОТЕКА
БИБЛИОТЕКА ЭЛЕМЕНТАРНЫХ ФУНКЦИЙ**

Руководство программиста

Листов 32

2005

Порядок использования настоящей документации

Настоящая документация охраняется действующим законодательством Российской Федерации об авторском праве и смежных правах, в частности, законом Российской Федерации «Об авторском праве и смежных правах». ГУП НПЦ «ЭЛВИС» является единственным правообладателем исключительных авторских прав на настоящую документацию.

Настоящую документацию без предварительного согласия ГУП НПЦ «ЭЛВИС» запрещается:

- воспроизводить, т.е. изготавливать один или более экземпляров настоящей документации, ее части, в любой форме, любым способом;
- сдавать в прокат;
- публично показывать, исполнять или сообщать для всеобщего сведения;
- переводить;
- переделывать или другим образом перерабатывать (дорабатывать).

ГУП НПЦ «ЭЛВИС» оставляет за собой право в любой момент вносить изменения (дополнения) в настоящую документацию без предварительного уведомления о таком изменении (дополнении).

ГУП НПЦ «ЭЛВИС» не несет ответственности за вред, причиненный при использовании настоящей документации.

Передача настоящей документации не означает передачи каких-либо авторских прав ГУП НПЦ «ЭЛВИС» на нее.

Возникновение каких-либо прав на материальный носитель, на котором передается настоящая документация, не влечет передачи каких-либо авторских прав на данную документацию.

Все указанные в настоящей документации товарные знаки принадлежат их владельцам.

ГУП НПЦ «ЭЛВИС» ©, 2004

АННОТАЦИЯ

«Библиотека элементарных функций» (далее – библиотека) входит в состав программы «Платформа мультикор. Прикладная библиотека» РАЯЖ.00013-01.

В документе «Платформа мультикор. Прикладная библиотека. Библиотека элементарных функций. Руководство программиста» РАЯЖ.00013-01 33 03 приведено описание действий программиста для вычисления алгебраических и тригонометрических функций в формате плавающей точки - 24Е8 (стандарт IEEE 754) на DSP-ядре ИМС MultiCore-12.

СОДЕРЖАНИЕ

1. ОБЩИЕ СВЕДЕНИЯ	5
1.1. Необходимое программное обеспечение для функционирования библиотеки	5
1.2. Языки программирования, используемые для написания библиотеки.....	5
2. НАЗНАЧЕНИЕ И УСЛОВИЯ ПРИМЕНЕНИЯ	6
3. СОСТАВ БИБЛИОТЕКИ.....	7
4. Характеристика программы. Обращение, входные и выходные данные	8
4.1. Функция $y = \sin_f(x)$	8
4.2. Функция $y = \cos_f(x)$	9
4.3. Функция $y = \tan_f(x)$	10
4.4. Функция $y = \operatorname{ctan}_f(x)$	11
4.5. Функция $y = \operatorname{asin}_f(x)$	12
4.6. Функция $y = \operatorname{atan}_f(x)$	13
4.7. Функция $z = \operatorname{atan2}_f(y,x)$	14
4.8. Функция $y = \exp_f(x)$	15
4.9. Функция $y = \exp2_f(x)$	16
4.10. Функция $y = \log_f(x)$	17
4.11. Функция $y = \operatorname{recip}_f(x)$	18
4.12. Функция $y = \operatorname{sqrt}_f(x)$	20
4.13. Функция $y = \operatorname{sqrt}_f(x)$	21
4.14. Функция $y = \operatorname{pow}_f(x,y)$	22
4.15. Функция $y = \operatorname{abs}_f(x)$	24
4.16. Функция $y = \operatorname{floor}_f(x)$	25
4.17. Функция $y = \operatorname{ceil}_f(x)$	26
4.18. Функция $[y,n] = \operatorname{frexp}_f(x)$	27
4.19. Функция $[y,n] = \operatorname{modf}_f(x)$	29
4.20. Функция $y = \operatorname{ldexp}_f(x,n)$	30

1. ОБЩИЕ СВЕДЕНИЯ

1.1. Необходимое программное обеспечение для функционирования библиотеки

Библиотека предназначена для использования на DSP-ядре MultiCore-12.

Библиотека представляет собой файл с расширением .a, в котором находятся объектные файлы, содержащие готовый к линковке исполняемый код и данные.

Для подключения и использования библиотеки в проекте необходимо:

- 1) в диалоге *Tools > Settings >* для линкера DSP добавить директиву -L с указанием адреса библиотеки и её именем, заменяя обязательный префикс “lib” на префикс “l”;
- 2) (-lcmf12);
- 3) Вызов функции в программе DSP-ядра осуществляется инструкцией:
BS <имя функции>;
- 4) передача входных параметров и возврат значения функции осуществляется через регистры, указанные в описании конкретной функции. Там же перечислены и другие задействованные в вычислениях регистры.

1.2. Языки программирования, используемые для написания библиотеки

Ассемблер DSP ядра ELcore платформы МУЛЬТИКОР.

2. НАЗНАЧЕНИЕ И УСЛОВИЯ ПРИМЕНЕНИЯ

Библиотека предназначена для ускорения программирования алгебраических и тригонометрических вычислений в формате плавающей точки с одинарной точностью (формат float, 24e8) с использованием ИМС MultiCore-12 (МС-12). Признак формата входит в качестве постфикса в названия всех библиотечных функций.

3. СОСТАВ БИБЛИОТЕКИ

В настоящее время библиотека включает следующие элементарные функции:

`sin_f(x)`, `cos_f(x)`, `tan_f(x)`, `ctan_f(x)`, `exp_f(x)`, `exp2_f(x)`, `log_f(x)`, `asin_f(x)`, `atan_f(x)`,
`atan2_f(x)`, `pow_f(x,y)`, `div_f(x)`, `recip_f(x)`, `sqrt_f(x)` и `sqrt_recip_f(x)`, `abs_f(x)`, `floor_f(x)`,
`ceil_f(x)`, `frexp_f(x)`, `modf_f(x)`, `ldexp_f(x,n)`.

4. ХАРАКТЕРИСТИКА ПРОГРАММЫ. ОБРАЩЕНИЕ, ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ

4.1. Функция $y = \sin_f(x)$

4.1.1. Описание функции

Функция вычисляет синус x .

4.1.2. Входные данные

Регистр R2 (float x): вход (в радианах), $-\infty < x < +\infty$.

4.1.3. Выходные данные

Регистр R0 (float y): выход $y = \sin(x)$, где $-1.0 \leq y \leq +1.0$.

4.1.4. Характеристики программы

4.1.4.1. Затраты памяти

P_Ram: 31 32разр. Слов.

X_Ram: 8 32разр. Слов.

4.1.4.2. Регистры, используемые в процессе вычисления

R24.L, R26.L, R28.L, R30.L, A7, M7.

4.1.4.3. Алгоритм вычисления функции

Аргумент x , где $-\infty < x < +\infty$ преобразуется к интервалу $[-\pi/2, +\pi/2]$.

Синус приведенного входного аргумента вычисляется посредством аппроксимации минимаксным многочленом 9-го порядка.

4.1.4.4. Количество тактов.

Количество тактов – 23.

4.1.4.5. Точность.

Функция вычисляется с 23-битной точностью при $|x| \leq 2\pi$.

4.1.4.6. Синтаксис.

<i>MOVE</i> <i>x, R2.L</i>	; заносим <i>x</i> в регистр <i>R2</i>
<i>BS sin_f</i>	
<i>NOP</i>	; результат в регистре <i>R0</i>

4.2. Функция $y = \cos_f(x)$

4.2.1. Описание функции

Функция вычисляет косинус x .

4.2.2. Входные данные

Регистр *R2* (float x): вход (в радианах), $-\infty < x < +\infty$.

4.2.3. Выходные данные

Регистр *R0* (float y): выход $y = \cos(x)$, где $-1.0 \leq y \leq +1.0$.

4.2.4. Характеристики программы

4.2.4.1. Затраты памяти

P_Ram: 31 32разр. Слов.

X_Ram: 8 32разр. Слов.

4.2.4.2. Регистры, используемые в процессе вычисления

R24.L, *R26.L*, *R28.L*, *R30.L*, *A7*, *M7*.

4.2.4.3. Алгоритм вычисления .

Вычисление $\cos(x)$, где $-\infty < x < +\infty$ приводится к вычислению $\sin(x + \pi/2)$ для смещенного на $\pi/2$ входного аргумента.

4.2.4.4. Количество тактов.

Количество тактов - 23.

4.2.4.5. Точность.

Функция вычисляется с 23-битной точностью при $|x| \leq 2\pi$.

4.2.4.6. Синтаксис.

<i>MOVE</i> <i>x</i> , <i>R2.L</i>	; заносим <i>x</i> в регистр <i>R2</i>
<i>BS</i> <i>cos_f</i>	
<i>NOP</i>	; результат в регистре <i>R0</i>

4.3. Функция $y = \tan_f(x)$

4.3.1. Описание функции

Функция вычисляет тангенс x .

4.3.2. Входные данные

Регистр *R2* (float x): вход (в радианах), $|x| < 105414000.0$

4.3.3. Выходные данные

Регистр *R0* (float y): выход $y = \tan(x)$, где $-\infty \leq y \leq +\infty$.

4.3.4. Характеристики программы

4.3.4.1. Затраты памяти

P_Ram: 82 32 разр. Слов.

X_Ram: 8 32 разр. Слов.

4.3.4.2. Регистры, используемые в процессе вычисления

R18.L, *R20.L*, *R22.L*, *R24.L*, *R26.L*, *R28.L*, *R30.L*, *A7*, *M7*.

4.3.4.3. Алгоритм вычисления функции.

Аргумент x , где $-\infty < x < +\infty$ преобразуется к интервалу $[-\pi/4, +\pi/4]$.

Тангенс приведенного входного аргумента вычисляется посредством дробно-рациональной аппроксимации.

4.3.4.4. Количество тактов.

Количество тактов - 41.

4.3.4.5. Точность.

Функция вычисляется с 22-битной точностью при $|x| < \pi/2$.

4.3.4.6. Синтаксис.

<i>MOVE</i> <i>x, R2.L</i>	; заносим <i>x</i> в регистр <i>R2</i>
<i>BS tan_f</i>	
<i>NOP</i>	; результат в регистре <i>R0</i>

4.4. Функция $y = \text{ctan}_f(x)$

4.4.1. Описание функции

Функция вычисляет котангенс x .

4.4.2. Входные данные

Регистр R2 (float x): вход (в радианах), $|x| < 105414000.0$

4.4.3. Выходные данные

Регистр R0 (float y): выход $y = \text{ctan}(x)$, где $-\infty \leq y \leq +\infty$.

4.4.4. Характеристики программы

4.4.4.1. Затраты памяти

P_Ram: 81 32разр. Слов.

X_Ram: 8 32разр. Слов.

4.4.4.2. Регистры, используемые в процессе вычисления

R18.L, R20.L, R22.L, R24.L, R26.L, R28.L, R30.L, A7, M7.

4.4.4.3. Алгоритм вычисления функции.

Аргумент x , где $-\infty < x < +\infty$ преобразуется к интервалу $[-\pi/4, +\pi/4]$.

Котангенс приведенного входного аргумента вычисляется посредством дробно-рациональной аппроксимации.

4.4.4.4. Количество тактов.

Количество тактов - 41.

4.4.4.5. Точность.

Функция вычисляется с 22-битной точностью при $|x| < \pi/2$.

4.4.4.6. Синтаксис.

<i>MOVE</i> <i>x, R2.L</i>	; заносим <i>x</i> в регистр <i>R2</i>
<i>BS</i> <i>stan_f</i>	
<i>NOP</i>	; результат в регистре <i>R0</i>

4.5. Функция $y = \text{asin}_f(x)$

4.5.1. Описание функции

Функция вычисляет арксинус x .

4.5.2. Входные данные

Регистр R2 (float x): вход, $-1.0 \leq x \leq +1.0$.

4.5.3. Выходные данные

Регистр R0 (float y): выход (в радианах) $y = \text{asin}(x)$, $-\pi/2 \leq y \leq +\pi/2$

4.5.4. Характеристики программы

4.5.4.1. Затраты памяти

P_Ram: 94 32разр. Слов.

X_Ram: 16 32разр. Слов.

4.5.4.2. Регистры, используемые в процессе вычисления

R18.L, R20.L, R22.L, R24.L, R26.L, R28.L, R30.L, A7, M7.

4.5.4.3. Алгоритм вычисления функции.

Арксинус входного аргумента вычисляется посредством аппроксимации многочленом 19-ого порядка.

4.5.4.4. Количество тактов.

Количество тактов - 41.

4.5.4.5. Точность.

Функция вычисляется с 23-битной точностью.

4.5.4.6. Синтаксис.

<i>MOVE</i> <i>x, R2.L</i>	; заносим <i>x</i> в регистр <i>R2</i>
<i>BS</i> <i>asin_f</i>	
<i>NOP</i>	; результат в регистре <i>R0</i>

4.6. Функция $y = \text{atan}_f(x)$

4.6.1. Описание функции

Функция вычисляет арктангенс x .

4.6.2. Входные данные

Регистр R2 (float x): вход, $-\infty < x < +\infty$.

4.6.3. Выходные данные

Регистр R0 (float y): выход - $\text{atan}(x)$, $-\pi/2 \leq y \leq +\pi/2$

4.6.4. Характеристики программы

4.6.4.1. Затраты памяти

P_Ram: 70 32разр. Слов.

X_Ram: 9 32разр. Слов.

4.6.4.2. Регистры, используемые в процессе вычисления

R18.L, R20.L, R22.L, R24.L, R26.L, R28.L, R30.L, A7, M7.

4.6.4.3. Алгоритм вычисления функции.

Арктангенс входного аргумента вычисляется посредством полиномиальной аппроксимации 17-ого порядка.

4.6.4.4. Количество тактов.

Количество тактов - 39.

4.6.4.5. Точность.

Функция вычисляется с 23-битной точностью.

4.6.4.6. Синтаксис.

<i>MOVE</i> <i>x, R2.L</i>	; заносим <i>x</i> в регистр <i>R2</i>
<i>BS atan_f</i>	
<i>NOP</i>	; результат в регистре <i>R0</i>

4.7. Функция $z = \text{atan2}_f(y,x)$

4.7.1. Описание функции

Функция вычисляет арктангенс y/x , используя знаки обоих аргументов для определения квадранта возвращаемого значения.

4.7.2. Входные данные

Регистр R2 (float y): вход y .

Регистр R4 (float x): вход x .

4.7.3. Выходные данные

Регистр R0 (float z): выход $z = \text{atan}(y/x)$, $-\pi \leq z \leq +\pi$

4.7.4. Характеристики программы

4.7.4.1. Затраты памяти

P_Ram: 103 32разр. Слов.

X_Ram: 9 32разр. Слов.

4.7.4.2. Регистры, используемые в процессе вычисления

R12.L, R14.L, R16.L, R18.L, R20.L, R22.L, R24.L, R26.L, R28.L, R30.L, A7, M7.

4.7.4.3. Алгоритм вычисления функции.

Арктангенс2 входных аргументов вычисляется посредством полиномиальной аппроксимации 17-ого порядка. Используются знаки обоих аргументов для определения квадранта выходного значения.

4.7.4.4. Количество тактов.

Количество тактов - 60.

4.7.4.5. Точность.

Функция вычисляется с 22-битной точностью.

4.7.4.6. Синтаксис.

<i>MOVE</i> <i>y</i> , <i>R2.L</i>	; заносим <i>x</i> в регистр <i>R2</i>
<i>MOVE</i> <i>x</i> , <i>R4.L</i>	; заносим <i>x</i> в регистр <i>R4</i>
<i>BS atan2_f</i>	
<i>NOP</i>	; результат в регистре <i>R0</i>

4.8. Функция $y = \exp_f(x)$

4.8.1. Описание функции

Функция вычисляет значение экспоненциальной функции от x .

4.8.2. Входные данные

Регистр R2 (float x): вход x , $|x| \leq 88.72283906$.

4.8.3. Выходные данные

Регистр R0 (float y): выход - $\exp(x)$.

4.8.4. Характеристики программы

4.8.4.1. Затраты памяти

P_Ram: 61 32разр. Слов.

X_Ram: 14 32разр. Слов.

4.8.4.2. Регистры, используемые в процессе вычисления

R18.L, R20.L, R22.L, R24.L, R26.L, R28.L, R30.L, A7, M7

4.8.4.3. Алгоритм вычисления функции

Аргумент x представляется в виде (4.0)

$$x = n + y, \text{ где } n = E(x), y = \{x\}, 0 \leq y < 1. \quad (4.0)$$

Экспонента входного аргумента представляется в виде (4.1):

$$\text{Exp}(x) = \exp(n) * \exp(y), \quad (4.1)$$

Функция $\exp(y)$ вычисляется с помощью полиномиального многочлена 10-ого порядка.

4.8.4.4. Количество тактов.

Количество тактов - 40.

4.8.4.5. Точность.

Функция вычисляется с 22-битной точностью.

4.8.4.6. Синтаксис.

<i>MOVE</i>	<i>x, R2.L</i>	; заносим <i>x</i> в регистр <i>R2</i>
<i>BS</i>	<i>exp_f</i>	
<i>NOP</i>	; результат в регистре <i>R0</i>

4.9. Функция $y = \exp2_f(x)$

4.9.1. Описание функции

Функция вычисляет значение 2^x .

4.9.2. Входные данные

Регистр *R2* (float *x*): вход *x*, $|x| < 126.233$.

4.9.3. Выходные данные

Регистр *R0* (float *y*): выход - 2^x .

4.9.4. Характеристики программы

4.9.4.1. Затраты памяти

P_Ram: 72 32разр. Слов.

X_Ram: 7 32разр. Слов.

4.9.4.2. Регистры, используемые в процессе вычисления

R18.L, R20.L, R22.L, R24.L, R26.L, R28.L, R30.L, A7, M7

4.9.4.3. Алгоритм вычисления функции

Аргумент x представляется в виде (4.2):

$$x = n + y + \alpha * 0.5, \text{ где } n = [x], \alpha = \{0,1\}, 0 \leq y < 0.5 \quad (4.2)$$

Функция 2^x представляется в виде (4.3):

$$2^x = 2^n * 2^{0.5 * \alpha} * 2^y \quad (4.3)$$

Функция 2^y вычисляется с помощью полиномиального многочлена 6-ого порядка.

4.9.4.4. Количество тактов.

Количество тактов - 45.

4.9.4.5. Точность.

Функция вычисляется с 22-битной точностью.

4.9.4.6. Синтаксис.

<pre> MOVE x, R2.L ; заносим x в регистр R2 BS exp2_f NOP ; результат в регистре R0</pre>

4.10. Функция $y = \log_f(x)$

4.10.4. Описание функции

Функция вычисляет значение натурального логарифма от x .

4.10.5. Входные данные

Регистр R2 (float x): вход, $0 < x$.

4.10.6. Выходные данные

Регистр R0 (float y): выход - $\log(x)$.

4.10.7. Характеристики программы

4.10.7.2. Затраты памяти

P_Ram: 83 32разр. Слов.

X_Ram: 15 32разр. Слов.

4.10.7.3. Регистры, используемые в процессе вычисления

R14.L, R16, R18.L, R20.L, R22.L, R24.L, R26.L, R28.L, R30.L, A7, M7

4.10.7.4. Алгоритм вычисления функции.

Натуральный логарифм входного аргумента вычисляется посредством полиномиальной аппроксимации 10-ого порядка.

4.10.7.5. Количество тактов.

Количество тактов - 42.

4.10.7.6. Точность.

Функция вычисляется с 23-битной точностью.

4.10.7.7. Синтаксис.

<i>MOVE</i> <i>x</i> , <i>R2.L</i> ; заносим <i>x</i> в регистр <i>R2</i>
<i>BS</i> <i>log_f</i>
<i>NOP</i> ; результат в регистре <i>R0</i>

4.11. Функция $y = \text{recip_f}(x)$

4.11.4. Описание функции

Функция вычисляет значение $1/x$.

4.11.5. Входные данные

Регистр R2 (float x): вход.

4.11.6. Выходные данные

Регистр R0 (float y): выход – $1/x$.

4.11.7. Характеристики программы

4.11.7.2. Затраты памяти

P_Ram: 19 32разр. Слов.

X_Ram: 0 32разр. Слов.

4.11.7.3. Регистры, используемые в процессе вычисления

R28.L, R30.L.

4.11.7.4. Алгоритм вычисления функции

Вычисление функции $1/x$ основано на итерационном процессе по формуле Ньютона – Рафсона (4.4):

$$y_{n+1} = y_n * (2 - x * y_n). \quad (4.4)$$

4.11.7.5. Количество тактов.

Количество тактов - 12.

4.11.7.6. Точность.

Функция вычисляется с 23-битной точностью.

4.11.7.7. Синтаксис.

<i>MOVE</i> <i>x, R2.L</i> ; заносим <i>x</i> в регистр <i>R2</i>
<i>BS recip_f</i>
<i>NOP</i> ; результат в регистре <i>R0</i>

Функция $z = \text{div}_f(y,x)$

4.11.8. Описание функции

Функция вычисляет значение y/x .

4.11.9. Входные данные

Регистр R2 (float y) рег. R4 (float x): вход $-\infty < y < +\infty, -\infty < x < +\infty$.

4.11.10. Выходные данные

Регистр R0 (float z): выход $-y/x$.

4.11.11. Характеристики программы

4.11.11.2. Затраты памяти

P_Ram: 18 32 разр. Слов.

X_Ram: 0 32 разр. Слов.

4.11.11.3. Регистры, используемые в процессе вычисления

R28.L, R30.L.

4.11.11.4. Алгоритм вычисления функции.

Представляем z в виде (4.5):

$$z = y * 1/x. \quad (4.5)$$

Вычисление функции $1/x$ основано на итерационном процессе по формуле Ньютона – Рафсона (4.4).

4.11.11.5. Количество тактов.

Количество тактов - 12.

4.11.11.6. Точность.

Функция вычисляется с 23-битной точностью.

4.11.11.7. Синтаксис.

<i>MOVE</i>	<i>y, R2.L</i>	<i>; заносим y в регистр R2</i>
<i>MOVE</i>	<i>x, R4.L</i>	<i>; заносим x в регистр R4</i>
<i>BS</i>	<i>div_f</i>	
<i>NOP</i>	<i>.....</i>	<i>; результат в регистре R0</i>

4.12. Функция $y = \text{sqrt_recip_f}(x)$

4.12.4. Описание функции

Функция вычисляет значение $1/\sqrt{x}$.

4.12.5. Входные данные

Регистр R2 (float x): вход $x > 0$

4.12.6. Выходные данные

Регистр R0 (float y): выход $-1/\sqrt{x}$.

4.12.7. Характеристики программы

4.12.7.2. Затраты памяти

P_Ram: 32 32разр. Слов.

X_Ram: 0 32разр. Слов.

4.12.7.3. Регистры, используемые в процессе вычисления

R26.L, R28.L, R30.L.

4.12.7.4. Алгоритм вычисления функции

Вычисление функции $1/\sqrt{x}$ основано на итерационном процессе по формуле Ньютона – Рафсона (4.6):

$$y' = y_0 * (3 - x * y_0^2) / 2 \quad (4.6)$$

4.12.7.5. Количество тактов.

Количество тактов - 13.

4.12.7.6. Точность.

Функция вычисляется с 23-битной точностью.

4.12.7.7. Синтаксис.

<pre>MOVE x, R2.L ; заносим x в регистр R2 BS sqrt_recip_f NOP ; результат в регистре R0</pre>
--

4.13. Функция $y = \text{sqrt_f}(x)$

4.13.4. Описание функции

Функция вычисляет значение квадратного корня из числа x .

4.13.5. Входные данные

Регистр R2 (float x): вход $x > 0$

4.13.6. Выходные данные

Регистр R0 (float y): выход - \sqrt{x} .

4.13.7. Характеристики программы

4.13.7.2. Затраты памяти

P_Ram: 22 32разр. Слов.

X_Ram: 0 32разр. Слов.

4.13.7.3. Регистры, используемые в процессе вычисления

R26.L, R28.L, R30.L.

4.13.7.4. Алгоритм вычисления функции

Функция $y = \sqrt{x}$ представляется в следующем виде (4.7):

$$y = x * 1/\sqrt{x} \quad (4.7)$$

Вычисление функции $1/\sqrt{x}$ основано на итерационном процессе по формуле Ньютона – Рафсона, формула (4.6).

4.13.7.5. Количество тактов.

Количество тактов - 16.

4.13.7.6. Точность.

Функция вычисляется с 23-битной точностью.

4.13.7.7. Синтаксис.

<pre>MOVE x, R2.L ; заносим x в регистр R2 BS sqrt_f NOP ; результат в регистре R0</pre>
--

4.14. Функция $y = \text{pow}_f(x,y)$

4.14.4. Описание функции

Функция возводит x в степень y .

4.14.5. Входные данные

Регистр R2 (float x): вход x

Регистр R4 (float y): вход y

4.14.6. Выходные данные

В регистр R0 (float z): выход - x^y .

В таблице 1 приведены значения функции в граничных и особых точках.

Таблица 1

Значение x, y	Значение функции
$x = \pm(\text{любое число}), y = 0$	1.0
$x = \text{NaN}, y = \pm(\text{любой число})$	NaN
$x = \pm(\text{любое число}), y = \text{NaN}$	NaN
$x = +\text{INF}, y = +(\text{любое число})$	+INF
$x = +\text{INF}, y = -(\text{любое число})$	0
$x = -\text{INF}, y = \pm(\text{любое число})$	0
$x > \pm 1.0, y = +\text{INF}$	+INF
$x > \pm 1.0, y = -\text{INF}$	0
$x < \pm 1.0, y = +\text{INF}$	0
$x < \pm 1.0, y = -\text{INF}$	+INF
$x = \pm(\text{любое число}), y = 1.0$	x
$x = 0, y < 0$	+INF

4.14.7. Характеристики программы

4.14.7.2. Затраты памяти

P_Ram: 160 32-битных слов;

X_Ram: 9 32-битных слов.

4.14.7.3. Регистры, используемые в процессе вычисления

R18.L, R20.L, R22.L, R24.L, R26.L, R28.L, R30.L, A7, M7

4.14.7.4. Алгоритм вычисления функции

Функция $Z = \text{pow}(x,y)$ приводится к виду $Z = 2^{(y \cdot \log_2(x))}$, где $x > 0$. Пусть α – мантисса аргумента x , p – порядок.

Величина $\log_2(\alpha)$ аппроксимируется полиномом 9-й степени:

$$\log_2(\alpha) = (((((C9 \cdot \beta^2 + C7) \cdot \beta^2 + C5) \cdot \beta^2 + C3) \cdot \beta^2 + C1) \cdot \beta,$$

где $\beta = (\alpha - 1) / (\alpha + 1)$, $\beta^2 = \beta \cdot \beta$.

Тогда $\log_2(x) = p + \log_2(\alpha)$,

$D = y \cdot \log_2(x)$. Пусть n, γ - целая и дробная части D , $n = [D]$, $\gamma = D - n$.

Величина 2^γ аппроксимируется полиномом 6-й степени:

$$2^\gamma = ((((((T6 \cdot \gamma + T5) \cdot \gamma + T4) \cdot \gamma + T3) \cdot \gamma + T2) \cdot \gamma + T1) \cdot \gamma + T0.$$

Теперь $Z = 2^D = 2^n \cdot 2^\gamma$.

4.14.7.5. Количество тактов.

Количество тактов - 64.

4.14.7.6. Точность.

Максимальная погрешность при $0 \leq x \leq 5.0$, $0 \leq y \leq 5.0$ составляет 6 бит.

Максимальная погрешность при $5.0 \leq x \leq 10.0$, $5.0 \leq y \leq 10.0$ составляет 10 бит.

4.14.7.7. Синтаксис.

<i>MOVE</i>	<i>x, R2.L</i>	; заносим <i>x</i> в регистр <i>R2</i>
<i>MOVE</i>	<i>y, R4.L</i>	; заносим <i>x</i> в регистр <i>R4</i>
<i>BS</i>	<i>pow_f</i>	
<i>NOP</i>	; результат в регистре <i>R0</i>

4.15. Функция $y = \text{abs}_f(x)$

4.15.4. Описание функции

Функция вычисляет абсолютное значение числа x .

4.15.5. Входные данные

Регистр *R2* (float x): вход, $-\infty < x < +\infty$.

4.15.6. Выходные данные

Регистр R0 (float y): выход $\text{abs}(x)$.

4.15.7. Характеристики программы

4.15.7.2. Затраты памяти

P_Ram: 3 32разр. Слов.

X_Ram: 0 32разр. Слов.

4.15.7.3. Регистры, используемые в процессе вычисления

4.15.7.4. Алгоритм вычисления функции

Функция $\text{abs}(x)$ вычисляется с помощью логического умножения входного значения на число $0x7FFFFFFF$.

4.15.7.5. Количество тактов.

Количество тактов - 3.

4.15.7.6. Точность.

Функция вычисляется с 24-битной точностью.

4.15.7.7. Синтаксис.

<pre> MOVE x, R2.L ; заносим x в регистр R2 BS abs_f NOP ; результат в регистре R0</pre>
--

4.16. Функция $y = \text{floor}_f(x)$

4.16.4. Описание функции

Функция вычисляет наибольшее целое значение, не большее чем x .

4.16.5. Входные данные

Регистр R2 (float x): вход, $-\infty < x < +\infty$.

4.16.6. Выходные данные

Регистр R0 (float y): выход floor(x).

4.16.7. Характеристики программы

4.16.7.2. Затраты памяти

P_Ram: 3 32разр. Слов.

X_Ram: 0 32разр. Слов.

4.16.7.3. Регистры, используемые в процессе вычисления

4.16.7.4. Алгоритм вычисления функции

Функция floor(x) вычисляется с помощью инструкции ассемблера DSP ядра «FLOOR».

4.16.7.5. Количество тактов.

Количество тактов - 3.

4.16.7.6. Точность.

Функция вычисляется с 24-битной точностью.

4.16.7.7. Синтаксис.

<pre> MOVE x, R2.L ; заносим x в регистр R2 BS floor_f NOP ; результат в регистре R0</pre>
--

4.17. Функция $y = \text{ceil}_f(x)$

4.17.4. Описание функции

Функция вычисляет наименьшее целое значение, не менее чем x .

4.17.5. Входные данные

Регистр R2 (float x): вход, $-\infty < x < +\infty$.

4.17.6. Выходные данные

Регистр R0 (float y): выход $\text{ceil}(x)$.

4.17.7. Характеристики программы

4.17.7.2. Затраты памяти

P_Ram: 5 32разр. Слов.

X_Ram: 0 32разр. Слов.

4.17.7.3. Регистры, используемые в процессе вычисления

4.17.7.4. Алгоритм вычисления функции

При вычислении функция $\text{ceil}(x)$ используется инструкция ассемблера DSP ядра «FLOOR». Если результат инструкции «FLOOR» меньше входного значения, то он увеличивается на единицу.

4.17.7.5. Количество тактов.

Количество тактов - 7.

4.17.7.6. Точность.

Функция вычисляется с 24-битной точностью.

4.17.7.7. Синтаксис.

<pre> MOVE x, R2.L ; заносим x в регистр R2 BS ceil_f NOP ; результат в регистре R0</pre>

4.18. Функция $[y,n] = \text{frexp}_f(x)$

4.18.4. Описание функции

Функция возвращает мантиссу и экспоненту числа x .

4.18.5. Входные данные

Регистр R2 (float x): вход, $-\infty < x < +\infty$.

4.18.6. Выходные данные

Регистр R0 (int n): выход $\exp(x)$.

Регистр R2 (float y): выход $\frac{1}{2} \leq \text{mantissa}(x) < 1$.

В таблице 2 приведены значения функции в граничных и особых точках

Таблица 2

Значение x	Значение функции
$x = -\text{INF}$	exponent = 255, mantissa = -HUGE_VAL
$x = +\text{INF}$	exponent = 255, mantissa = HUGE_VAL
$x = \pm\text{NaN}$	exponent = 255, mantissa = QNaN
$x = 0$	exponent = 0, mantissa = ± 0

HUGE_VAL = $3.402823466 \cdot 10^{38}$

4.18.7. Характеристики программы

4.18.7.2. Затраты памяти

P_Ram: 30 32разр. Слов.

X_Ram: 0 32разр. Слов.

4.18.7.3. Регистры, используемые в процессе вычисления

R30.L

4.18.7.4. Алгоритм вычисления функции

Экспонента и мантисса входного числа вычисляется по следующим формулам:

$$\text{exp} = (x \gg 23) \& 0\text{xff} - 126;$$

$$\text{mantissa} = (x \& 0\text{x807ffff}) | 0\text{x3f000000}.$$

4.18.7.5. Количество тактов.

Количество тактов - 14.

4.18.7.6. Точность.

Функция вычисляется с 24-битной точностью.

4.18.7.7. Синтаксис.

<i>MOVE</i> <i>x, R2.L</i>	; заносим <i>x</i> в регистр <i>R2</i>
<i>BS</i> <i>fexp_f</i>	
<i>NOP</i>	; результат в регистре <i>R0, R2</i>

4.19. Функция $[y,n] = \text{modf}_f(x)$

4.19.4. Описание функции

Функция преобразует число x в целую и дробную часть, имеющие тот же знак, что и число x .

4.19.5. Входные данные

Регистр *R2* (float x): вход, $-\infty < x < +\infty$.

4.19.6. Выходные данные

Регистр *R0* (float y): выход *fractional_part* (x).

Регистр *R2* (float n): выход *int_part*(x).

В таблице 3 приведены значения функции в граничных и особых точках.

Таблица 3

Значение x	Значение функции
$x = \pm\text{INF}$	<i>fractional_part</i> = 0.0, <i>int_part</i> = $\pm\text{INF}$
$x = \pm\text{NAN}$	<i>fractional_part</i> = QNAN, <i>int_part</i> = $\pm\text{NAN}$

4.19.7. Характеристики программы

4.19.7.2. Затраты памяти

P_Ram: 16 32разр. Слов.

X_Ram: 0 32разр. Слов.

4.19.7.3. Регистры, используемые в процессе вычисления

R30.L

4.19.7.4. Алгоритм вычисления функции

Целая часть входного числа вычисляется с использованием функции ассемблера DSP ядра «FLOOR».

4.19.7.5. Количество тактов.

Количество тактов - 11.

4.19.7.6. Точность.

Функция вычисляется с 24-битной точностью.

4.19.7.7. Синтаксис.

<pre> MOVE x, R2.L ; заносим x в регистр R2 BS modf_f NOP ; результат в регистре R0, R2</pre>
--

4.20. Функция $y = \text{ldexp}_f(x, n)$

4.20.4. Описание функции

Функция вычисляет значение $x * 2^n$.

4.20.5. Входные данные

Регистр R2 (float x): вход, $-\infty < x < +\infty$.

Регистр R4 (int n): вход, $-\infty < n < +\infty$.

4.20.6. Выходные данные

Регистр R0 (float y): выход $x * 2^n$.

В таблице 4 приведены значения функции в граничных и особых точках.

Таблица 4

Значение x	Значение функции
$x = 0, n = \pm$ (любое число)	0.0
$x = \pm INF, n = \pm$ (любое число)	$\pm INF$
$x = \pm NAN, n = \pm$ (любое число)	$\pm NAN$
$fabs (x * 2^n) > MAX_VAL$	$\pm MAX_VAL$
$fabs (x * 2^n) < MIN_VAL$	0.0

$MIN_VAL = 3.402823466 * 10^{-38}$

$MAX_VAL = 3.402823466 * 10^{38}$

4.20.7. Характеристики программы

4.20.7.2. Затраты памяти

P_Ram: 26 32разр. Слов.

X_Ram: 0 32разр. Слов.

4.20.7.3. Регистры, используемые в процессе вычисления

R30.L

4.20.7.4. Алгоритм вычисления функции

Выходное значение вычисляется по формуле:

$y = ((x \& 0x807ffff) | ((x \gg 23) \& 0xff + n))$, где n – экспонента, x – мантисса.

4.20.7.5. Количество тактов.

Количество тактов - 17.

4.20.7.6. Точность.

Функция вычисляется с 24-битной точностью.

4.20.7.7. Синтаксис.

<i>MOVE</i> x, R2.L	; заносим x в регистр R2
<i>MOVE</i> n, R4.L	; заносим n в регистр R4
<i>BS</i> l _{dexp_f}	
<i>NOP</i>	; результат в регистре R0

