В этом случае при $|x| \le 1$ для получения относительной ошибки вычисления $Er = 1 \cdot 10^{-3}\%$ необходимо использовать 5 членов ряда $(Er = 4.54 \cdot 10^{-4}\%)$. Однако число выполняемых при этом операций увеличивается.

Известны методы повышения точности вычисления функции, основанные на масштабировании ее аргументы. Проведем в исходном разложении замену:

$$x = \frac{x}{2} \cdot 2 = 2 \cdot \left(\frac{4 \cdot x - 1}{8} + \frac{1}{8} \right) = 2 \cdot \left(\frac{y}{8} + \frac{1}{8} \right).$$

В этом случае расчетное выражение для функции примет вид:

$$exp(x) = exp(0.25) \cdot \left[I_0(0.125) + 2 \cdot \sum_{k=1}^{\infty} I_k(0.125) \cdot T_k(4 \cdot x - 1) \right]^2$$

Теперь для получения относительной ошибки вычисления в том же диапазоне изменения аргумента на уровне $Er = 1 \cdot 10^{-3}\%$ достаточно использование только двух членов ряда $(Er = 2.8 \cdot 10^{-4}\%)$ и вычисление функции потребует выполнения 6 операций сложения и 5 операций умножения.

Последний алгоритм был положен в основу разработки вычислителя функции $exp(x)$ с использованием ПЛИС фирмы Altera.

В таблице 2 приведены результаты исследования модели разработанного устройства

Таблица 2.

Значение аргумента	0.109375	0.3046875	0.5	0.625	0.90625
Относительная ошибка	1.452664e-4	1.472459e-4	4.115197e-4	6.841889e-3	0.075145

При вычислении функций Бесселя учитывалось, как минимум, 7 правильных десятичных знаков после запятой, что соответствует 24 разрядам дробной части двоичного числа $\left(2^{-24} = 5.9 \cdot 10^{-8}\right)$. Разрядность аргумента функции – 8 бит дробной части. $\left(2^{-8} = 3.9 \cdot 10^{-3}\right)$

На основании изложенного можно сделать вывод, что применение при вычислении функции $exp(x)$ разложения по многочленам Чебышева позволяет, как уменьшить необходимые для реализации аппаратные ресурсы ПЛИС, так и уменьшить суммарное время ее вычисления.

ОСОБЕННОСТИ РЕАЛИЗАЦИИ ОПЕРАЦИИ УМНОЖЕНИЯ НА ПЛИС

Мо Чжо Чо

Кафедра электроники и информатики, Московский авиационный технический институт

Система проектирования Quartus II, используемая для разработки устройств на основе ПЛИС фирмы Altera, включает библиотеку стандартных подпрограмм, предназначенных для выполнения основных математических операций. В частности, для реализации операции умножения используют подпрограмму lpm_mult. Эта подпрограмма позволяет синтезировать схему устройства, реализующего на аппаратном уровне умножитель двух двоичных чисел заданной разрядности. Очевидно, что чем больше разрядность сомножителей, тем сложнее устройство и при его реализации расходуются большие аппаратные ресурсы ПЛИС и возрастает время вычисления. Поэтому проблема сокращения аппаратных затрат и времени вычисления является весьма важной.

Согласно выдвинутой А. Н. Колмогоровым гипотезе $\mathbf{n^2}$ [1] сложность операции умножения, то есть количество простых операций выполняемых для получения произведения, пропорциональна квадрату разрядности сомножителей. Следовательно и аппаратные затраты на реализации умножителя должны расти пропорционально квадрату разрядности сомножителей. Желание упростить реализацию умножителей привело к разработке методов быстрого умножения, для которых сложность реализации пропорциональна $\mathbf{n^{\log_2 3}}$. [2].

Ниже рассмотрены результаты реализации метода быстрого умножения, базирующиеся на использовании выражения:

$$(a+b\cdot x)\cdot(c+d\cdot x)=a\cdot c+\left[(a+b)\cdot(c+d)-a\cdot c-b\cdot d\right]\cdot x+b\cdot d\cdot x^2.$$

Суть метода базируется на том, что для позиционной системы счисления умножение на число, равное основанию этой системы равносильно сдвигу записи числа на один разряд, то есть его выполнение не требует применения дополнительных средств. Практически при реализации умножителя двоичные $2\cdot n$ разрядные числа A и B записывают в виде $A=A_1+2^n\cdot A_2$ и $B=B_1+2^n\cdot B_2$. Тогда алгоритм умножения приобретает вид:

$$A\cdot B=A_1\cdot B_1+2^n\cdot\left[(A_1+A_2)\cdot(B_1+B_2)-(A_1\cdot B_1+A_2\cdot B_2)\right]+2^{2\cdot n}\cdot A_2\cdot B_2$$

В этом случае перемножаются числа, разрядность которых в два раза меньше исходных со сдвигом промежуточных результатов на требуемое число разрядов, что предполагает уменьшение аппаратных затрат на реализацию.

Исследование критического пути, параллельной канонической формы графа данного алгоритма, показало, что время получения результата определяется выражением:

$$t_{K.\Sigma}=max\{t_{dataa}(n),t_{datab}(n)\}+t_{add}(2\cdot n+2)+t_{add}(4\cdot n),$$

где: $t_{datab}(n)=t_{mult}(n)+t_{add}(2\cdot n)$, $t_{dataa}(n)=t_{add}(n)+t_{mult}(n+1)$,

$t_{add}(n)$ - время суммирования n разрядных аргументов,

$t_{mult}(n)$ - время умножения n разрядных аргументов.

Для исследования зависимости аппаратных затрат, требуемых на реализацию данного алгоритма, от разрядности аргументов использовалась написанная на языке AHDL параметризированная программа. Исследования проводились для ПЛИС семейств Cyclone II и Stratix II. Эти ПЛИС предполагают выполнение операции умножения либо с использованием только логических блоков (ЛБ), либо с использованием специализированных встроенных умножителей, разрядности 9 x 9 бит.

Исследования показали, что в первом случае при увеличении разрядности аргументов использование алгоритма быстрого умножения позволяет уменьшить затраты на реализацию умножителя для ПЛИС Cyclone II при $n>15$ (параметр maximize_speed = 0) и для ПЛИС Stratix II при $n>35$. При использовании встроенных умножителей применение метода быстрого умножения также позволяет снизить затраты на реализацию. В этом случае, например для ПЛИС семейства Cyclone II существует несколько областей ($20\le n\le 34$, $56\le n\le 70$, $n>90$) где его применение предпочтительнее стандартной мегафункции.

Полученные аналитические зависимости времени выполнения операции от разрядности аргументов для алгоритма быстрого умножения и стандартной мегафункции позволили найти минимальное число разрядов аргументов $(n_{гр})$ для которого применение метода быстрого умножения позволяет уменьшить время вычисления относительно стандартной мегафункции **lpm_mult** :

- для ПЛИС Cyclone II, случай только ЛБ и maximize_speed = 0 $n_{гр}\ge 24$;
- для ПЛИС Cyclone II, случай только ЛБ и maximize_speed = 10 $n_{гр}\ge 42$;
- для ПЛИС Cyclone II, случай встроенных умножителей $n_{гр}\ge 90$;
- для ПЛИС Stratix II, случай только ЛБ $n_{гр}\ge 170$;
- для ПЛИС Stratix II, случай встроенных умножителей $n_{гр}\ge 107$

Подытоживая сказанное можно сделать вывод, что при увеличении разрядности сомножителей применение метода быстрого умножения позволяет получить выигрыш как по требуемым аппаратным ресурсам ПЛИС, так и по скорости вычисления.

СПИСОК ЛИТЕРАТУРЫ:

1. http://mech.math.msu.su/probab/Kolmogorov/kolmogorov.html
2. http://www.ccas.ru/personal/karatsuba/alg.htm.

КЛАССИЧЕСКАЯ ФИЗИКА В КАТАСТРОФЕ

Ростовцев А.К.
Камышин, Россия

Рассмотрим, о какой катастрофе в классической физике идёт речь?

Задача. Шарик массой m подвешен на нерастяжимой нити ℓ. Нить равномерно вращается в пространстве, образуя с вертикалью угол α (конический маятник). Определить центростремительную силу **F** и силу, которая отклоняет шарик от положения равновесия?

Физики предлагают общепринятый вариант (см. Рис. 1а).

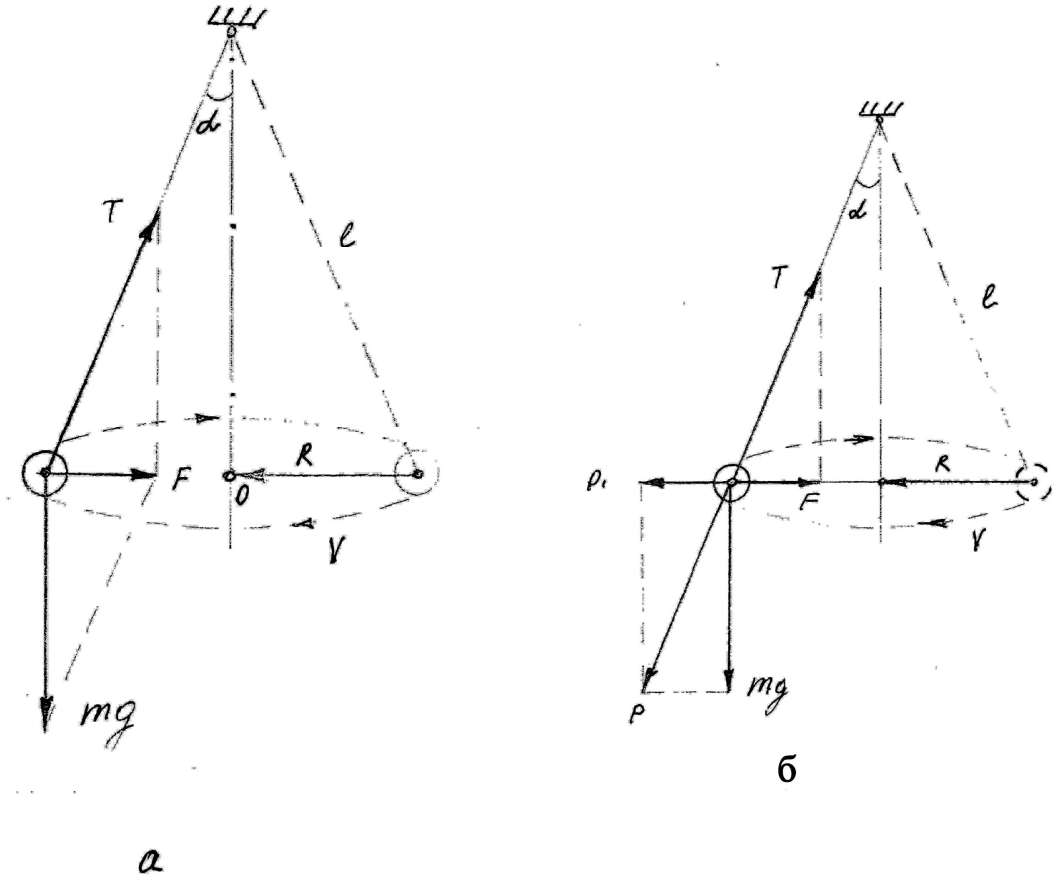

Рис. 1.

Решение

Чтобы найти центростремительную силу нужно сложить силы $m\vec{g}$ и $\vec{T}$ по правилу параллелограмма и найти диагональ зная, что равнодействующая этих сил, согласно второму закону Ньютона, направлена по радиусу. Но откуда такая уверенность? Ведь для того, чтобы соблюдался второй закон Ньютона, при движении тела по окружности, необходимо точно знать величи-