УДК 681.325.5(07)

МОДЕЛИРОВАНИЕ И ПРОГРАМИРОВАНИЕ ЦЕЛОЧИСЛЕННЫХ АЛГОРИТМОВ АРИФМЕТИЧЕСКИХ ОПЕРАЦИЙ

М.И.Ледовской
Технологический институт Южного федерального университета в г.Таганроге

Рассматриваются целочисленные алгоритмы арифметических операций, аналитическая оценка погрешности алгоритмов, приводятся структура модели для экспериментального анализа погрешности алгоритмов и пример моделирования и программирования целочисленного алгоритма.

Современные микропроцессоры и микроконтроллеры позволяют производить математические вычисления над вещественными переменными в 2-х режимах: с фиксированной и плавающей точкой. Режим вычислений с фиксированной точкой, реализуемый в целочисленном формате данных, обеспечивает минимальное время вычислений. Он находит широкое применение в микроконтроллерах, предназначенных для решения одной или нескольких вполне определенных задач. В этих случаях использование микроконтроллеров с аппаратной реализацией режима плавающей точки зачастую неоправданно с точки зрения стоимости, габаритов и энергопотребления.

Однако реализация режима фиксированной точки приводит к необходимости получения специальных целочисленных алгоритмов математических операций, у которых разрядность представления данных должна быть согласована с допустимой инструментальной погрешностью [1]. Кроме того, возникает необходимость программирования этих алгоритмов на языке низкого уровня с целью получения программы с минимальным объемом и временем выполнения.

Для получения целочисленных алгоритмов арифметических операций, заданных относительно вещественных значений аргументов, достаточно воспользоваться табл.1 [2]. Здесь приведены исходные операции над вещественными переменными x, y, z, для которых интервалы изменения $|x|_{min} \le |x| \le |x|_{max}$, $|y|_{min} \le |y| \le |y|_{max}$, и $|z|_{min} \le |z| \le |z|_{max}$ считаются известными.

Таблица 1

Арифметическая операция	Масштабированная операция	Целочисленный алгоритм операции
$z = x \pm y$	$Z = \frac{M_z}{M_x} X \pm \frac{M_z}{M_y} Y$	$2^k = \max\{2^s, 2^r\}$, $Z = \begin{cases} \left[2^s X \pm 2^r Y\right]_ц^{1/2}, \text{ при } 2^k \le 1; \\ \left[2^k (2^{s-k} X \pm 2^{r-k} Y)\right]_ц^{1/2}, \text{ при } 2^k > 1 \end{cases}$
$z = x \cdot y$	$Z = \frac{M_z}{M_x M_y} X \cdot Y$	$Z = \left[2^{-m} (X \cdot Y)\right]_ц^{1/2}$
$z = x / y$	$Z = \frac{M_z M_y}{M_x} \frac{X}{Y}$	$Z = (2^d X) \text{ div } Y$, $W = (2^d X) \text{ mod } Y$, $Z = \begin{cases} Z, \text{ если } \lvert W \rvert < 2^{-1} \lvert Y \rvert, \\ Z + \text{Sign}(Z), \text{ если } \lvert W \rvert \ge 2^{-1} \lvert Y \rvert \end{cases}$

Затем каждая из операций представлена в масштабированном виде, где M_x, M_y, M_z – масштабы переменных x, y, z, выбираемые из следующих условий:

$$M_x \le \frac{2^{n-1} - 1}{|x|_{max}}, \quad M_y \le \frac{2^{n-1} - 1}{|y|_{max}}, \quad M_z \le \frac{2^{n-1} - 1}{|z|_{max}}, \tag{1}$$

n – разрядность целочисленного формата данных. Для упрощения целочисленных алгоритмов масштабы выбираются в виде степени числа 2. Величины X, Y, Z представляют собой масштабированные целочисленные значения переменных x, y, z и связаны с ними операторами масштабирования и округления:

$$X = \left[x \cdot M_x\right]_ц^{1/2}, \quad Y = \left[y \cdot M_y\right]_ц^{1/2}, \quad Z = \left[z \cdot M_z\right]_ц^{1/2}, \tag{2}$$

где – $[\circ]_ц^{1/2}$ оператор округления до целого значения по ½.

Получаемые далее целочисленные алгоритмы операций содержат следующие обозначения:

$$2^s = \frac{M_z}{M_x}, \quad 2^r = \frac{M_z}{M_y}, \quad \frac{M_z}{M_x M_y} = 2^{-m}, \quad \frac{M_z M_y}{M_x} = 2^d, \tag{3}$$

где s, r, p, q – некоторые целые числа. Заметим, что алгоритм целочисленного деления выполняется с использованием операторов div и mod, а полученное частное Z округляется по ½ с учетом значения целого остатка W.

Преобразования (2) позволяют представить вещественные аргументы x и y в целочисленном формате с погрешностями $\beta_x=x-X/M_x$, $\beta_y=y-Y/M_y$, которые подчиняются следующим оценкам:

$$|\beta_x| \le \frac{1}{2M_x}, \quad |\beta_y| \le \frac{1}{2M_y}. \tag{4}$$

Погрешности β_x и β_y являются причиной образования инструментальной погрешности целочисленных алгоритмов $\gamma_z=z-Z/M_z$. В табл.2 приведены предельные абсолютные значения инструментальной погрешности γ_z.

Таблица 2

Арифметическая операция	Оценка инструментальной погрешности целочисленного алгоритма операции
$z = x \pm y$	$|\gamma_z| \le \frac{1}{2}\left(\frac{1}{M_x}+\frac{1}{M_y}+\frac{1}{M_z}\right)$
$z = x \cdot y$	$|\gamma_z| \le \frac{1}{2}\left(|y|_{max}\frac{1}{M_x}+|x|_{max}\frac{1}{M_y}+\frac{1}{M_z}\right)$
$z = x / y$	$|\gamma_z| \le \frac{1}{2}\left(\frac{1}{|y|_{min}}\frac{1}{M_x}+\frac{|x|_{max}}{|y|^2_{min}}\frac{1}{M_y}+\frac{1}{M_z}\right)$

На рис.1 приведена схема модели для экспериментального анализа погрешности γ_z, а также погрешностей β_x и β_y.

Ниже в качестве примера рассматривается операция умножения вещественных переменных $z=x\cdot y$, которую необходимо выполнить в 8-разрядном целочисленном формате данных при $-1\le x\le 1$ и $-2\le y\le 2$. Для этой операции получен целочисленный алгоритм вида

$$Z = [2^{-6}(X \cdot Y)]_{ц}^{1/2} . \tag{5}$$

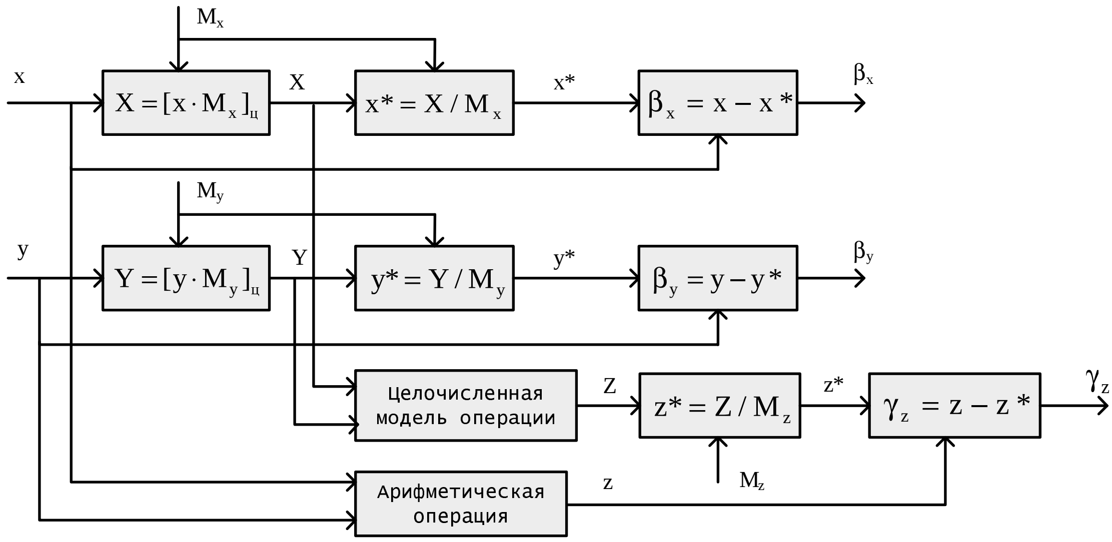

Рис.1. Структура модели для экспериментального анализа погрешностей

На рис.2 приведены результаты моделирования алгоритма (5) в среде MATLAB, которое выполнялось в соответствии с рис.1.

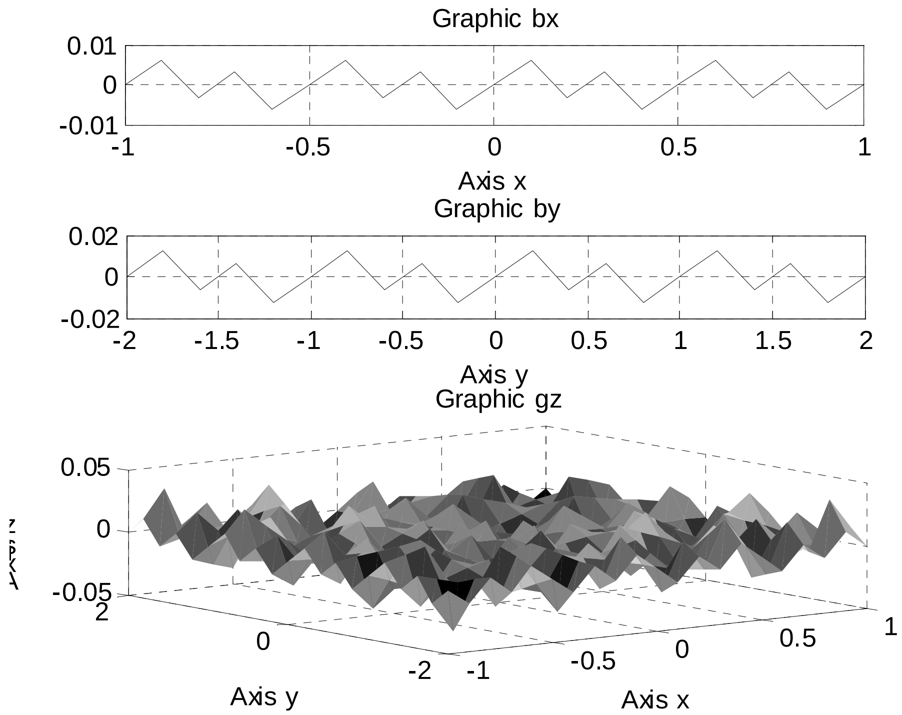

Рис.2. Графики погрешностей β_x, β_y и γ_z

Если используется язык Ассемблер, то целочисленные алгоритмы арифметических операций целесообразно оформить в виде макроопределений. Тогда в основной программе выполнение операций можно задавать с помощью макровызовов, заменяя формальные параметры макроопределений фактическими. Такой подход существенно упрощает разработку и отладку программ, так как программирование целочисленных алгоритмов выполняется один раз в виде 4-х макроопределений (сложения, вычитания, умножения и деления).

Рассмотрим макроопределение на языке Ассемблер IBM PC, реализующее целочисленный алгоритм операции умножения

$$Z = \left[2^{-n}(2^{p}(X \cdot Y))\right]_{ц}^{1/2}, \qquad (6)$$

где n=16. По сравнению с аналогичным алгоритмом из табл.1 здесь коэффициент 2^{-m} представлен в виде $2^{-m}=2^{-n}\cdot 2^{p}$, что в большинстве случаев сокращает время вычислений.

```
MIMUL16  MACRO   X, Y, p, Z
         LOCAL   NULP, SHLZ
         PUSH    AX            ;Сохраняем используемые регистры
         PUSH    CX
         PUSH    DX
         MOV     AX, X         ;Загружаем X
         IMUL    Y             ;Получаем промежуточный результат
         MOV     CX, p         ;Учитываем 2^p
         JCXZ    NULP          ;Переход, если p=0
SHLZ:    SHL     AX, 1
         RCL     DX, 1
         LOOP    SHLZ
NULP:    ADD     AX, 8000H     ;Округляем результат по ½
         ADC     DX, 0
         MOV     Z, DX         ;Сохраняем результат
         POP     DX            ;Восстанавливаем регистры
         POP     CX
         POP     AX
         ENDM
```

Использование языка Ассемблер позволяет довести преимущества режима фиксированной точки до исполняемого кода программы.

Литература

1. Пьявченко О.Н. Проектирование локальных микрокомпьютерных систем. – Таганрог: Изд-во ТРТУ, 2005. – 238 с.
2. Ледовской М.И. Обработка вещественных данных в микроконтроллерах с арифметикой фиксированной точки // Известия ТРТУ. – Таганрог: Изд-во ТРТУ, 2004. №2 (37). – С. 52-58.

Получено 25.05.2009г.