УДК 621.372.061

Я. К. ТРОХИМЕНКО, д-р техн. наук, Н. И. ЯСТРЕБОВ, ст. науч. сотр., Л. В. ГАЕВАЯ, студентка

РЕАЛИЗАЦИЯ НА ЯЗЫКЕ ПЛ/1 АРИФМЕТИЧЕСКИХ ДЕЙСТВИЙ С ЧИСЛАМИ, АБСОЛЮТНЫЕ ВЕЛИЧИНЫ ПОРЯДКОВ КОТОРЫХ ПРЕВЫШАЮТ ДОПУСТИМЫЕ ЗНАЧЕНИЯ

При решении радиотехнических задач на ЭВМ широко используются символьные методы, предполагающие вычисление дробно-рациональных функций комплексной переменной p ($F(p)=A(p)/B(p)$, где $A(p)$ и $B(p)$ — полиномы аргумента p), и их анализ [5]. Полиномы числителя и знаменателя $F(p)$ являются алгебраическими дополнениями матрицы эквивалентных параметров W, поэтому коэффициенты $A(p)$ и $B(p)$ определяются суммами произведений весов определенных комбинаций элементов анализируемой схемы. Если матрица W имеет размерность $n\times n$, то в зависимости от вида схемной функции для $A(p)$ и $B(p)$ приходится вычислять определители n-го, $(n-1)$-го или $(n-2)$-го порядка [1] и, следовательно, находить произведения n, $n-1$, $n-2$ сомножителей. Это может привести к получению таких чисел, которые выходят за диапазон стандартно представимых чисел в ЭВМ. Так как коэффициенты полиномов представляются числами с плавающей точкой, то при программировании на ПЛ/1 их величины не должны быть меньше 5.4E—78 и больше 7.2E+75, ибо в противном случае произойдут ситуации исчезновения порядка (UNDERFLOW) или переполнения порядка (OVERFLOW) соответственно [3]. Указанные ограничения связаны с тем, что в ЭВМ серии ЕС для порядка числа с плавающей точкой во внутреннем представлении отводится только семь бит. При анализе небольших схем для предотвращения подобных ситуаций обычно производят нормирование весов элементов схемы [5], однако для сложных схем в большинстве практических случаев нормирование не дает желаемого результата, так как веса элементов могут отличаться на несколько порядков, например омы и сотни килоом, пикофарады и сотни микрофарад. Кроме того, ситуации UNDERFLOW и OVERFLOW могут возникнуть при вычислении значений самих полиномов для получения частотных и временных характеристик, определения устойчивости посредством нахождения корней полиномов.

В силу изложенного в программах символьного анализа, ориентированных на исследование схем, содержащих десятки узлов и сотни компонентов и более, числа с плавающей точкой представляют раздельно в виде мантиссы M и порядка K и раздельно производят над ними действия [1]. Это позволяет практически избежать ситуаций исчезновения и переполнения порядка, так как целой переменной K можно отвести двух- или четырехбайтовое поле, и максимальная ее абсолютная величи-

на будет соответственно равна 2^{15} или 2^{31} [3]. В наиболее совершенных программах символьного анализа, таких как АС-13 [6] или ТОПАН-4 [2], анализирующие блоки, выполняющие действия над полиномами, написаны на языке Ассемблер и в машинных кодах, что позволяет эффективно оперировать с внутренним представлением чисел и производить их нормализацию после каждой арифметической операции с помощью логических команд. Нами предлагаются эффективные приемы программирования арифметических действий над числами с разделенными мантиссами и порядками на языке ПЛ/1, доступном широкому кругу пользователей ЕС ЭВМ.

Прежде всего отметим, что для переменных с плавающей точкой внутренняя кодированная форма является нормализованной шестнадцатиричной формой с плавающей точкой. В первом бите первого байта машинного представления числа A записывается его знак, в битах со второго по восьмой — порядок AP (в двоичной системе), причем за нуль принято значение AP-64 [3]. В остальных байтах размещена мантисса — по две шестнадцатиричные цифры в каждом байте. Следовательно, значение числа с плавающей точкой равно $\pm AM \cdot 16^{AP-64}$. После каждой арифметической операции с числами, у которых мантиссы и порядки представлены двумя разными переменными, необходимо производить нормализацию мантиссы AM, корректируя соответствующим образом порядок AP. Если число рассматривать как десятичное, то для его нормализации необходимо производить умножение AM на 10^x, что приводит к непроизводительным затратам времени. Однако средства ПЛ/1 позволяют работать со строками битов, и для нормализации AM достаточно записывать подстроку '1000000'B в биты числа AM, отведенные для его порядка. Следует учитывать, что это будет нормализация шестнадцатиричного числа, поэтому в дальнейшем все действия будут выполняться в шестнадцатиричной системе.

Итак, для выделения из числа A его шестнадцатиричного порядка и мантиссы введем соответственно две переменные AP—FIXED BINARY и AM—FLOAT, а также вспомогательную переменную S—строку N битов (N=32, если A объявлено как FLOAT(6), и N=64, если A FLOAT(16)). Оператором S=UNSPEC(A) в строке S получим машинное представление числа A. Отметим, что этому оператору соответствует одна команда Ассемблера MVC [4]. Теперь из этой строки необходимо выделить порядок AP. Может показаться, что наиболее просто это сделать выборкой из S подстроки битов со второго по восьмой функцией SUBSTR(S, 2, 8), переводом ее в число AP и уменьшением последнего на 64. Однако нами установлено, что для выполнения только функции SUBSTR(S, 2, 8) требуется 13 команд Ассемблера, в то время как для выборки одного целого байта из S функций SUBSTR(S, 1, 8) — только одна команда MVC. Исходя из этого выделение порядка AP наибо-

лее эффективно реализуется операторами ПЛ/1

UNSPEC(AP) = (8)'∅'*B*!! SUBSTR(S, 1, 8); (три команды MVC)

IFAP > 127 THEN AP = AP — 192; ELSE AP = AP — 64;

(одно сравнение и одно вычитание). Здесь в условном операторе учтено, что исходное число может быть как положительным, так и отрицательным. Для того чтобы переменной *AM* присвоить значение нормализованной мантиссы числа *A*, достаточно в первый байт строки *S* записать '01000000'B и выполнить оператор UNSPEC (*AM*) = *S*. Кроме того, если *A* отрицательно, необходимо поменять знак у *AM*: *AM* = —*AM*. Все это требует не более трех команд MVC.

Перейдем к операции умножения. Пусть числа *XM* и *YM* — нормализованные мантиссы чисел *X* и *Y*, а *XP* и *YP* — их порядки, полученные вышеизложенным способом. Очевидно, для того чтобы получить результат произведения *H* в виде двух переменных — мантиссы *HM* и порядка *HP*, необходимо перемножить *XM* и *YM*, а *XP* и *YP* сложить, т. е. $HM = XM \times YM$, $HP = XP + YP$. Так как *XM* и *YM* нормализованы, то шестнадцатиричный порядок *HM* может уменьшиться только на единицу. В этом случае *HM* необходимо домножить на 16, а из *HP* вычесть единицу. Такое домножение *HM* на 16 реализуется косвенным путем засылкой в первый байт внутреннего представления *HM* строки битов '01000000'B. Однако если сомножители *XM* и *YM* имели разные знаки, то у *HM* необходимо поменять знак: *HM* = = —*HM*. Таким образом, при реализации умножения работа с подстроками битов организуется тоже побайтно, что сводит к минимуму число команд Ассемблера, реализующих операторы ПЛ/1.

Для выполнения сложения чисел *X* и *Y* необходимо прежде всего уравнять их порядки *XP* и *YP*. При этом вычисляется вспомогательная переменная $K = 64 - (XP - YP)$, если $XP > YP$, или $K = 64 - (YP - XP)$, если $XP < YP$. В том случае, когда $K < 48$, а это будет если $ABS(XP - YP) > 16$, слагаемым с меньшим порядком можно пренебречь, так как оно на сумму не влияет (полагается, что *XM* и *YM* объявлены как FLOAT (16). Если *K* больше или равно 48, то мантисса слагаемого с меньшим порядком, например *YM*, корректируется следующим образом: *S* = UNSPEC(YM); SUBSTR(*S*, 1, 8) = SUBSTR (UNSPEC(*K*, 9, 8)); UNSPEC(*YM*) = *S*; где *S* — вспомогательная строка битов. Эти три оператора ПЛ/1 выполняются тремя командами Ассемблера MVC. Если исходная *YM* была отрицательна, то у скорректированной *YM* необходимо изменить знак. После такого выравнивания порядков получим $HM = XM + YM$; $HP = XP$, и нормализуем *HM* описанным выше способом.

Операция деления выполняется аналогично умножению, с той лишь разницей, что у делимого и делителя порядки соответственно вычитаются, а мантиссы делятся. Кроме того, в результате

деления двух нормализованных мантисс порядок у частного может стать на единицу больше, что учитывается при его последующей нормализации.

При извлечении квадратного корня порядок делится на 2, и корень извлекается только из мантиссы с ее последующей нормализацией. Если порядок был нечетным и положительным, то результат домножается на 4, а если нечетным и отрицательным — на 0,25.

В местах программы, где есть уверенность, что ситуаций OVERFLOW и UNDERFLOW происходить не будет, мантиссу *AM* и порядок *AP* можно снова объединить в одну переменную *A* и работать с ней, как обычно. Для этого используется тот же прием, что и при выравнивании порядков чисел при их сложении S=UNSPEC (*AM*); *AP*=*AP*+64: SUBSTR (*S*, 1, 8)= =SUBSTR(UNSPEC (*AP*), 9, 8); UNSPEC (*A*)=*S*; *IF AM*< <∅ THEN *A*=—*A*.

Предложенные приемы программирования реализованы в процедурах сложения, умножения, извлечения квадратного корня, деления чисел, представленных раздельно мантиссами и порядками, в программах формирования и исследования схемных функций на языке ПЛ/1. При этом учтены ситуации, когда один из операндов равен нулю. Время вызова и выполнения этих процедур на ЭВМ ЕС-1033 находится в пределах сотен микросекунд.

1. *Блажкевич Б. И., Мочернюк Ю. П.* Метод системных графов и его применение для анализа линейных систем. Львов, 1977. 56 с. (Препр. / АН УССР ФМИ; № 2). 2. *Ларин А. Г., Томашевский Д. И., Шумков Ю. М., Эйдельнант В. М.* Машинная оптимизация электронных узлов РЭА. М.: Сов. радио, 1978. 192 с. 3. *Лепин-Дмитрюков Г. А.* Программирование на языке ПЛ/1 (для ДОС ЕС ЭВМ). М.: Сов. радио, 1978. 288 с. 4. *Рад У.* Программирование на языке Ассемблера и вычислительные системы IBM-360 и 370. М.: Мир, 1979. 592 с. 5. *Трохименко Я. К.* Проектирование радиотехнических схем на инженерных ЭЦВМ. К.: Техніка, 1976. 272 с. 6. *Шаповалов Ю. И., Давидюк Р. Д.* Особенности реализации метода топологического анализа схем в программе ACIЗEC // Изв. вузов СССР. Радиоэлектроника. 1983. № 6. С. 79—81.

Поступила в редколлегию 15.09.86

УДК 621.3.012.8

В. С. ВУНТЕСМЕРИ, канд. техн. наук

СОГЛАСОВАНИЕ ГЕЛИКОНОВОГО ВЕНТИЛЯ

В работе [1] получена матрица рассеяния геликонового вентиля. Элементы ее представляют собой коэффициенты отражения и передачи при включении вентиля в линию передачи с волновым сопротивлением Z_0[Ом]. Анализ коэффициентов $S_{11}S_{22}$ показывает, что входное сопротивление вентиля носит индуктивный характер. Поэтому для узкополосного согласования достаточно включить на входах вентиля емкости, величина