Х. К. БРУТЯН

ОБ АВТОМАТИЧЕСКОМ ПОВЫШЕНИИ ТОЧНОСТИ ВЫЧИСЛЕНИЙ НА МАШИНЕ С ФИКСИРОВАННОЙ ЗАПЯТОЙ

Ограниченность длины разрядной сетки ячеек запоминающего устройства в современных электронных цифровых вычислительных машинах (ЭЦВМ) часто приводит к потере необходимой точности при вычислениях, а в некоторых случаях даже к неправильному решению задачи. Чтобы избежать этого, естественно, для запоминания достаточно „длинного" числа использовать несколько ячеек, в которых записывать такое число по частям.

В настоящей работе описывается методика вычислений на ЭЦВМ для случая, когда исходные числа записываются в m ячейках запоминающего устройства (m — произвольное число). Отметим, что частный случай ($m=2$) для вычислений на ЭЦВМ „БЭСМ-2" рассматривается в работе [1].

Как известно, при вычитании близких друг другу чисел количество значащих цифр может сократиться, а это, в свою очередь, приводит к понижению относительной точности результатов вычислений. Поэтому нужно, по возможности, избегать таких вычитаний, что может быть осуществлено либо преобразованием формул, либо изменением порядка вычислений.

В качестве примера преобразования формул можно рассмотреть случай вычисления выражения

$$\pi (R+\Delta R)^2 - \pi R^2,$$

в котором при малых ΔR разность очень мала, но после преобразования этой формулы к виду

$$2\pi R \cdot \Delta R + \pi \cdot \Delta R^2$$

вычисления уже не приводят к указанной потере значащих цифр. Иногда можно избежать вычитания близких друг другу чисел также путем изменения порядка вычислений. Указанное изменение порядка вычислений часто может быть достигнуто посредством удобной группировки действий. Например, это можно осуществить при суммировании знакопеременных рядов, сложив в них отдельно все положительные слагаемые, отдельно все отрицательные, и только после этого вычислив разность полученных сумм. Следует, однако, заметить, что в некоторых случаях ни один из указанных методов не приводит к желаемому результату. В качестве примера потери значащих цифр при вычитании близких друг другу чисел мы рассмотрим задачу о вычислении следующего знакопеременного ряда:

$$S=\sum_{n=1}^{\infty}(-a)^n \frac{1}{4^n n! \prod_{i=1}^{n}(4i-2)(4i-3)^2}, \text{ где } a=0{,}1125\cdot 10^{10}. \qquad (1)$$

Из таблицы, приводимой ниже, видно, что сумма этого ряда имеет порядок 10^{57} и что эта величина с достаточной точностью приближается к сумме первых 90 членов ряда. При этом, однако, промежуточные частичные суммы ряда имеют гораздо большую величину (например, сумма первых 45 членов имеет порядок 10^{80}). Поэтому, если мы вычисляем требуемую частичную сумму непосредственным суммированием членов ряда, то нам приходится терять при вычитаниях по крайней мере 23 значащие десятичные цифры. Не видно никакого способа избежать этой потери значащих цифр, не видно также и какого-либо иного способа подсчета суммы ряда. В подобных случаях для получения удовлетворительной точности ответа необходимо резкое повышение точности вычислений.

В настоящей статье рассматривается один из способов повышения точности вычислений при решении задач указанного типа, разработанный в ВЦ АН АрмССР. Предлагаемый способ заключается в том, что каждое число записывается с повышенной точностью, причем недостаточная длина разрядной сетки машины компенсируется путем записи числа в нескольких ячейках. Так, например, при использовании указанного способа для расчетов на машине М-3 применялась запись чисел с повышенной точностью в 2, 3, 4 ячейках машины, т. е. 60, 90, 120 двоичных разрядах соответственно.

Опишем вначале результаты вычисления суммы ряда (1) на машине М-3 рассматриваемым способом. В этих расчетах все числа представлялись в виде

$$10^{Pa}\cdot A, \text{ где } 0{,}1 \leqslant |A| < 1,$$

причем порядок Pa хранился в отдельной ячейке, а мантисса изображалась с помощью 30, 60, 90 или 120 разрядов. Соответственно с этими случаями мы различаем четыре различные частичные суммы

$$S_n^p = \sum_{k=1}^{n} U_k^p.$$

(Здесь p равняется поделенному на 30 числу двоичных разрядов, использованных для изображения мантиссы A. Фактически индекс p

при сумме S_n^p означает число ячеек, использованных для изображения мантиссы при вычислении этой суммы).

После выполнения каждого действия результат подвергался нормализации. Для каждого p вычисления велись до тех пор, пока прибавление очередного числа уже не вызывало изменения в соответствующей частичной сумме. Результаты этих вычислений приведены в табл. 1, в которой, кроме сумм S_n^1, S_n^2, S_n^3 и S_n^4, для некоторых значений n приведены также соответствующие значения членов ряда U_n^1 (в первом столбце)*. Все числа в таблице приведены с семью значащими цифрами. В столбце S_n^p в левой колонке приведены значения порядка Pa и в средней колонке — значения мантиссы A. Смысл третьей колонки — S_n^p — станет ясен несколько ниже. Как видно из этой таблицы, первые члены нашего ряда очень быстро возрастают по модулю, но, начиная с 47-го члена, они убывают (ясно, что $U_n^p \to 0$ при $n \to \infty$ для всех p). Сумма S_n^1 достигает максимума при $n = 46$, но после этого, вследствие знакопеременности ряда, сумма S_n^1 начинает резко убывать. Например, при переходе от S_{46}^1 к S_{47}^1 пропадает одна значащая цифра, в связи с чем порядок S_{47}^1 меньше порядка S_{46}^1 на единицу.

Так как обычно все вычисления ведутся с фиксированным числом разрядов, результат вычисления, в силу указанной потери значащих цифр, может быть получен с очень малой точностью или даже совершенно неверным. В нашем примере сумма ряда S_{78}^1 отличается от суммы S_{90}^4 почти в 10^{14} раз, а сумма S_{86}^2 от суммы S_{90}^4 почти в 10^5 раз.

Подсчет потери точности (а следовательно, и необходимого числа разрядов для избежания этих потерь) легко произвести по следующим правилам. Пусть порядок n-й частичной суммы S_n знакопеременного ряда отличается от наибольшего порядка тех же сумм S_l при $l < n$ на Δ единиц. Чтобы сумма S содержала k верных значащих цифр, необходимо вести все вычисления с числом разрядов, равным $k + \Delta$. Как известно, для изображения одного десятичного разряда требуется приблизительно 3,32 двоичных разряда. Следовательно, для получения $k + \Delta$ верных десятичных знаков необходимо использовать $(k + \Delta) \cdot 3{,}32$ двоичных разряда. Для иллюстрации сказанного рассмотрим табл. 1. Как видно из нее, S_{46}^p для всех четырех случаев имеет одинаковые значения, но, начиная с $n > 46$, суммы S_n^p в разных столбцах становятся различными и уже для $n = 68$ S_{68}^1 и S_{68}^2 не имеют ни одного общего знака. Действительно, 30 двоичных разрядов могут обеспечить точность только в 9 десятичных значащих

---

* Столбцы U_n^2, U_n^3, U_n^4, если брать их с точностью до семи значащих цифр, полностью совпадают со столбцом U_n^1, поэтому они не приведены в таблице.

цифрах. Между тем уже разность порядков S^1_{68} и S^1_{46} равна 9, т. е. в S^1_{68} уже нет ни одной верной значащей цифры.

Тем же самым объясняются и остальные случаи расхождения значений S^p_n при фиксированном n и различных p. Например, S^3_{90} и S^4_{90} имеют 4 одинаковые значащие цифры. Поскольку разность порядков S^4_{45} и S^4_{90} равна 23, то для получения 4 верных значащих цифр вычисления должны были вестись, по крайней мере, с 27 десятичными разрядами или с 89 двоичными разрядами, что имеет место только для случаев $p=3$ и $p=4$. Анализ значений сумм в двух соседних столбцах одной строки таблицы показывает, что если разность порядков между суммой S^p_n и наибольшей из предыдущих сумм S^p_l, $l<n$ равна Δ и если счет идет по p ячейкам, то ответ содержит

$$k = 9p - \Delta$$

верных знаков. Это число k приведено в правой колонке каждого из четырех столбцов таблицы.

Автоматический контроль потери относительной точности вычислений, связанный с потерей значащих цифр при вычитании близких чисел, легко осуществить для того случая, когда при вычислениях выполняются только операции суммирования и вычитания. Например, это имеет место для вышеприведенного примера, где мы полагаем, что все числа ряда заранее вычислены, и речь идет только о подсчете суммы: $S_n = U_1 + U_2 + U_3 + \cdots + U_n$. Такой автоматический контроль проводится на основании анализа разности

$$9p - \Delta.$$

Как только эта разность становится меньше требуемого числа верных знаков в окончательном ответе, вычисления прерываются и происходит программный переход к вычислению суммы ряда с помощью увеличенного на единицу числа ячеек.

Ниже приводятся схемы программ, с помощью которых этот метод был реализован на машине М-3. Машина М-3 двухадресная, с фиксированной запятой, длина разрядной сетки ячейки ЗУ—30 разрядов. Кроме того, имеется знаковый разряд. Машина не имеет дополнительных разрядов для округления. При реализации схем на машине М-3 числа изображаются в виде $2^{Pa} \cdot A$, причем порядок Pa хранится в одной ячейке, а мантисса A хранится в m ячейках, где число m автоматически выбирается в процессе вычислений. При этом число Pa записывается как целое число в младшие разряды некоторой „ячейки" α. Иными словами, предполагается, что в ячейке α запятая фиксирована справа от самого младшего разряда. Мантиссы записываются в 28 младших разрядах соответствующих m ячеек (старшие два разряда каждой ячейки свободны). Такая система записи была принята из соображений удобства реализации метода. Впредь, говоря о „ячейках", в которых хранятся мантиссы чисел, мы будем предполагать, что каждая такая „ячейка" содержит 28 разрядов (это

Таблица 1

Порядковый номер n-го члена ряда	Величина соответствующего члена ряда ($U_n^1=10^{Pa}\cdot A$, $0{,}1<\|A\|<1$)		$S_n^p=\sum\limits_{k=1}^{n} U_k^p$ $\left(S_n^p=10^{Pa}\cdot A,\ \text{где}\ \frac{1}{10}<\|A\|<1\right)$											
			S_n^1			S_n^2			S_n^3			S_n^4		
	Pa	A	Pa	A	k	Pa	A	k	Pa	A	k	Pa	A	k
1	09	—1406249	09	—1406249	>7	09	—1406249	>7	09	—1406249	>7	09	—1406249	>7
2	15	1318359	15	1318357	>7	15	1318357	>7	15	1318357	>7	15	1318357	>7
3	20	—1525878	20	—1525865	>7	20	—1525865	>7	20	—1525865	>7	20	—1525865	>7
4	24	4534588	24	4534435	>7	24	4534435	>7	24	4534435	>7	24	4534435	>7
45	80	—1979697	80	—1024806	>7	80	—1024806	>7	80	—1024806	>7	80	—1024806	>7
46	80	2030042	80	1005235	>7	80	1005235	>7	80	1005235	>7	80	1005235	>7
47	80	—1908285	79	—9030500	>7	79	—9030500	>7	79	—9030500	>7	79	—9030500	>7
55	78	—6251727	78	—2010162	7	78	—2010162	>7	78	—2010162	>7	78	—2010162	>7
56	78	2895789	77	8856275	6	77	8856273	>7	77	8856273	>7	77	8856273	>7
58	77	5020846	77	1388594	6	77	1388593	>7	77	1388593	>7	77	1388593	>7
59	77	—1884038	76	—4954437	5	76	—4954452	>7	76	—4954452	>7	76	—4954452	>7
61	76	—2167058	75	—5152101	4	75	—5152247	>7	75	—5152247	>7	75	—5152247	>7
63	75	—1917420	74	—4121291	3	74	—4122750	>7	74	—4122750	>7	74	—4122750	>7
67	72	—7060885	72	—1098066	1	72	—1244059	>7	72	—1244059	>7	72	—1244059	>7
68	72	1494770	71	3967032	0	71	2507103	>7	71	2507103	>7	71	2507103	>7
71	69	—9997804	71	1445439	0	69	—1448956	7	69	—1448956	>7	69	—1448956	>7
72	69	1681159	71	1462250	0	68	2322031	6	68	2322026	>7	68	2322026	>7
74	67	4026828	71	1459979	0	66	5057022	4	66	5056520	>7	66	5056520	>7
77	64	—9969765	71	1459928	0	64	—1037661	2	64	—1087955	>7	64	—1087955	>7
78	64	1214518	71	1459928	0	63	1768574	1	63	1265633	>7	63	1265633	>7
79	63	—1405566				62	3630080	0	62	—1399322	>7	62	—1399322	>7
81	61	—1618265				62	5014605	0	60	—1479816	7	60	—1479817	>7
82	60	1611922				62	5030725	0	59	—1321057	6	59	—1321051	>7
84	58	1382391				62	5029334	0	57	—6986305	4	57	—6986879	>7
86	55	9799129				62	5029323	0	57	—8079907	4	57	—8080482	>7
87	54	—7692127				62	5029323	0	57	—8087600	4	57	—8088174	>7
90	51	2829553							57	—8087061	4	57	—8087636	>7
91	50	—1853727							57	—8087062	4	57	—8087636	>7

28 младших разрядов истинной ячейки). Роль знакового разряда при этом не меняется. Чтобы отличить такие „ячейки“ от обычных, мы будем ставить кавычки. Рабочие ячейки будем обозначать через α_i. Появление единицы во втором разряде ячейки (не включая в „ячейку“) будет означать обычно наличие переноса в старшие разряды той же мантиссы, хранящиеся в другой „ячейке“. Поэтому указанный разряд мы будем называть разрядом переноса.

Пусть мантиссы чисел $b = 2^{Pb} \cdot B$, $c = 2^{Pc} \cdot C$ хранятся, соответственно, в „ячейках“ $b+1$, $b+2, \ldots, b+m$ и $c+1$, $c+2, \ldots, c+m$, причем содержимое каждой „ячейки“ имеет знак мантиссы. В дальнейшем содержимое „ячейки“ α обозначается через $\overline{\alpha}$: адрес „ячейки“, в которой хранится константа a, обозначается через „a“. Для простоты описания введем несколько операторов.

I. Оператор нормализации A

Под нормализацией числа $b = 2^{Pb} \cdot B$ понимается приведение этого числа к такому виду $b = 2^{Pb'} \cdot B'$, при котором мантисса B' удовлетворяет неравенствам

$$\frac{1}{2} \leqslant |B| < 1.$$

Нормализация выполняется по следующим правилам: Если $\overline{b+1}$, $\overline{b+2}, \ldots, \overline{b+k}$ $(k < m)$ равны нулю, то $\overline{b+k+1}$, $\overline{b+k+2}, \ldots, \overline{b+m}$ передаются, соответственно, в „ячейки“ $b+1$, $b+2, \ldots, b+m-k$ (в случае $k = m$ нормализация не происходит).

Далее, для нормализации нужно выполнить сдвиг влево $\overline{b+1}$, $\overline{b+2}, \ldots, \overline{b+m-k}$ на столько разрядов, сколько нулей имеется в „ячейке“ $b+1$ до первой слева единицы. При этом такое же количество начальных разрядов каждой „ячейки“ $b+i$ должно быть передано в освобождающиеся разряды предыдущей „ячейки“ $b+i-1$, где $i = 2, 3, \ldots, m$. Функционирование оператора A легко может быть описано в терминах следующих операторов:

оп. A_1 — определяет, на сколько разрядов должно быть сдвинуто $\overline{b+1}$. Это число записывается как целое число в младшие разряды „ячейки“ α_4.

оп. A_2 — формирует константы для оператора A_3.

оп. A_3 — осуществляет следующие преобразования:

1) производит сдвиг числа $\overline{b+i}$ влево на $\overline{\alpha_4}$ единиц;
2) выделяет начальные $\overline{\alpha_4}$ разрядов числа $\overline{b+i+1}$, результат переносит в последние разряды ячейки и складывает с числом $\overline{b+i}$;
3) превращает в нуль начальные $\overline{\alpha_4}$ разрядов числа $\overline{b+i+1}$.

оп. A_4 — осуществляет соответствующий сдвиг в „ячейке“ $b+m-k$. Схемная реализация оператора A имеет вид

$$\prod_{k=1}^{m}\Big\{P(\overline{b+1}=\text{„}O\text{“})^{\uparrow 1}(\overline{Pb}-\text{„}28\text{“ II } A\to Pb)\prod_{j=1}^{m-1}(b+j+1\to b+j)\Big\}$$

$$(\text{„}O\text{“}\to Pb)^{\uparrow 2\downarrow 1}A_1P(\overline{\alpha}_4\neq\text{„}O\text{“})^{\uparrow 2}(\overline{Pb}+\overline{\alpha}_4\to Pb)A_2(\text{„}O\text{“}\to\alpha_2)$$

$$\prod_{l=1}^{m-1}(A_3^l)A_4^{\downarrow 2}.$$

II. Оператор сдвига при выравнивании порядков B

При выравнивании порядков чисел для их сложения (или вычитания) мантисса числа с меньшим порядком должна быть сдвинута вправо на число разрядов, равное разности порядков этих чисел (разность порядков записывается в „ячейке“ α).

Обозначим через t величину $\left[\frac{\alpha}{28}\right]$. Если $m>t>0$, то нужно t раз совершить следующую операцию: передать содержимое всех „ячеек“ $b+i$ в „ячейки“ $b+i+1$ ($i=m-1,\ m-2,\ldots,\ 1$), а в „ячейку“ $b+1$ заслать нуль. После этого содержимое всех „ячеек“ $b+i$ нужно сдвинуть вправо на $\left\{\frac{\alpha}{28}\right\}\cdot 28$ разрядов; при этом соответствующее количество последних разрядов каждой „ячейки“ должно быть передано в освобождающиеся разряды следующей „ячейки“. Если $t=0$, то все действия производятся так же, с тем лишь исключением, что операция передачи содержимого „ячеек“ $b+i$ в $b+i+1$ вообще не производится.

Введем операторы:

оп. B_1 — формирует константы для оператора B_2^j,

оп. B_2^j — осуществляет следующие преобразования:

1) выделяет последние $\left\{\frac{\alpha}{28}\right\}\cdot 28$ разрядов числа $\overline{b+j}$, результат переносит в начальные разряды „ячейки“ и запоминает в рабочей ячейке α_5;

2) производит сдвиг числа $\overline{b+j}$ вправо на $\left\{\frac{\alpha}{28}\right\}\cdot 28$ разрядов и результат складывает с числом $\overline{\alpha}_6$.

Функционирование оператора B можно изобразить в виде следующей схемы:

$$\prod_{k=1}^{m}\Big\{P(\overline{|\alpha|}\geqslant\text{„}28\text{“ II } A)^{\uparrow 2}(\overline{|\alpha|}-\text{„}28\text{“ II } A\to\alpha)\prod_{i=m-1}^{k}(b+i\to b+i+1)\times$$

$$\times(\text{„}O\text{“}\to b+k)\Big\}^{\uparrow 1}.$$

$$\downarrow^{2}P(\overline{\alpha}\neq\text{„}O\text{“})^{\uparrow 1}B_1(\text{„}O\text{“}\to\alpha_5)\prod_{j=k}^{m}(\alpha_5\to\alpha_6)\cdot B_2^{j\downarrow 1}.$$

III. Оператор приведения к одному знаку C

При алгебраическом сложении двух чисел сначала выравниваются порядки, а потом уже складываются мантиссы. При этом, если мантиссы записаны в m „ячейках“

$$b+1,\ b+2, \ldots,\ b+m,$$
$$c+1,\ c+2, \ldots,\ c+m,$$

то каждая пара $b+i$ и $c+i$ складывается (или вычитывается) с помощью операций сложения (или вычитания), имеющихся в машине. Как мы указывали выше, знаковый разряд каждой из „ячеек“ мантиссы имеет одно и то же значение, совпадающее со знаком самой мантиссы. Следовательно, после выполнения операции сложения над каждой отдельной парой „ячеек“ результаты могут иметь разные знаки для разных пар. Оператор C изменяет мантиссу таким образом, чтобы каждая из „ячеек“ имела один и тот же знак, совпадающий со знаком старшей, отличной от нуля, суммы $\overline{b+i}+\overline{c+i}$. Пусть в „ячейках“ $b+1,\ b+2, \ldots,\ b+m$ хранится мантисса результата действия. Выравнивание знаков производится по следующим правилам:

1. Если $\overline{b+1}>0$ и для некоторых $i=2, 3, \ldots, m$ числа $\overline{b+i}<0$, то из $\overline{b+i-1}$ вычитается единица младшего разряда, а к $\overline{b+i}$ прибавляется число, у которого все разряды равны нулю, кроме разряда переноса, в котором стоит единица ($010\cdots0$).

2. Если $\overline{b+1}<0$ и для некоторых $i=2, 3, \ldots, m$ числа $\overline{b+i}>0$, то к $\overline{b+i-1}$ прибавляется единица младшего разряда, а из $\overline{b+i}$ вычитается такое же число, как построенное в предыдущем пункте.

3. Если $\overline{b+1}=0$, то переходим к анализу $\overline{b+2}$ и т. д., пока не достигаем первой „ячейки“, содержимое которой отлично от нуля.

Введем операторы:

оп. $\widetilde{D}_1^j$ — осуществляет операцию, описываемую правилом 1.

оп. $\widetilde{D}_2^j$ — осуществляет операцию, описываемую правилом 2.

Работу оператора C можно изобразить в виде следующей схемы:

$$\prod_{k=1}^{m} \{P(\overline{b+k} \neq „O“)\uparrow^{5} P(\overline{b+k} > „O“)\uparrow^{6}\downarrow^{5}\}\uparrow^{1}\downarrow^{6}(„1“ \to \alpha)\uparrow^{2}$$

$$\downarrow^{1}(„-1“ \to \alpha)\downarrow^{2}\prod_{j=1}^{m} \{P(\bar{\alpha}\cdot\overline{\overline{b+j}+1} < 0)\uparrow^{4} P(\bar{\alpha} > 0)\uparrow^{3}\widetilde{D}_1^j\uparrow^{2}\downarrow^{3}\widetilde{D}_2^j\uparrow^{2}\downarrow^{4}\}.$$

IV. Схема алгебраического сложения

Опишем схему, по которой выполняется сложение. Рассмотрим правила сложения одной пары „ячеек“ $b+i$ и $c+i$. Пусть результат суммирования записывается в „ячейку“ $b+i$. Если при этом сложении в рязряде переноса образовалась единица, то эта единица, взятая со знаком числа $\overline{b+i}$, прибавляется к младшему разряду в „ячейку“ $\overline{b+i-1}$, а в „ячейке“ $b+i$ в разряд переноса засылается нуль. Иными словами, если в „ячейке“ образовалось число

$$\pm 01\,\gamma_1\gamma_2\gamma_3\cdots\gamma_{28},$$

то к содержимому „ячейки“ $b+i-1$ прибавляется число $\pm 00\cdots01$, а в „ячейке“ $b+i$ образуется число $\pm 00\,\gamma_1\gamma_2\cdots\gamma_{28}$. Число $\pm 00\cdots01$, которое мы будем называть единицей переноса, формируется в ячейке α_3. Практически в программе не производится проверка разряда переноса на наличие в нем нуля, а просто из каждой „ячейки“ $b+i$ выделяется содержимое этого разряда, сдвигается на 28 разрядов вправо, затем ему присваивается знак числа $\overline{b+i}$ и полученный результат прибавляется к содержимому „ячейки“ $b+i-1$. Эти операции выполняет оператор D_1^i, который для всех $i=1, 2, \ldots, m$ производит следующие действия:

а) складывает содержимое „ячеек“ $b+i$ и $c+i$;

б) к результату прибавляет единицу переноса из младшей пары $b+i+1$, которая уже содержится в „ячейке“ α_3 (для $i=m$, $\overline{\alpha}_3=0$);

в) формирует в „ячейке“ α_3 новую единицу переноса в старшую пару $b+i-1$;

г) засылает нуль в разряд переноса „ячейки“ $b+i$.

Теперь сложение чисел b и c и запись результата в „ячейки“ $b+1$, $b+2, \ldots, b+m$ осуществляются следующим образом:

1. Вычисляется разность $Pb-Pc$ и засылается в ячейку α, затем выравниваются порядки чисел b и c.

2. Если $\overline{\alpha}>0$, то мантисса C сдвигается вправо на число разрядов, равное $\overline{\alpha}$ (оператор B); если $\overline{\alpha}<0$, то мантисса B сдвигается вправо на число разрядов, равное $\overline{\alpha}$ (оператор B).

3. Вычисляется порядок суммы, равный $\max(Pb, Pc)$.

4. Работает оператор D_1^i при $i=1, 2, \ldots, m$.

5. Если результат сложения больше единицы, т. е. в разряде переноса „ячейки“ $b+1$ образовалась единица, то порядок результата увеличивается на единицу, а мантисса сдвигается вправо на один разряд.

Оператор, который проверяет разряд переноса в „ячейке“ $b+1$ на наличие в нем единицы, обозначим через D_2.

6. Знак каждой „ячейки“ $b+i$ $(i=1, 2\ldots, m)$ приводится к знаку $b+1$ (оператор C).

7. Результат суммирования нормализуется (оператор A).

В терминах всех введенных операторов схема алгебраического сложения двух чисел имеет вид

$$(\overline{Pb}-\overline{Pc}\to\alpha)\,P(\overline{\alpha}>0)^{\uparrow^1}B(c)^{\uparrow^2\downarrow^1}B(b)\cdot(Pc\to Pb)^{\downarrow^2}(0\to\alpha_3)$$

$$\prod_{l=m}^{1}(D_1^l)\cdot P(\overline{\alpha}_3\neq 0)^{\uparrow^3}D_2(\text{„1 II }A\text{“}\to\alpha)\,B(b)\,(\overline{Pb}+\text{„1 II }A\text{“}\to Pb)^{\downarrow^3}CA.$$

Вычитание приводится к сложению путем изменения знака вычитаемого.

V. Схема умножения

Пусть заданы два числа $b=2^{Pb}\cdot B$ и $c=2^{Pc}C$,

$$B=\sum_{l=1}^{m}b_l\cdot 2^{-28(l-1)},\qquad C=\sum_{l=1}^{m}c_l\cdot 2^{-28(l-1)}$$

и требуется построить произведение этих чисел $bc=2^{Pb+Pc}\cdot BC$. Пусть мантиссы B и C хранятся, как обычно, в „ячейках“ $b+1$, $b+2,\ldots$, $b+m$ и $c+1$, $c+2,\ldots$, $c+m$. Построим произведение BC этих мантисс и поместим его в „ячейки“ $b+1$, $b+2,\ldots$, $b+m$. Для этого выделим содержимое старших четырнадцати разрядов „ячейки“ $c+i$ и поместим их в „ячейку“ φ_{2l-1} со 2-го разряда по 15-й разряд включительно. Младшие 14 разрядов „ячейки“ $c+i$ передадим в разряды 2—15 ячейки φ_{2l}. Точно так же разделим пополам содержимое „ячеек“ $b+i$ и поместим первую и вторую половины в разряды 2—15 ячеек ψ_{2l-1} и ψ_{2l}, соответственно, здесь $i=1, 2,\ldots, m$, и вышеописанные действия выполняет оператор D_1^l.

Легко видеть, что имеет место формула

$$BC=\left[\sum_{l=1}^{m}(\overline{\varphi}_{2l-1}+2^{-14}\overline{\varphi}_{2l})\,2^{-28(l-1)}\right]\cdot\left[\sum_{l=1}^{m}(\overline{\psi}_{2l-1}+2^{-14}\overline{\psi}_{2l})\,2^{-28(l-1)}\right]=$$

$$=\left(\sum_{l=1}^{2m}\overline{\varphi}_l2^{-14(l-1)}\right)\left(\sum_{l=1}^{2m}\overline{\psi}_l2^{-14(l-1)}\right)\approx\sum_{l=1}^{2m}\left(\sum_{j=1}^{l}\overline{\varphi}_j\overline{\psi}_{l-j+1}\right)2^{-14(l-1)}.$$

Каждое из чисел $\overline{\varphi}_l$, $\overline{\psi}_{l-j+1}$ является пятнадцатиразрядным двоичным числом. Поэтому произведение $\overline{\varphi}_l\cdot\overline{\psi}_{l-j+1}$ образует не более чем тридцатиразрядное двоичное число. Так как старшие разряды чисел $\overline{\varphi}_l$ и $\overline{\psi}_{l-j+1}$ равны нулю, то практически результат их произведения может заполнить разряды с 3-го по 30-й включительно.

Обозначим через ν_l сумму

$$\sum_{j=1}^{l}\overline{\varphi}_j\overline{\psi}_{l-j+1},$$

а через D_2^l оператор, который в некоторой „ячейке“ ω_l накапливает сумму ν_l и, кроме того, в некоторой „ячейке“ α_4 (в ее младших разрядах) накапливает сумму всех единиц переноса, образующихся при накапливании числа $\overline{\omega}_l$, т. е. ν_l. Знак $\overline{\alpha}_4$ совпадает со знаком ν_l (ясно, что все ν_l имеют один и тот же знак).

В силу введения обозначений

$$B\cdot C=\sum_{l=1}^{2m}\nu_l 2^{-14(l-1)}.$$

Каждое из ν_l домножается на вес $2^{-14(l-1)}$; это означает, что перед суммированием чисел ν_l каждое из них должно быть сдвинуто относительно числа ν_{l-1} ($i=2, 3,\dots, m$) на 14 разрядов вправо. Таким образом, результирующее произведение мантисс BC, которое мы хотим получить в „ячейках“ $b+1$, $b+2,\dots, b+m$, можно образовать по следующему правилу: содержимое „ячейки“ $b+i$ ($i=m, m-1,\dots, 2$) равно сумме трех чисел:

а) числа $\overline{\omega}_{2l}$;

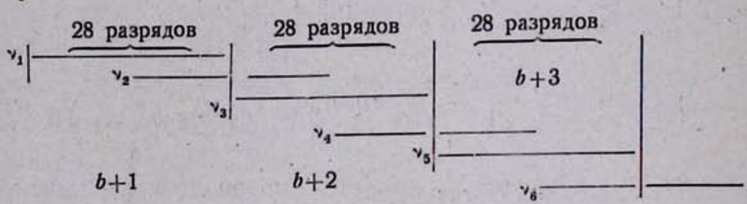
б) числа, образованного старшими 14 разрядами числа $\overline{\omega}_{2l-1}$, сдвинутыми на 14 разрядов вправо;

в) числа, образованного младшими 14 разрядами числа $\overline{\omega}_{2l+1}$, сдвинутыми на 14 разрядов влево.

Если при таком суммировании в разряде переноса „ячейки“ $b+i$ образуется единица, то она должна быть прибавлена (со своим знаком) к младшему разряду ячейки $b+i-1$.

Содержимое ячейки $b+1$ равно сумме числа $\overline{\omega}_1$ и числа, образованного старшими 14 разрядами $\overline{\omega}_2$, предварительно сдвинутыми на 14 разрядов вправо.

Сказанное можно проиллюстрировать следующей схемой расположения разрядов чисел $\overline{b+1}$, $\overline{b+2}$, $\overline{b+3}$ (для случая $m=3$)



Оператор, который строит содержимое „ячейки“ $b+i$ ($i=2, 3,\dots, m$), мы обозначим через D_3^l, а оператор, который строит содержимое „ячейки“ $b+1$, обозначим через D_4. Теперь схема программы, вычисляющей произведение мантисс BC, запишется следующим образом:

$$\prod_{k=1}^{m}[D_1^k(b, \varphi)\cdot D_1^k(c, \varphi)]\cdot(„O“\to\alpha_4)\prod_{l=2m}^{1}\Big\{(\alpha_4\to\alpha_2)(„O“\to\alpha_4)(\alpha_4\to\alpha_5)$$

$$\prod_{j=l}^{1}D_2^j(\alpha_2\to\omega_l)\Big\}\cdot\prod_{l=n}^{2}D_3^l D_4 A.$$

Построение произведения двух чисел производится при помощи следующих операций:

1. Вычисляется порядок произведения, равный сумме порядков сомножителей.

2. Строится произведение мантисс.

3. Если это необходимо, результат произведения нормализуется.

Для деления двух чисел $c:b$ строится величина $1:b$, которая вычисляется по итерационной формуле

$$d_{n+1} = d_n(2 - bd_n). \text{ где } n = 1, 2, 3, \ldots$$

Здесь $d \to \frac{1}{b}$ при $0 < d_n < \frac{1}{b}$ и в качестве d_1 фактически при реализации метода берется число, равное $2^{-Pb} \cdot 1$, если число b имеет вид $b = 2^{Pb} \cdot B$, где $\frac{1}{2} \leqslant B < 1$. Затем производится умножение $\frac{1}{b} C$.

В заключение выражаю глубокую благодарность кандидату физико-математических наук Т. М. Тер-Микаэляну за ряд ценных и полезных советов в этой работе.

Խ. Կ. ԲՐՈՒՏՅԱՆ

ՖԻՔՍԱԾ ՍՏՈՐԱԿԵՏՈՎ ԱՇԽԱՏՈՂ ՄԵՔԵՆԱՆԵՐԻ ՎՐԱ ԱՎՏՈՄԱՏԻԿ ԿԵՐՊՈՎ ՀԱՇՎՈՒՄՆԵՐԻ ՃՇՏՈՒԹՅՈՒՆԸ ՄԵԾԱՑՆԵԼՈՒ ՄԱՍԻՆ

Ա մ փ ո փ ո ւ մ

Էլեկտրոնային հաշվիչ մեքենաներում թիվը սովորաբար պահվում է մի բջիջում, բայց վերջինի կարգերի սահմանափակության պատճառով որևէ խնդրի սկզբնական տվյալները և նրանց միջոցով հետագա հաշվումները կատարվում են մոտավոր կերպով, իսկ դա որոշ չափով ազդում է արդյունքի ճշտության վրա։ Ինչպես ցույց է տրված հոդվածում, լինում են դեպքեր, երբ հաշվման արդյունքում ստացվում է լրիվ սխալ պատասխան, չնայած նրան, որ բոլոր նախնական տվյալները և հաշվումները կատարվել են 7 նշանակալի թվանշանների ճշտությամբ։ Դա լինում է այն պատճառով, որ հաշվման ընթացքում ստացվում են մոտիկ թվեր, իսկ նրանց հետ հանման գործողության կատարելիս արդյունքում ստացվում է ավելի պակաս նշանակալի թվանշաններ, քան ունեին թվերը նախքան հանման գործողությունը։ Պարզ է, որ հաշվումների ճշտությունը կմեծանա, եթե թվերը վերցվեն ավելի շատ նշանակալի թվանշաններով։ Հոդվածում առաջարկվում է նշանակալի թվանշանները պահելու համար օգտագործել մի քանի հաջորդող բջիջներ։

Օրինակ, եթե a *թիվը ունի* $m \cdot n$ *նշանակալի թվանշան,* $a = a_1 a_2 \cdots a_{mn}$, *որտեղ* $0 \leqslant a_i \leqslant 9$ $(1 < i \leqslant m.n)$, $a_1 \neq 0$, *և յուրաքանչյուր բջիջում կարելի է պահել* n *թվանշաններ ունեցող թիվ, ապա մենք* a *թվի* $a_1, a_2, \ldots, a_n$ *մասը կպահենք* $a+1$ *բջիջում,* a *թվի* $a_{n+1}, a_{n+2}, \ldots, a_{2n}$ *մասը կպահենք* $a+2$ *բջիջում,* a *թվի* $a_{(m-1)n+1}, \ldots, a_{mn}$ *մասը կպահենք* $a+m$ *բջիջում։*

Այսպիսով, մեր a *թիվը պահվեց* m *իրար հաջորդող բջիջներում։ Բայց, քանի որ ԷՀՄ-ում թվաբանական գործողությունները կատարվում են միայն մի բջիջում պահվող թվերի հետ, ուրեմն հարկավոր է այնպես անել, որ*

թվաբանական գործողությունները մեքենան կատարի մի քանի բջիջներում պահվող թվերի հետ։

Այս կերպ վարվելով մենք կարող ենք ցանկացած չափով մեծացնել հաշվումների ճշտությունը m-ը ավելի մեծ վերցնելու միջոցով։

Հոդվածում բերված է m բջիջներում պահվող թվերի հետ թվաբանական գործողություններ կատարելու կանոնները և նրանց լոգիկական սխեմաները։

ЛИТЕРАТУРА

1. *Войтишек В. В.* Программа счета с удвоенным числом разрядов. Сборник стандартных и типовых программ для БЭСМ, 1960.