Алгоритмы быстрого выполнения сложных операций в системе остаточных классов

ПОЛИССКИЙ Ю.Д.

НИИ автоматизации черной металлургии

Приведено решение задачи быстрого выполнения сложных операций в системе остаточных классов. Метод решения базируется на синтезе идей опорных модулей и парной нулевизации при табличной реализации операций

Приведено рішення завдання швидкого виконання складних операцій в системі залишкових класів. Метод рішення базується на синтезі ідей опорних модулів і парної нулевізації при табличній реалізації операцій.

Solving of the problem of quickly perform complex operations in the system of residual classes. The method of solution is based on a synthesis of ideas supporting modules and vanishing of the two modules in tabular implementation of operations.

Введение. Система остаточных классов (СОК) [1,2] обладает высокой степенью параллелизма, что позволяет существенно повысить эффективность вычислений в параллельных структурах. Однако возникают серьезные трудности при реализации сложных, так называемых немодульных, операций, для выполнения которых необходимо знание цифр операндов по всем разрядам. В связи с этим значительная часть работ по совершенствованию машинной арифметики СОК посвящена ускорению реализации сложных операций.

Постановка задачи. В работе [3] показана взаимосвязь немодульных операций, так что, получив решение одной из них, можно найти решения остальных.

Одним из важных достоинств СОК является малая разрядность операндов и результата операции, что позволяет применять табличные методы [4] с получением результата простой выборкой из таблиц.

Представление числа N в полиадическом коде с основаниями $m_1, m_2, \dots, m_n$ имеет вид

$$N = \pi_1 + \pi_2 m_1 + \pi_3 m_1 m_2 + \dots + \pi_i m_1 m_2 \dots m_{i-1} +$$

$$+ \dots + \pi_{n-1} m_1 m_2 \dots m_{n-2} + \pi_n m_1 m_2 \dots m_{n-1},$$

где π_i-позиционная характеристика i-го разряда, $\pi_i = 0,1,2,\dots,m_i - 1$, $i = 1,2,\dots,n$. Поскольку N задано не в полиадическом коде, а своими остатками по модулям $m_1, m_2, \dots, m_n$, определение значений π_i выполняется итеративным путем. С этой целью на t-ой итерации определяется константа

$$\Delta^j_{i,t} = (\pi^j m_1 m_2 \dots m_{i-1}) (\text{mod } m_i),$$

где $t = 1, 2, \dots, n$ - итерация, $i = t, t+1, \dots, n$ - индекс модуля системы, $j = 0,1,2,\dots, m_i - 1$ - строка состояния, $\pi^j = 0,1,2,\dots, m_i - 1$ - j - я позиционная характеристика, $\Delta^j_{i,1} = \pi^j$.

Назовем приведенным остатком $\tilde{\alpha}^j_{i,t}$ по модулю m_i на t–ой итерации остаток

$$\tilde{\alpha}^j_{i,t} = (\tilde{\alpha}^j_{i,t-1} - \Delta^j_{i,t}) \ (\text{mod } m_i).$$

В табл. 1-4 для системы модулей $m_1 = 11, m_2 = 5, m_3 = 7, m_4 = 3, m_5 = 2$ представлены константы для каждой из четырех итераций, вычисленные в соответствии с [2].

Таблица 1. **Константы первой итерации**

Модули				
11	**5**	**7**	**3**	**2**
Константы				
$\Delta_{1,1}$	$\Delta_{2,1}$	$\Delta_{3,1}$	$\Delta_{4,1}$	$\Delta_{5,1}$
0	0	0	0	0
1	1	1	1	1
2	2	2	2	0
3	3	3	0	1
4	4	4	1	0
…	**…**	**…**	**…**	**…**
8	3	1	2	0
9	4	2	0	1
10	0	3	1	0

Таблица 2. **Константы второй итерации**

Модули				
11	**5**	**7**	**3**	**2**
Константы				
$\Delta_{1,2}$	$\Delta_{2,2}$	$\Delta_{3,2}$	$\Delta_{4,2}$	$\Delta_{5,2}$
0	0	0	0	0
0	1	4	2	1
0	2	1	1	0
0	3	5	0	1
0	4	2	2	0

Таблица 3. **Константы третьей итерации**

Модули				
11	**5**	**7**	**3**	**2**
Константы				
$\Delta_{1,3}$	$\Delta_{2,3}$	$\Delta_{3,3}$	$\Delta_{4,3}$	$\Delta_{5,3}$
0	0	0	0	0
0	0	6	1	1
0	0	5	2	0
0	0	4	0	1
0	0	3	1	0
0	0	2	2	1
0	0	1	0	0

Таблица 4. **Константы четвертой итерации**

Модули				
11	**5**	**7**	**3**	**2**
Константы				
$\Delta_{1,4}$	$\Delta_{2,4}$	$\Delta_{3,4}$	$\Delta_{4,4}$	$\Delta_{5,4}$
0	0	0	0	0
0	0	0	1	1
0	0	0	2	0

Таблица 5. **Работа базового алгоритма**

	Модули				
	11	**5**	**7**	**3**	**2**
282	7	2	2	0	0
2	2	2	2	2	0
141	**9**	**1**	**1**	**0**	***0***
	9	4	2	0	1
Итерация 1	=	2	6	0	1
		2	1	1	0
Итерация 2		=	5	2	1
			5	2	0
Итерация 3			=	0	1
				0	0
Итерация 4				=	**1**

Табл. 5 иллюстрирует работу базового алгоритма на примере деления числа $N_1 = 282$ =(7, 2, 2, 0, 0) на N_2 =2= (2, 2, 2, 2, 0) в системе модулей $m_1 = 11$, $m_2 = 5$, $m_3 = 7$, $m_4 = 3$, $m_5 = 2$.

В результате формального деления N_1 на N_2 получаем число $N = (9,\ 1,\ 1,\ 0,\ \frac{0}{0})$.

Как видно из табл. 5, неопределенность вида $\frac{0}{0}$ для $m_5 = 2$ раскрывается по базовому алгоритму за 4 итерации.

Цель статьи. Описание алгоритмов ускоренного выполнения сложных операций в системе остаточных классов

Основная часть. В заданной системе модулей $m_1, m_2, \ldots, m_i, \ldots, m_n$ назовем модули с нечетными индексами основными, модули с четными индексами – опорными. Тогда табл. 1 и табл. 3 представляют константы основных модулей, табл. 2 и табл. 4 – константы опорных модулей. В данном алгоритме для ускорения получения результата предлагается на очередной итерации использовать одновременно наборы пар вычитаемых строк констант, каждая пара которых включает выбранную строку констант основного модуля m_{r-1} и $j-ю$, $j = 0,1,2,\ldots,m_r - 1$ строку констант опорного модуля m_r, т.е.

$$\tilde{\alpha}^{j}_{r,t} = (\tilde{\alpha}^{j_0}_{r,t-1} - \Delta^{j_0}_{r,t} - \Delta^{j}_{r,t}) \pmod{m_r},$$

$$j = 0,1,2,\ldots,m_r - 1$$

Та из пар, приведенный остаток по данному опорному модулю которой равен нулю, служит для выбора строки констант очередного, m_{r+1}-го модуля.

Таблица 6

	Модули				
	11	**5**	**7**	**3**	**2**
282	7	2	2	0	0
2	2	2	2	2	0
141	**9**	**1**	**1**	**0**	***0***
	9	4	2	0	1
		0	0	0	0
	=	2	6	0	1
	…	**…**	**…**	**…**	**…**
141	**9**	**1**	**1**	**0**	***0***
	9	4	2	0	1
		4	2	2	0
	=	3	4	1	1

Таким образом, для обработки пары модулей требуется всего один такт, что повышает быстродействие вдвое.

Работу предложенного алгоритма также рассмотрим на предыдущем примере.

Первая итерация иллюстрируется табл. 6.

На каждый из $m_2 = 5$ сумматоров записывается в качестве уменьшаемого строка значений остатков полученного частного, т.е. (9, 1, 1, 0, 0). При этом значение α_5 принимается равным нулю. По значению $\alpha_1 = 9$ из табл.1 выбирается строка констант (9, 4, 2, 0, 1) , которая в качестве первого вычитаемого также записывается на каждый из $m_2 = 5$ сумматоров. В качестве второго вычитаемого на $j-й$, $j=1,2,\ldots,m_2$ сумматор записывается $j-я$ строка констант табл. 2. На каждом из сумматоров выполняется операция вычитания, и для второй итерации выбирается строка значений α_i , для которой $\widetilde{\alpha}_2=0$.

Работа второй итерации иллюстрируется табл. 7.

Таблица 7. **Работа второй терации**

	Модули				
	11	**5**	**7**	**3**	**2**
141	=	=	**5**	**2**	**1**
			5	**2**	**0**
			0	**0**	**0**
	=	=	=	**0**	**1**
	…	…	…	…	…
141	=	=	**5**	**2**	**1**
			5	**2**	**0**
			0	**1**	**1**
	=	=	=	**2**	**0**
	…	…	…	…	…
141	=	=	**5**	**2**	**1**
			5	**2**	**0**
			0	**2**	**0**
	=	=	=	**1**	**1**

На каждый из $m_4=3$ сумматоров записывается в качестве уменьшаемого строка значений остатков полученного частного, т.е. (0, 0, 5, 2, 1). По значению $\alpha_3=5$ из табл. 3 выбирается строка констант (0, 0, 5, 2, 0) , которая в качестве первого вычитаемого также записывается на каждый из $m_4=3$ сумматоров. В качестве второго вычитаемого на $j-й$, $j=1,2,\ldots,m_4$ сумматор записывается $j-я$ строка констант табл. 4. На каждом из сумматоров выполняется операция вычитания, и для определения результата выбирается строка значений α_i , для которой $\widetilde{\alpha}_4=0$. Искомое значение $\alpha_5=1$. Таким образом, неопределенность вида $\frac{0}{0}$ для $m_5=2$ раскрывается по данному алгоритму вместо 4 за 2 итерации.

Быстродействие алгоритма можно еще более увеличить при синтезе идеи использования опорных модулей с идеей парной нулевизации [1]. При парной нулевизации для обработки каждой пары модулей m_1,m_2, $m_3,m_4,\ldots$, $m_{i-1},m_i\ldots$, . . . , m_{n-1},m_n требуется один такт. Поэтому при таком синтезе каждую нечетную пару модулей примем в качестве основной, каждую четную – в качестве опорной. В результате быстродействие базового алгоритма увеличивается в четыре раза.

Работу этого алгоритма также рассмотрим на предыдущем примере.

В табл. 8 представлены константы пары основных модулей, в табл. 9 – константы пары опорных модулей.

Таблица 8. **Константы основной пары модулей**

Модули						
11	**5**	**11**	**5**	**7**	**3**	**2**
$\alpha_{i-1,i}$		$\Delta_{i-1,i}$				
0	**0**	0	0	0	0	0
1	**1**	1	1	1	1	1
2	**2**	2	2	2	2	0
3	**3**	3	3	3	0	1
4	**4**	4	4	4	1	0
…	**…**	…	…	…	…	…
9	**4**	9	4	2	0	1
10	**0**	10	0	3	1	0
…	**…**	**…**	**…**	**…**		**…**
7	**1**	7	1	2	0	1
8	**2**	8	2	3	1	0
9	**3**	9	3	4	2	1
10	**4**	10	4	5	0	0

Таблица 9. **Константы опорной пары модулей**

Модули				
7	**3**	7	3	2
$\alpha_{i-1,i}$		$\Delta_{i-1,i}$		
0	**0**	0	0	0
1	**1**	6	1	1
2	**2**	5	2	0
3	**0**	4	0	1
…	**…**	…	….	….
2	**0**	5	0	1
3	**1**	4	1	0
…	**…**	**…**	**…**	**…**
5	**0**	2	0	0
6	**1**	1	1	1
…	**….**	….	….	….
4	**0**	3	0	0
5	**1**	2	1	1
6	**2**	1	2	0

Работа алгоритма иллюстрируется табл. 10.

Таким образом, в отличие от базового алгоритма, при котором результат достигается за четыре итерации, по данному алгоритму результат получен за одну итерацию.