УДК: 519.6

А.М. Терещенко

АНАЛІЗ СКЛАДНОСТІ ОПЕРАЦІЇ МНОЖЕННЯ БАГАТОРОЗРЯДНИХ ЧИСЕЛ ПРИ РЕАЛІЗАЦІЇ У ПАРАЛЕЛЬНІЙ МОДЕЛІ ОБЧИСЛЕНЬ

В даній роботі при обчисленні операції багаторозрядного множення чисел довжини N стандартним методом «у стовпчик» в паралельній моделі обчислень аналізується складність за кількістю операцій додавання та множення цілих однорозрядних чисел, виконаних одним паралельним процесором, для двох випадків, коли кількість паралельних процесорів необмежена, та коли кількість процесорів обмежена та кратна N.

Вступ

На даний час мікропроцесорна техніка розвивається дуже швидко. Якщо декілька десятиліть назад кластери з великою кількістю процесорів були доступні тільки потужним фінансовим організаціям для проведення надобчислень, то нині кластери з кількістю процесорів більше ніж 2000 можливі в домашніх умовах на персональних комп'ютерах на основі графічних прискорювачів. Поряд з подальшим розвитком алгоритмів у послідовній моделі обчислень [1–6], необхідних для енергоекономних нешвидких пристроїв, таких як смарткарти, є потреба в розробці нових методів ефективних у паралельній моделі обчислень. При розробці та вдосконаленні алгоритмів у паралельній моделі обчислень, дуже важливо мати методи для оцінки складності алгоритмів для порівняння з існуючими та знаходження ефективних діапазонів їх використання.

При розробці алгоритмів багаторозрядного множення в паралельній моделі обчислень необхідно завжди мати на увазі наступні пункти, які мають значний вплив на загальний час виконання:

– кількість доступних процесорів;

– кількість операцій (ітерацій), виконуваних кожним з паралельних процесорів;

– тип операцій (цілі або комплексні числа, числа з плаваючою комою і т. д.);

– врахування знаку переносу;

– кількість тісно пов'язаних кроків;

– обсяг використовуваної пам'яті.

Даний перелік не є вичерпним. В даній роботі основна увага приділяється першим двом пунктам.

Перед тим, як розглядати більш складні алгоритми, розглянемо спочатку самий простий алгоритм, який знаходить суму цілих чисел у паралельній моделі обчислень.

Алгоритм 1. Знаходження суми N цілих чисел $r = \sum_{i=0}^{N-1} x_i$ у паралельній моделі обчислень при задіяні P паралельних процесорів ($P \ge \lfloor N/2 \rfloor$) (без урахування знаків переносу).

Вхід: N, x_i, $i = \overline{0, N-1}$.

Вихід: r.

1. $n \leftarrow \lceil \log_2 N \rceil$.
2. $r_i \leftarrow 0$, $i = \overline{0, 2^n - 1}$. // Ініціалізація.
3. $r_i \leftarrow x_i$, $i = \overline{0, N-1}$. // Завантаження.
4. Для j від 1 до n // Кількість ітерацій.
5. $I \leftarrow 2^{n-j}$.
6. Для i від 0 до $I-1$ // Одночасне виконання на I процесорах.
7. $P_i(r_i \leftarrow r_i + r_{i+I})$. // Виконується паралельно кожним з P_i процесорів.
8. Кінець по i.
9. Кінець по j.

Після виконання алгоритму r_0 буде мати результат додавання цілих чисел.

Розглянемо на прикладі $N = 7$.

$P_i(r_i + \leftarrow r_k)$ позначає, що P_i процесор при паралельному виконанні додає

значення з комірки r_k до значення, яке зберігається в комірці r_i.

З табл. 1 видно, що кожний з чотирьох процесорів виконає не більше трьох операцій додавання ($\lceil \log_2 7 \rceil = 3$).

Схематично кількість комірок для додавання на кожній ітерації показана на рис. 1, де значення комірок білого кольору додаються до комірок сірого кольору на кожній ітерації $j = 1, 2, 3$.

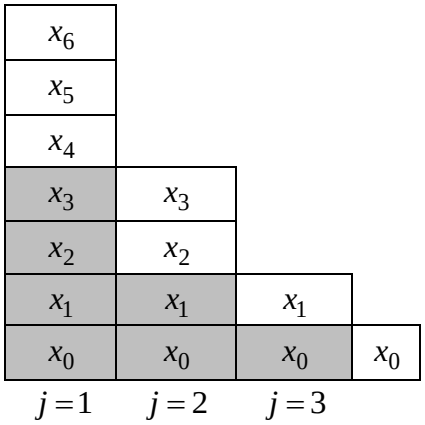

Рис. 1. Ітераційне додавання значень при сумуванні $N = 7$

Алгоритм 1 не є оптимальним, бо операція $P_3(r_3 + \leftarrow r_7)$ не є обов'язковою для випадку $N = 7$. Алгоритм 1 є загальним для випадків $5 \le N \le 8$ і основна мета його опису – це знаходження найбільшої кількості операцій додавань, виконаних одним з процесорів. З Алгоритму 1 видно, що процесор P_0 завжди задіяний.

Лема 1. При виконанні Алгоритму 1 на кожній ітерації j кількість задіяних паралельних процесорів дорівнює $\left\lfloor \left\lceil \frac{N}{2^j} \right\rceil \Big/ 2 \right\rfloor$, та залишається підсумувати $\left\lceil \frac{N}{2^j} \right\rceil$ комірок після виконання кожної ітерації j.

Доведення. Розглядаючи додавання двох чисел x_0 та x_1 ($N = 2$) досить задіяти один процесор. При додаванні трьох чисел x_0, x_1 та x_2 ($N = 3$) другий процесор буде чекати поки одне із значень x_1 або x_2 буде додане до x_0, щоб додати інше, тобто другий процесор буде зайвий. Так само при отримані суми з п'яти чисел x_0, x_1, x_3, x_4 та x_5 ($N = 5$) не має сенсу задіяти більше двох процесорів. Можна стверджувати, що максимальна кількість процесорів буде задіяна на першій ітерації $j = 1$, коли $\lfloor N/2 \rfloor$ значень додається до значень в $\lfloor N/2 \rfloor$ комірках. Зрозуміло, що необхідно задіяти $\lfloor N/2 \rfloor$ процесорів для виконання всіх додавань одночасно. Кількість комірок, які залишаються та які необхідно доопрацювати, дорівнює

$$N - \lfloor N/2 \rfloor = \lceil N/2 \rceil, \text{ при } N \ge 2. \quad (1)$$

При $j = 2$ значення з $\left\lfloor \left\lceil \frac{N}{2} \right\rceil \Big/ 2 \right\rfloor$ комірок будуть додані до $\left\lfloor \left\lceil \frac{N}{2} \right\rceil \Big/ 2 \right\rfloor$ комірок

Таблиця 1. Покрокове виконання Алгоритму 1 при $N = 7$

	$n \leftarrow \lceil \log_2 N \rceil = 3;\ r_i,\ i = \overline{0,2^n - 1} = \overline{0,7}$							
Крок	r_0	r_1	r_2	r_3	r_4	r_5	r_6	r_7
2	0	0	0	0	0	0	0	0
3	x_0	x_1	x_2	x_3	x_4	x_5	x_6	0
7	$P_0(r_0 + \leftarrow r_4)$	$P_1(r_1 + \leftarrow r_5)$	$P_2(r_2 + \leftarrow r_6)$	$P_3(r_3 + \leftarrow r_7)$				
7	$P_0(r_0 + \leftarrow r_2)$	$P_1(r_1 + \leftarrow r_3)$						
7	$P_0(r_0 + \leftarrow r_1)$							

при задіянні $\left\lfloor \left\lceil \frac{N}{2} \right\rceil / 2 \right\rfloor$ паралельних процесорів для виконання цієї ітерації за один крок, де $\lceil N/2 \rceil$ – кількість комірок, які залишилися після виконання першої ітерації. Після завершення ітерації $j=2$ маємо кількість значень, які залишилося додати, буде $\lceil N/2 \rceil - \left\lfloor \left\lceil \frac{N}{2} \right\rceil / 2 \right\rfloor$. Згідно формули (1) можна записати.

$$\lceil N/2 \rceil - \left\lfloor \left\lceil \frac{N}{2} \right\rceil / 2 \right\rfloor = \left\lceil \left\lceil \frac{N}{2} \right\rceil / 2 \right\rceil, \text{ при } N \geq 4.$$

Попередній вираз можна спростити.

$$\lceil N/2 \rceil - \left\lfloor \left\lceil \frac{N}{2} \right\rceil / 2 \right\rfloor = \lceil N/4 \rceil, \text{ при } N \geq 4.$$

Аналогічно отримуємо, що після ітерації $j=3$ залишається $\lceil N/8 \rceil$ комірок.

$$\lceil N/4 \rceil - \left\lfloor \left\lceil \frac{N}{4} \right\rceil / 2 \right\rfloor = \lceil N/8 \rceil, \text{ при } N \geq 8.$$

При виконанні останньої ітерації j кількість комірок, які залишаються для додавання, виражається.

$$\lceil N/2^{j-1} \rceil - \left\lfloor \left\lceil \frac{N}{2^j} \right\rceil / 2 \right\rfloor = \lceil N/2^j \rceil, \text{ при } N \geq 2^j.$$

З попереднього виразу видно, що кількість задіяних процесорів (або комірок, які додаються) дорівнює $\left\lfloor \left\lceil \frac{N}{2^j} \right\rceil / 2 \right\rfloor$ на ітерації j. Лема доведена.

Лема 2. При виконанні Алгоритму 1 кожний з паралельних процесорів P виконає не більше $\lceil \log_2 N \rceil$ операцій додавання N цілих чисел при $P \geq \lfloor N/2 \rfloor$.

Доведення. Згідно умови леми можемо вважати, що кількість паралельних процесорів достатня на кожному кроці при $P \geq \lfloor N/2 \rfloor$. Згідно Леми 1 кількість комірок, які залишаються після виконання ітерації j, дорівнює $\left\lceil \frac{N}{2^j} \right\rceil$. Після виконання Алгоритму 1 сума всіх значень записується в одну комірку $\left\lceil \frac{N}{2^j} \right\rceil = 1$, тобто кількість ітерацій дорівнює $j = \lceil \log_2 N \rceil$. Лема доведена.

Наступна лема розглядає загальний випадок, коли кількість задіяних процесорів не завжди достатня для виконання всіх додавань одночасно на кожній ітерації.

Лема 3. При виконанні Алгоритму 1 кожний з паралельних процесорів P виконає не більше

$$\sum_{j=0}^{\lfloor \log_2 N \rfloor} \left\lceil \frac{\lceil N/2^j \rceil - \lceil N/2^{j+1} \rceil}{P} \right\rceil$$

операцій додавання цілих чисел при $P \leq \lfloor N/2 \rfloor$.

Доведення. При виконані Алгоритму 1 кількість задіяних процесорів зменшується і дорівнює одному задіяному на останній ітерації. На кожній ітерації $j = 1, 2, \ldots$ кількість задіяних процесорів дорівнює $\left\lfloor \left\lceil \frac{N}{2^j} \right\rceil / 2 \right\rfloor$ (або $\left\lceil \frac{N}{2^{j-1}} \right\rceil - \left\lceil \frac{N}{2^j} \right\rceil$), для одночасного виконання всіх додавань, як було показано в Лемі 1. Якщо кількість процесорів P менше $\left\lceil \frac{N}{2^{j-1}} \right\rceil - \left\lceil \frac{N}{2^j} \right\rceil$ на ітерації j, то кожен з паралельних процесорів виконає не більше $\left\lceil \frac{\lceil N/2^{j-1} \rceil - \lceil N/2^j \rceil}{P} \right\rceil$ операцій додавання на ітерації j, при загальній кількості ітерацій $\lceil \log_2 N \rceil$. Лема доведена.

Примітка 1. Робиться припущення, що наступна ітерація $j+1$ не починається поки всі значення на ітерації j не додані. Інакше можна вважати, що Лема 2 може бути використана для знаходження нижньої обчислювальної оцінки.

Примітка 2. Основна увага приділяється знаходженню величини складності, де можна стверджувати, що будь-який задіяний паралельний процесор виконає кількість операцій «не більше» визначеного значення.

Алгоритм 2. Множення двох N-розрядних чисел $R_{2N} = X_N \cdot Y_N$ стандартним методом «у стовпчик» у паралельній моделі обчислень (без врахування знаків переносу).

1. Поелементне множення розрядів чисел $z_{i,j} \leftarrow x_i \cdot y_j$, $i, j = \overline{0, N-1}$.

2. Знаходження розрядів результату множення $r_k \leftarrow \sum_{k=i+j} z_{i,j}$, $k = \overline{0,2N-2}$.

Розглянемо виконання алгоритму на прикладі множення двох 2-розрядних чисел ($X = x_1 \cdot 2^{\omega} + x_0$, $Y = y_1 \cdot 2^{\omega} + y_0$, $N = 2$, $P = 4$). Обчислення кожного розряду результату можна представити наступним чином (рис. 2):

$$x_i \cdot y_j = h_{i,j} \cdot 2^{\omega} + l_{i,j},$$

де $l_{i,j} = L(x_i \cdot y_j)$, $h_{i,j} = H(x_i \cdot y_j)$

$(l - Low,\ h - High)$

$$\begin{array}{cccc} & & y_1 & y_0 \\ & & x_1 & x_0 \\ \hline & & h_{0,0} & l_{0,0} \\ & h_{0,1} & l_{0,1} & \\ & h_{1,0} & l_{1,0} & \\ h_{1,1} & l_{1,1} & & \\ \hline r_3 & r_2 & r_1 & r_0 \end{array}$$

Рис. 2. Множення двох 2-розрядних чисел на основі Алгоритму 2

З рис. 2 видно, що кожний старший розряд результату однорозрядних множень додається до наступного розряду результату множення.

Обчислення r_i, $i = \overline{0,3}$, можна виразити наступним чином:

$$r_0 = l_{0,0},$$

$$r_1 = (l_{0,1} + l_{1,0}) + h_{0,0},$$

$$r_2 = (h_{0,1} + h_{1,0}) + l_{1,1},$$

$$r_3 = h_{1,1}.$$

Для обчислення r_1 та r_2 виконується послідовно дві операції додавання при задіяні двох процесорів. При задіяні чотирьох паралельних процесорів кожний з них виконає наступну кількість операцій (табл. 2).

З табл. 2 видно, що кожний з P_i процесорів виконає не більше трьох операцій над цілими однорозрядними числами.

Таблиця 2. Кількість операцій над цілими числами при множенні двох 2-розрядних чисел, виконаних кожним процесором $P_i = \overline{0,3}$

P_i	Множення	Додавання	Кількість операцій
P_0	$x_0 \cdot y_0$		1
P_1	$x_0 \cdot y_1$	$(l_{0,1} + l_{1,0}) + h_{0,0}$	3
P_2	$x_1 \cdot y_0$	$(h_{0,1} + h_{1,0}) + l_{1,1}$	3
P_3	$x_1 \cdot y_1$		1

Лема 4. Обчислення результату множення двох N-розрядних чисел

$$R_{2N} = X_N \cdot Y_N = \sum_{k=0}^{2N-1} r_k \cdot 2^{\omega k},$$

$$r_i = \sum_{k=i+j} (x_i \cdot y_j),$$

може бути представлено у наступному вигляді:

$$r_0 = l_{0,0};\ r_k = \sum_{k-1=i+j} h_{i,j} + \sum_{k=i+j} l_{i,j},\ k = \overline{1,2N-2};$$

$$r_{2N-1} = h_{N-1,N-1},$$

де $l_{i,j} = L(x_i \cdot y_j)$, $h_{i,j} = H(x_i \cdot y_j)$.

Доведення. Розглянемо на прикладі множенні двох 4-розрядних чисел.

P_7	P_6	P_5	P_4	P_3	P_2	P_1	P_0
$h_{3,3}$	$l_{3,3}$					$h_{0,0}$	$l_{0,0}$
	$h_{2,3}$	$l_{2,3}$			$h_{0,1}$	$l_{0,1}$	
	$h_{3,2}$	$l_{3,2}$			$h_{1,0}$	$l_{1,0}$	
		$h_{1,3}$	$l_{1,3}$	$h_{0,2}$	$l_{0,2}$		
		$h_{2,2}$	$l_{2,2}$	$h_{1,1}$	$l_{1,1}$		
		$h_{3,1}$	$l_{3,1}$	$h_{2,0}$	$l_{2,0}$		
			$h_{0,3}$	$l_{0,3}$			
			$h_{1,2}$	$l_{1,2}$			
			$h_{2,1}$	$l_{2,1}$			
			$h_{3,0}$	$l_{3,0}$			
r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0

Рис. 3. Обчислення r_i, $i = \overline{0,7}$

Обчислення r_i, $i = \overline{0,7}$, може виконати при задіяні P_i, $i = \overline{0,7}$, процесорів, кожний з яких знайде суму всіх елементів у своїй колонці, як показано на рис. 3.

На основі рис. 3 обчислення r_i, $i = \overline{0,7}$, можна представити наступним чином:

$$r_0 = l_{0,0},$$

$$r_1 = h_{0,0} + l_{0,1} + l_{1,0},$$

$$r_2 = h_{0,1} + h_{,0,1} + l_{0,2} + l_{1,1} + l_{2,0},$$

$$r_3 = h_{0,2} + h_{1,1} + h_{2,0} + l_{0,3} + l_{1,2} + l_{2,1} + l_{3,0},$$

$$r_4 = h_{0,3} + h_{1,2} + h_{2,1} + h_{3,0} + l_{1,3} + l_{2,2} + l_{3,1},$$

$$r_5 = h_{1,3} + h_{2,2} + h_{3,1} + l_{2,3} + l_{3,2},$$

$$r_6 = h_{2,3} + h_{3,2} + l_{3,3},$$

$$r_7 = h_{3,3}.$$

З використанням знаків сум наступні вирази можна записати наступним чином:

$$r_0 = l_{0,0},$$

$$r_1 = \sum_{0=i+j} h_{i,j} + \sum_{1=i+j} l_{i,j},\ r_2 = \sum_{1=i+j} h_{i,j} + \sum_{2=i+j} l_{i,j},$$

$$r_3 = \sum_{2=i+j} h_{i,j} + \sum_{3=i+j} l_{i,j},\ r_4 = \sum_{3=i+j} h_{i,j} + \sum_{4=i+j} l_{i,j},$$

$$r_5 = \sum_{4=i+j} h_{i,j} + \sum_{5=i+j} l_{i,j},\ r_6 = \sum_{5=i+j} h_{i,j} + \sum_{6=i+j} l_{i,j},$$

$$r_7 = h_{3,3}.$$

Розглядаючи випадок для N, отримуємо необхідні вирази. Лема доведена.

Примітка 3. Так як кількість однорозрядних доданків $l_{i,j}$ та $h_{i,j}$ дорівнює по N^2, то зрозуміло, що, загалом, необхідно опрацювати $2N^2$ однорозрядних значень при додаванні.

Примітка 4. Найбільшу кількість доданків $2N-1$ необхідно опрацювати для комірок результату r_N та r_{N+1}.

В паралельній моделі обчислення операцій множення та додавання відрізняються тим, що всі операції множення можуть бути обчисленні одночасно при задіяні N^2 процесорів, а при додаванні це неможливе, так як технічно можна додавати тільки одне значення до однієї комірки пам'яті за одну операцію.

На основі попередньої леми 3 Алгоритм 2 може бути покращений за рахунок виконання Алгоритму 1 кожним з паралельних потоків.

Алгоритм 3. Множення двох N-розрядних чисел $R_{2N} = X_N \cdot Y_N$ стандартним методом «у стовпчик» у паралельній моделі обчислень (без врахування знаків переносу).

1. $h_{i,j} \leftarrow H(x_i \cdot y_j)$, $l_{i,j} \leftarrow L(x_i \cdot y_j)$, $i, j = \overline{0, N-1}$. // Поелементне множення.

2. $r_0 \leftarrow l_{0,0}$, $r_{2N-1} \leftarrow h_{N-1,N-1}$. // Два паралельних потоки обчислення.

3. Для i від 1 до $N-1$ // $N-1$ паралельних потоків обчислення.

4. $r_i \leftarrow Алгоритм\_1(i+1, \{l_{j,i-j},\ j = \overline{0, i}\})$. // Додавання l-розрядів.

5. $r_i \leftarrow r_i + Алгоритм\_1(i, \{h_{j,i-j-1}, j=\overline{0,i-1}\})$. // Додавання h-розрядів.

6. Кінець по i.

7. Для i від 1 до $N-1$ // $N-1$ паралельних потоків обчислення.

8. $r_i \leftarrow Алгоритм\_1(N-i, \{l_{j,N+i-j-1}, j=\overline{i,N-1}\})$. // Додавання l-розрядів.

9. $r_i \leftarrow r_i + Алгоритм\_1(N-i+1, \{h_{j,N+i-j-1}, j=\overline{i-1,N-1}\})$. // Додавання h-розрядів.

10. Кінець по i.

Аналізуючи рис. 3, бачимо, що для представлення старших та молодших розрядів ($l_{i,j}$ та $h_{i,j}$, $i,j=\overline{0,3}$) необхідно використовувати матрицю 10×8. Наступна лема 5 доводить, що матриця розміром 7×8 достатня для розміщення всіх елементів, що більш ефективне при додаванні.

Лема 5. При виконані Алгоритму 2 кількість однорозрядних доданків для обчислення кожного розряду результату множення r_i, $i=\overline{0,2N-1}$, не перевищує $2N-1$.

Доведення. Розглянемо на прикладі $N=4$. Далі в табл. 3 на основі рис. 3 наведена кількість операцій додавання, необхідних для обчислення кожного з тимчасових результатів r_i, $i=\overline{0,7}$, в паралельній моделі.

З табл. 3 видно, що кількість доданків $l_{i,j}$ збільшується до $i=N-1$ починаючи з 1, а потім зменшується до нуля, та, що кількість доданків $h_{i,j}$ збільшується до $i=N$ починаючи з 0, а потім зменшується до 1. Загальна кількість доданків може бути виражена наступним чином у загальному випадку для будь-якого N:

$$O_0=1,\ O_{2N-1}=1.$$

$$O_i=2i+1,\ O_{N+i-1}=2N-2i+1,$$

$$i=\overline{1,N-1}.$$

Аналізуючи попередні співвідношення видно, що $O_i \le 2N-1$. Лема доведена.

Лема 6. При виконанні Алгоритму 2 загальна кількість однорозрядних додавань дорівнює $2N^2-2N$.

Доведення. Розглянемо на прикладі $N=4$. На першій ітерації можливо додати 12 доданків (білого кольору) до значень 12 комірок (світло сірого кольору) при задіяні 12 процесорів (рис. 4).

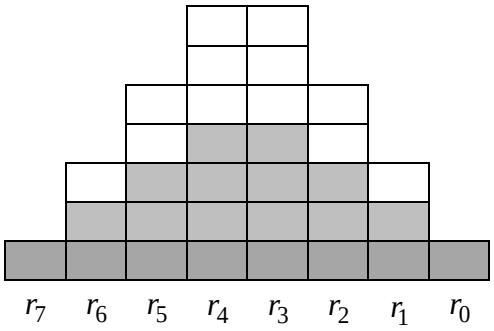

Рис. 4. Додавання значень на 1-й ітерації при множенні двох 4-разрядних чисел

Таблиця 3. Кількість доданків для обчислення кожного розряду результату r_i, $i=\overline{0,7}$, при $N=4$

r_i	r_0	r_1	r_2	r_3	r_4	r_5	r_6	r_7
Доданків $l_{i,j}$	1	2	3	4	3	2	1	0
Доданків $h_{i,j}$	0	1	2	3	4	3	2	1
Доданків	1	3	5	7	7	5	3	1
Ітерацій	0	2	3	3	3	3	2	0

На другій ітерації можливо додати 8 доданків (білого кольору) до значень 8 комірок (сірого кольору) при задіяні 8 процесорів (рис. 5).

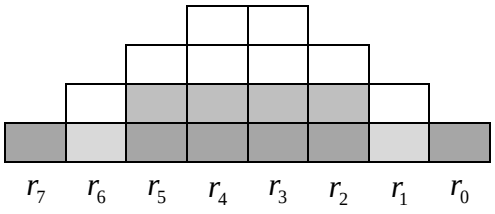

Рис. 5. Додавання значень на 2-й ітерації при множенні двох 4-разрядних чисел

На останній ітерації можливо додати 4 доданків (білого кольору) до значень 4 комірок (світло сірого кольору) при задіяні 4 процесорів (рис. 6).

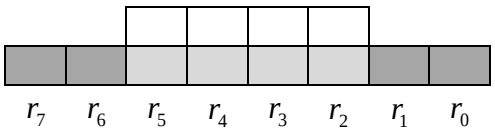

Рис. 6. Додавання значень на 3-й ітерації при множенні двох 4-разрядних чисел

Загалом необхідно виконати 24=12+ +8+4 операції додавання з 32 однорозрядних значень, бо 8 комірок залишається під результат. 32 значення отримано при обчисленні 16 однорозрядних множень, що дають 16 дворозрядних чисел при множення двох 4-розрядних чисел. Бачимо, що загальна кількість додавань буде $2\cdot(4\cdot 4)-2\cdot 4=24$, тобто $2N^2-2N$. Лема доведена.

Примітка 5. Попередня лема показує, що всі операції додавання неможливо виконати одночасно за одну операцію технічно, навіть якщо кількість паралельних процесорів необмежена.

Рис. 3 можна представити аналогічно рис. 4 після заміни значень l, h значеннями t де $t_{i,j}(h_{a,b})$, позначає, що значення $h_{a,b}$ знаходиться в комірці $t_{i,j}$ на рис. 7.

Наданий далі Алгоритм 4 дозволяє працювати з усіма значеннями без розділення на молодші та старші розряди. Використовуючи рис. 7, побудований для випадку $N=4$, можна перевірити коректність кроків 3, 4, 5, 6 Алгоритму 4.

Алгоритм 4. Множення двох N-розрядних чисел $R_{2N}=X_N\cdot Y_N$ стандартним методом «у стовпчик» у паралельній моделі обчислень (без врахування знаків переносу) при необмеженій кількості процесорів.

1. $h_{i,j}\leftarrow H(x_i\cdot y_j)$, $\quad l_{i,j}\leftarrow L(x_i\cdot y_j)$, $i,j=\overline{0,N-1}$. // Поелементне множення.

2. $r_0\leftarrow l_{0,0}$, $r_{2N-1}\leftarrow h_{N-1,N-1}$. // Два паралельних потоки обчислення.

3. Для i від 1 до $N-1$. // Індексація.

4. $t_{i,j}\leftarrow l_{j,i-j}$, $t_{2N-1-i,i+j}\leftarrow h_{N-1-i+j,N-1-j}$, $j=\overline{0,i}$.

P_7	P_6	P_5	P_4	P_3	P_2	P_1	P_0
			$t_{4,6}(h_{3,0})$	$t_{3,6}(h_{2,0})$			
			$t_{4,5}(h_{2,1})$	$t_{3,5}(h_{1,1})$			
		$t_{5,4}(h_{3,1})$	$t_{4,4}(h_{1,2})$	$t_{3,4}(h_{0,2})$	$t_{2,4}(h_{1,0})$		
		$t_{5,3}(h_{2,2})$	$t_{4,3}(h_{0,3})$	$t_{3,3}(l_{3,0})$	$t_{2,3}(h_{0,1})$		
	$t_{6,2}(h_{3,2})$	$t_{5,2}(h_{1,3})$	$t_{4,2}(l_{3,1})$	$t_{3,2}(l_{2,1})$	$t_{2,2}(l_{2,0})$	$t_{1,2}(h_{0,0})$	
	$t_{6,1}(h_{2,3})$	$t_{5,1}(l_{3,2})$	$t_{4,1}(l_{2,2})$	$t_{3,1}(l_{1,2})$	$t_{2,1}(l_{1,1})$	$t_{1,1}(l_{1,0})$	
$t_{7,0}(h_{3,3})$	$t_{6,0}(l_{3,3})$	$t_{5,0}(l_{2,3})$	$t_{4,0}(l_{1,3})$	$t_{3,0}(l_{0,3})$	$t_{2,0}(l_{0,2})$	$t_{1,0}(l_{0,1})$	$t_{0,0}(l_{0,0})$
r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0

Рис. 7. Заміна значень l, h значеннями t

5. $t_{i,i+1+j} \leftarrow h_{j,i-1-j}$, $t_{2N-1-i,j} \leftarrow l_{N-i+j,N-1-j}$, $j = \overline{0,i-1}$.

6. Кінець по i.

7. $n \leftarrow \lceil \log_2(2N-1) \rceil$.

8. Для j від 1 до n // Ітерації.

9. Для i від 1 до $N-1$

10. $T \leftarrow \left\lceil \frac{2i-1}{2^{j-1}} \right\rceil$, $M \leftarrow \left\lceil \frac{2i-1}{2^{j}} \right\rceil$, $I \leftarrow 2N-1-i$.

11. $t_{i,k} \leftarrow t_{i,k} + t_{i,M+k}$, $k = \overline{0,T-M}$. // Паралельне виконання в $T-M$ потоках.

12. $t_{I,k} \leftarrow t_{I,k} + t_{I,M+k}$, $k = \overline{0,T-M}$. // Паралельне виконання в $T-M$ потоках.

13. Кінець по i.

14. Кінець по j.

Алгоритм 4 розрахований на випадок, коли кількість процесорів $P \geq N^2$ достатня на кожному кроці алгоритму. Наступний алгоритм враховує кількість доступних процесорів при додаванні.

Алгоритм 5. Множення двох N-розрядних чисел $R_{2N} = X_N \cdot Y_N$ стандартним методом «у стовпчик» у паралельній моделі обчислень (без врахування знаків переносу) при задіяні P процесорів ($P \leq N^2$).

1. $h_{i,j} \leftarrow H(x_i \cdot y_j)$, $l_{i,j} \leftarrow L(x_i \cdot y_j)$, $i, j = \overline{0,N-1}$. // Поелементне множення.

2. $r_0 \leftarrow l_{0,0}$, $r_{2N-1} \leftarrow h_{N-1,N-1}$. // Два паралельних потоки обчислення.

3. Для i від 1 до $N-1$ // Індексація.

4. $t_{i,j} \leftarrow l_{j,i-j}$, $t_{2N-1-i,i+j} \leftarrow h_{N-1-i+j,N-1-j}$, $j = \overline{0,i}$.

5. $t_{i,i+1+j} \leftarrow h_{j,i-1-j}$, $t_{2N-1-i,j} \leftarrow l_{N-i+j,N-1-j}$, $j = \overline{0,i-1}$.

6. Кінець по i.

7. $n \leftarrow \lceil \log_2(2N-1) \rceil$.

8. Для j від 1 до n // Ітерацій.

9. $t_{i,kP+m} \leftarrow t_{i,kP+m} + t_{i,M+kP+m}$,

$t_{I,kP+m} \leftarrow t_{I,kP+m} + t_{I,M+kP+m}$,

$m = \overline{0, \begin{cases} k = K, & D-kP-1 \\ k < K, & P-1 \end{cases}}$, $k = \overline{0,K}$,

$K \leftarrow \left\lfloor \frac{D}{P} \right\rfloor$, $D \leftarrow \left\lceil \frac{2i-1}{2^{j-1}} \right\rceil - M$,

$M \leftarrow \left\lceil \frac{2i-1}{2^{j}} \right\rceil$, $I \leftarrow 2N-1-i$, $i = \overline{1,N-1}$.

10. Кінець по j.

Враховуючи всі попередні леми, наступна лема надає можливість знайти кількість операцій, не більше якої виконає кожен паралельній процесор, при обчисленні результату багаторозрядного множення на основі методу «у стовпчик» у паралельній моделі обчислень.

Лема 7. При виконанні Алгоритму 5 при задіяні P паралельних процесорів, кожний з процесорів виконає операцій з цілими числами не більше:

$$O^{*,+}(N,P) = O^{*}(N,P) + O^{+}(N,P). \quad (2)$$

$$O^{*}(N,P) = \left\lceil \frac{N^2}{P} \right\rceil,$$

$$O^{+}(N,P) = \sum_{j=0}^{\lfloor \log_2(2N-1) \rfloor} \left\lceil \frac{2\sum_{i=0}^{N-1} \left(\left\lceil \frac{2i+1}{2^j} \right\rceil - \left\lceil \frac{2i+1}{2^{j+1}} \right\rceil \right)}{P} \right\rceil.$$

Доведення. Розглянемо спочатку випадок, коли $P \leq N^2$. При обмеженій кількості задіяних процесорів число ітерацій для знаходження всіх однорозрядних множень виражається як

$$O^{*}(N,P) = \left\lceil \frac{N^2}{P} \right\rceil.$$

Зрозуміло, що кількість ітерацій, необхідних для знаходження всіх сум r_i, $i=\overline{0,\ 2N-1}$, буде залежати від самої довгої послідовності. З рис. 4 бачимо, що присутня симетрія, де ліва половина схеми обчислення дзеркально відповідає правій половині, так як для будь-якого N довжина результату множення завжди є парною. Для ітерації $j=1$ обчислимо загальну кількість значень (білі комірки на рис. 1), які додаються (до світло сірих комірок). Враховуючи, що кількість доданків для обчислення r_i, $i=\overline{0,N-1}$, виражається $A_i=2i+1$ (див. рис. 4, Лема 6), загальна кількість може бути виражена

$$A_{j=1}=2\sum_{i=0}^{N-1}\left(2i+1-\left\lceil\frac{2i+1}{2}\right\rceil\right).$$

В попередньому виразі множник 2 показує, що присутня симетрія, що дає можливість спрощення.

$$A_{j=1}=2\sum_{i=0}^{N-1}i\,.$$

Кількість задіяних процесорів дорівнює кількості значень, які додаються. Тобто кожен з процесорів здійснює одну операцію додавання.

Для ітерації $j=2$ кількість значень, які додаються, може бути представлена наступним чином.

$$A_{j=2}=2\sum_{i=0}^{N-1}\left(\left\lceil\frac{2i+1}{2}\right\rceil-\left\lceil\frac{2i+1}{4}\right\rceil\right).$$

Узагальнюючи попередній вираз для ітерації j, отримуємо.

$$A_j=2\sum_{i=0}^{N-1}\left(\left\lceil\frac{2i+1}{2^{j-1}}\right\rceil-\left\lceil\frac{2i+1}{2^{j}}\right\rceil\right).$$

Якщо кількість процесорів менша кількості значень, які додаються, то кількість операцій додавань, виконаних кожним з процесорів може бути представлена.

$$S_j=\left\lceil 2\sum_{i=0}^{N-1}\left(\left\lceil\frac{2i+1}{2^{j-1}}\right\rceil-\left\lceil\frac{2i+1}{2^{j}}\right\rceil\right)\Big/P\right\rceil.$$

Тоді загальна кількість додавань, виконана кожним з процесорів, на всіх ітераціях $\lceil\log_2(2N-1)\rceil$ не буде більша наступного значення.

$$O^+(N,P)=$$
$$=\sum_{j=1}^{\lceil\log_2(2N-1)\rceil}\left\lceil\frac{2\sum_{i=0}^{N-1}\left(\left\lceil\frac{2i+1}{2^{j-1}}\right\rceil-\left\lceil\frac{2i+1}{2^{j}}\right\rceil\right)}{P}\right\rceil.$$

Враховуючи кількість операцій множення кожним з процесорів, отримуємо необхідне співвідношення. Лема доведена.

Примітка 6. При достатній кількості паралельних процесорів $N^2\le P$, формула (2) приймає вигляд $1+\lceil\log_2 2N-1\rceil$, що показує кількість операцій, виконану кожним з паралельних процесорів, при виконані Алгоритму 1.

На основі формули (2) можна побудувати табл. 4, на основі якої отримуємо графік залежності кількості операцій від P, N (рис. 8).

Найбільш цікавою буде залежність кількості операцій від кількості доступних процесорів, яка кратна N (див. табл. 5).

З рис. 9 та табл. 5 видно, що графіки $P=64$ та $P=N/4$, $P=128$ та $P=N/2$, $P=256$ та $P=N$, $P=512$ та $P=2N$, $P=1024$ та $P=4N$, $P=2048$ та $P=8N$ сходяться в однакових точках при $N=256$. Графік $P=N/8$ співпадав би з графіком $P=32$, якщо би він був присутній на рис. 8.

Аналізуючи колонки $N/8$, $N/4$. $N/2$. N, $2N$, $4N$, $8N$ для кожного N табл. 5 можна зробити висновок, залежність між кількістю операцій, виконаних одним процесором, та кількістю таких процесорів, є дуже близькою до лінійної. Тобто, збільшуючи в $K=2^i$, i – ціле, разів кількість задіяних процесорів можна стверджувати, що кількість операцій, виконаних кожним з процесорів, зменшиться приблизно в K разів при виконанні Алгоритму 5.

Таблиця 4. Кількість операцій (додавання та множення) над цілими однорозрядними числами, не більше якої виконає кожен з паралельних процесорів P, при виконанні Алгоритму 5 для $2 \le N \le 32$

N	P					
	32	64	128	256	512	1024
2	3	3	3	3	3	3
3	4	4	4	4	4	4
4	4	4	4	4	4	4
5	5	5	5	5	5	5
6	6	5	5	5	5	5
7	7	5	5	5	5	5
8	7	5	5	5	5	5
9	11	8	6	6	6	6
10	12	8	6	6	6	6
11	13	8	6	6	6	6
12	17	11	8	6	6	6
13	18	11	8	6	6	6
14	21	12	8	6	6	6
15	23	13	8	6	6	6
16	24	13	8	6	6	6
17	31	18	12	9	7	7
18	34	19	12	9	7	7
19	36	20	12	9	7	7
20	40	22	13	9	7	7
21	43	23	14	9	7	7
22	47	25	14	9	7	7
23	52	29	17	11	8	7
24	55	30	18	12	9	7
25	60	32	18	12	9	7
26	66	35	20	12	9	7
27	69	36	20	12	9	7
28	75	40	22	13	9	7
29	81	42	23	14	9	7
30	86	45	24	14	9	7
31	91	47	25	14	9	7
32	94	48	25	14	9	7

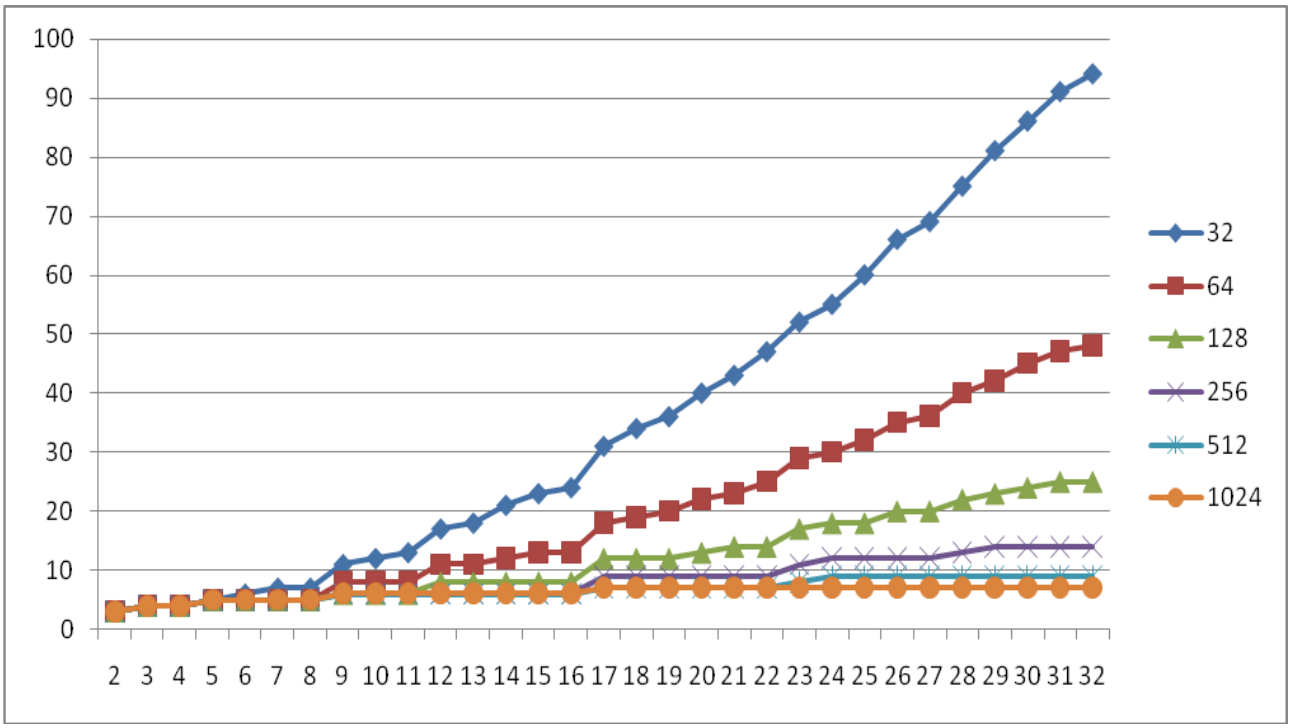

Рис. 8. Залежність кількості операцій над цілими однорозрядними числами, виконаних одним процесором, від кількості доступних процесорів P для $2 \le N \le 32$ при виконанні Алгоритму 5

Таблиця 5. Кількість операцій (додавання та множення) над цілими однорозрядними числами, не більше якої виконає кожен з паралельних процесорів $P = 2^j$, $j = \overline{6,11}$, та $P = N \cdot 2^j$, $j = \overline{-3,3}$, при виконанні Алгоритму 5 для $N = 8i$, $i = \overline{1,16}$

N	**64**	**128**	**256**	**512**	**1024**	**2048**	***N*/8**	***N*/4**	***N*/2**	***N***	**2*N***	**4*N***	**8*N***
16	13	8	6	6	6	6	368	184	92	46	24	13	8
32	48	25	14	9	7	7	752	376	188	94	48	25	14
48	109	57	31	19	13	10	1137	569	285	143	73	38	21
64	190	96	49	26	15	10	1520	760	380	190	96	49	26
80	300	153	79	43	25	16	1906	953	477	240	122	63	34
96	430	217	111	58	32	20	2289	1145	573	287	145	74	39
112	587	296	151	79	43	25	2673	1337	669	335	169	86	45
128	764	382	192	97	50	27	3056	1528	764	382	192	97	50
144	970	488	248	128	68	38	3441	1721	862	433	219	112	59
160	1196	600	303	154	80	44	3826	1913	957	480	242	123	64
176	1450	727	366	186	96	51	4210	2106	1054	528	266	135	70
192	1722	862	433	219	112	59	4593	2297	1149	575	289	146	75
208	2024	1016	511	259	133	70	4978	2490	1246	624	314	159	82
224	2346	1175	590	298	152	80	5361	2681	1341	671	337	170	87
240	2695	1350	678	343	175	91	5745	2873	1437	719	361	182	93
256	3064	1532	766	384	193	98	6128	3064	1532	766	384	193	98

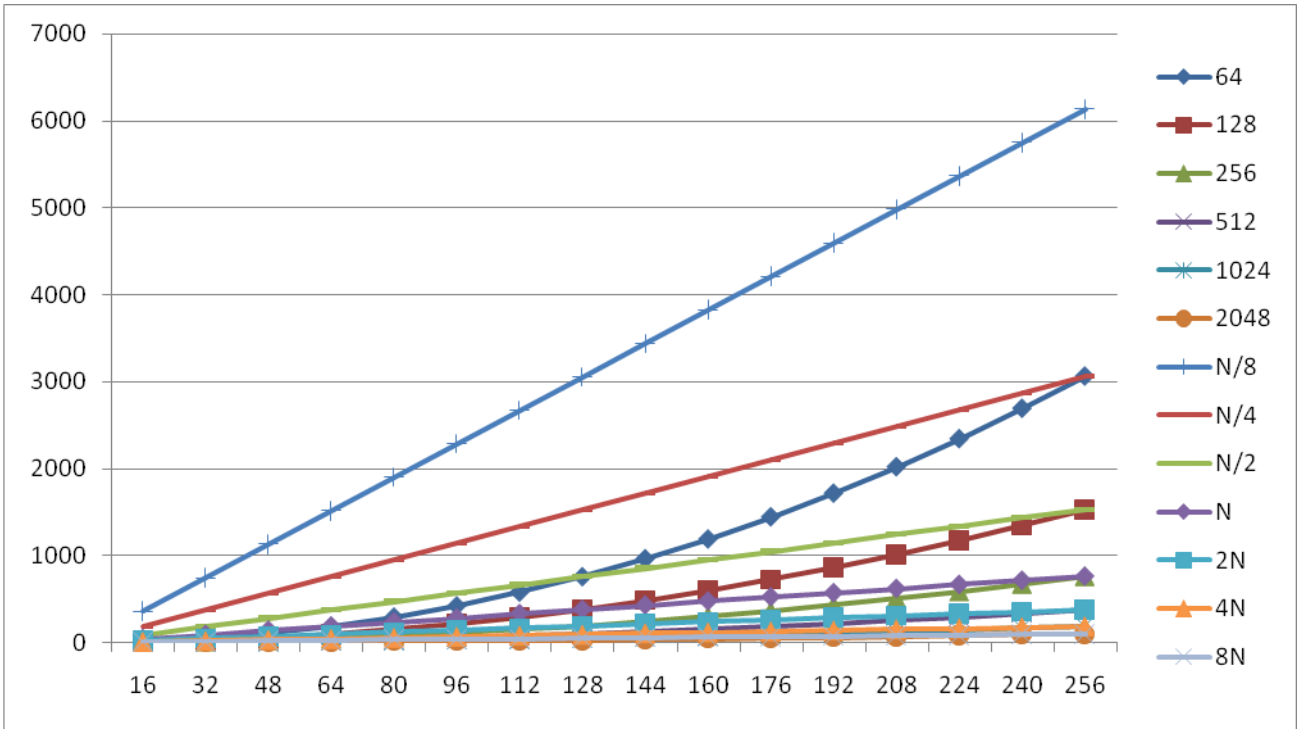

Рис. 9. Залежність кількості операцій над цілими однорозрядними числами, виконаних одним процесором, від кількості доступних процесорів P для $N = 8i$, $i = \overline{1,16}$, при виконанні Алгоритму 5

Найбільш цікавими для дослідження та вибору кількості процесорів є значення розрядністю $31 \le N \le 125$ (з 32 бітами в кожному розряді), тобто бітові довжини від 1000 до 4000, які відповідають найбільш поширеним на даний час довжинам електронних ключів. Сучасні графічні прискорювачі такі, як AMD Radeon HD 7990 Graphics та GeForce GTX TITAN, які мають 4096 та 2688 процесорів відповідно, дозволяють виконувати потужні обчислення на персональних комп'ютерах без використання дорогого серверного обладнання.

Використовуючи формулу (2), див. Лему 7, можна знайти складність за кількістю операцій над цілими числами, виконаних одним з паралельних процесорів графічних прискорювачів AMD Radeon та GeForce GTX, при виконанні Алгоритму 5 (табл. 6 і 7).

Таблиця 6. Кількість операцій (додавання та множення) над цілими однорозрядними числами, не більше якої виконає кожен з паралельних процесорів $P = 2688, 4096, 1536, 2048$ компаній AMD та NVIDIA при виконанні Алгоритму 5 для $N = 8i$, $i = \overline{1, 16}$

N	2688	4096	1536	2048
16	6	6	6	6
32	7	7	7	7
48	8	8	10	10
64	10	8	13	10
80	14	11	20	16
96	16	14	22	20
112	20	16	32	25
128	25	16	37	27
144	28	23	46	38
160	35	26	57	44
176	40	28	68	51
192	46	33	75	59
208	57	38	91	70
224	62	44	105	80
240	70	49	118	91
256	80	51	133	98

Таблиця 7. Характеристики графічних прискорювачів за кількістю процесорів

Виробник	Модель	Процесорів
NVIDIA	GTX TITAN	2688
AMD	RADEON 7990	4096
NVIDIA	GTX 770	1536
AMD	RADEON 7970	2048

Висновок

В даній роботі при виконанні багаторозрядного множення чисел довжини N стандартним методом «у стовпчик» в паралельній моделі обчислень проаналізовано складність за кількістю операцій додавання та множення над цілими однорозрядними числами, виконаних одним паралельним процесором, для двох випадків, коли кількість паралельних процесорів необмежена, та коли кількість процесорів обмежена і кратна N. Для випадку, коли кількість паралельних процесорів необмежна, запропоновано використання Алгоритму 4, для іншого випадку ефективніше використовувати Алгоритм 5 в паралельній моделі обчислень. Для Алгоритмів 4 та 5 доведені співвідношення для обчислення складності за кількістю операцій, виконаних одним паралельним процесором. Показано, що кількість операцій виконаних одним процесором обернено пропорційна кількості задіяних паралельних процесорів та має майже лінійну залежність, для випадку, коли кількість процесорів обмежена. В роботі доведено 7 лем, надані таблиці, на основі яких побудовані два графіки, які показують залежність кількості операцій від кількості задіяних паралельних процесорів для різних довжин N. Розроблена програма *paraNP* на мові програмування *APL*, яка підтверджує теоретичні результати отриманні для оцінки складності за кількістю операцій, виконаних одним процесором. Наведені результати можуть бути використані для обчислення складності при отримані порівняльних характеристик нових алгоритмів (або модифікованих) багаторозрядного множення та існуючих алгоритмів для знаходження ефективних діапазонів їх використання у паралельній моделі обчислень.

1. *Задірака В., Олексюк О.* Комп'ютерна арифметика багаторозрядних чисел. – К.: Наук. думка, 2003. – 263 с.
2. *Schonhage A., Strassen V.* Schnelle Multiplikation grossen Zahnel // Computing. – 1971. – 7, N 3–4. – P. 281–292.
3. *Шенхаге А., Шрассен В.* Быстрое умножение больших чисел // Кибернетика. – 1972. – Вып. 2. – С. 87–98.
4. *Cooley J.W., Tukey J.W.* An algorithm for the machine calculation of complex Fourier Series // Math Compt. – 1965. Apr. – P. 257–301.
5. *Березовский А.И., Задирака В.К., Шевчук Л.Б.* О тестировании быстродействия алгоритмов и программ выполнения основных операций для асимметричной криптографии // Кибернетика и системный анализ. – 1999. – № 5. – С. 61–68.
6. *Терещенко А.Н.* Умножение больших *N*-разрядных чисел с вычислением только *N*-разрядных ДПФ // Компьютерная математика. – 2008. – № 1. – С. 122–130.

Одержано 26.08.2014

Про автора:

Терещенко Андрій Миколайович,
кандидат фізико-математичних наук,
старший інженер-програміст.

Місце роботи автора:

ТОВ «Сімкорп Україна»,
м. Київ,
вул. В. Стуса 35/37.
E-mail: teramidi@ukr.net