Распараллеливание алгоритмов умножения чисел многократной точности

Качко Елена Григорьевна
Харьковский национальный университет радиоэлектроники
ЗАТ «Институт информационных технологий»

Алгоритмы умножения

Алгоритм	**Вычислительная сложность**
Вычисление в «Столбик»	$O(n^2)$
Карацубы и Офмана алгоритм	$O(n^{\log 3})$
Алгоритмы на основе БПФ	O(nlognloglogn)

Математическая постановка задачи. Цель работы

Даны 2 числа:

$$X = \sum_{i=0}^{n-1} x_i * B^i ; Y = \sum_{i=0}^{n-1} y_i * B^i$$

где:

- *B* - основание системы счисления:
- x_i, y_i- ”цифры” сомножителей.

(B = 2^{32} *$B = 2^{64}$)*

Вычислить

Z = X * Y

цель данной работы – дать практические рекомендации по использованию параллельных вычислений при умножении длинных чисел

Вычисление в «столбик». Последовательное вычисление

- Обычный

$$Carry = 0;$$
$$r64 = x_i * y_j + Carry + z_{i+j}$$
$$z_{i+j} = Low(r64)$$
$$Carry = High(r64)$$
$$i, j = 0..n-1$$

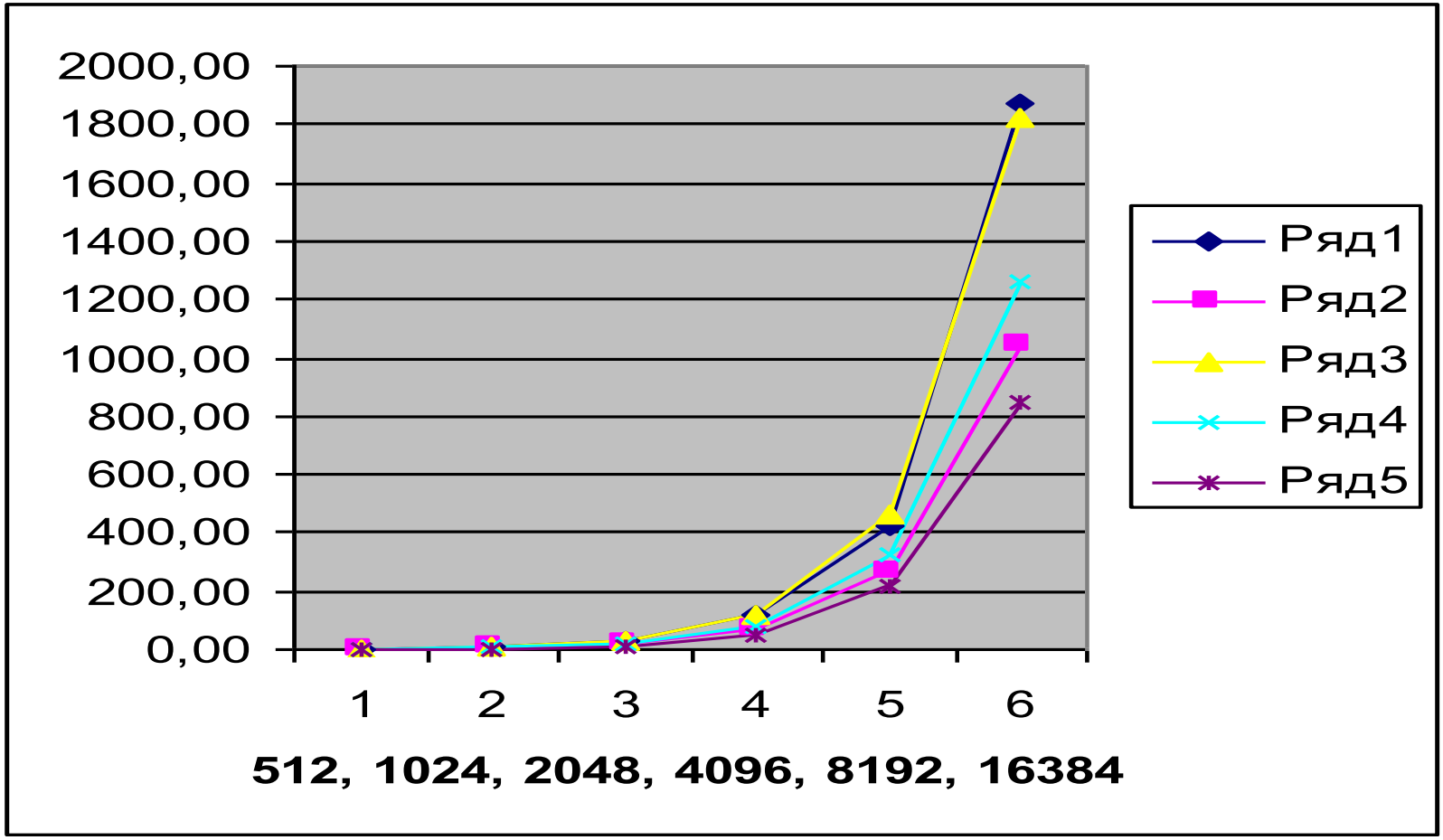
- «быстрый» столбик

$$Z_s = \sum_{i=0}^{i=s} x_{s-i} * y_i; \qquad s = 0..n-1;$$
$$Z_s = \sum_{i=s-n+1;k=n-1}^{i=n-1,k=n-i} x_k * y_i; s = n..2n-1;$$

- TA1 = TA2 = n^2 (tm + ta)

Вычисление в «столбик». Последовательное и параллельное вычисление (p<n/2, p кратно n)

Ряд 1 – Миракла библиотека (29.10.10 5.4.3), Ряд 2 Последовательная функция, Ряд 3 – параллельная (do), Ряд 4 – параллельная (sections)
Ряд 5 – теоретические расчеты(n = 2^{8+x})



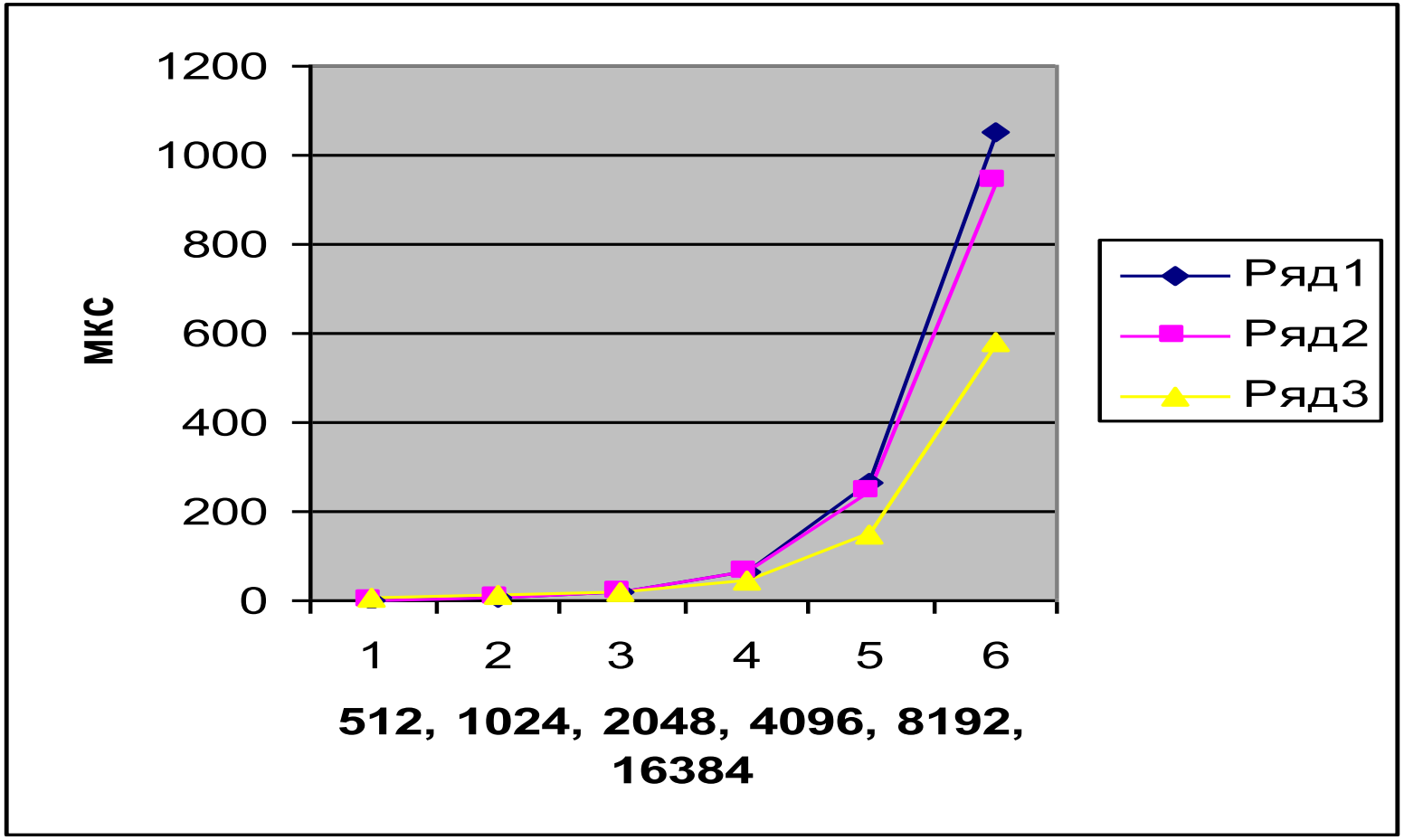
Быстрый столбик

- Max = n * 2^{64}; n < 2^{32}→ Max<2^{96}.

Цифра + перенос (64 бита) (накопление переноса) – carryinternal.

- После вычисления цифры – Накопление переносов в отдельном массиве (+64 бита после вычисления цифры) – переноса быть не может
- Сложение исходного и массива переносов

Быстрый столбик
Ряд 1 – столбик, ряд 2- последовательный быстрый столбик, ряд 3 – параллельный



Алгоритм Карацубы и Офмана

$$X = XH * B^{n/2} + XL$$

$$Y = YH * B^{n/2} + YL$$

$$Z = C * B^{n} + E * B^{n/2} + A$$

$$A = XL * YL; \quad C = XH * YH;$$

$$D = (XL + XH) * (YL + YH);$$

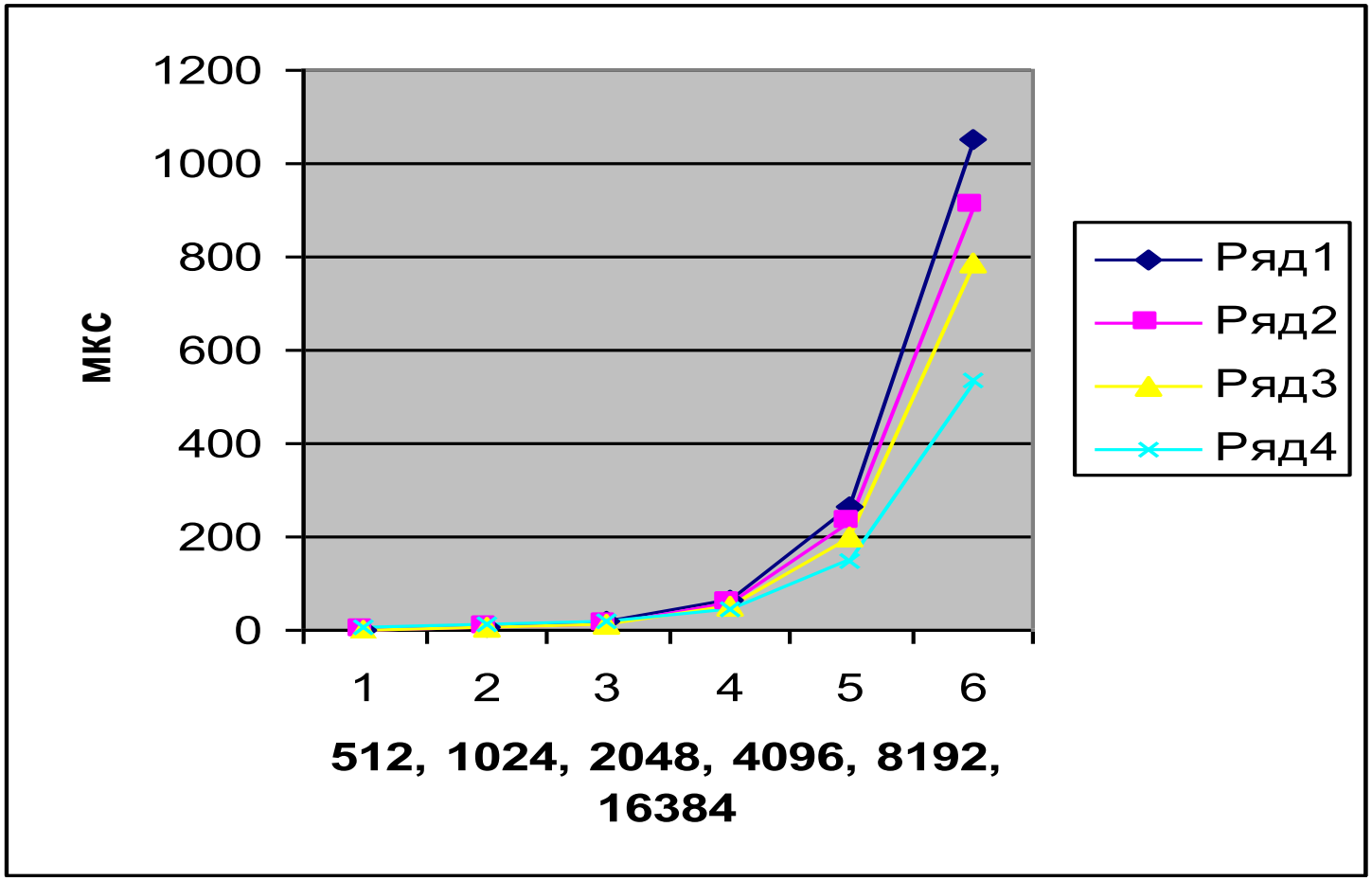
$$E = D - (A + C)$$

$$T_{KO} = (3n^2/4 + n + 1)t_m + (3n^2/4 + 11n/2 + 6)t_a$$

Алгоритм Карацубы и Офмана. Схема вычислений

Ядра процессора			
0	1	2	3
ZL=XL*YL	ZH=XH*YH	XL+XH	YL+YH
ZL+ZH		D= (XL+XH)* (YL+YH)	
E = D- (ZL+ZH)			
Z+=E	8->5		

Алгоритм Карацубы и Офмана
Ряд 1 – столбик, ряд 2-расчетное, ряд 3 – последовательный, ряд 4 - параллельный



Использование БПФ

- Предвычисление.

Выбор простых (предвычисление)

Модуль: $2nB^2$($n = 16384$ бит = 512 слов, $B = 2^{32}$, модуль 2^{74}) ($p1*p2*p3$, $p < 2^{32}$).

Вычисление обратного

Вычисление корней

Использование БПФ. Алгоритм

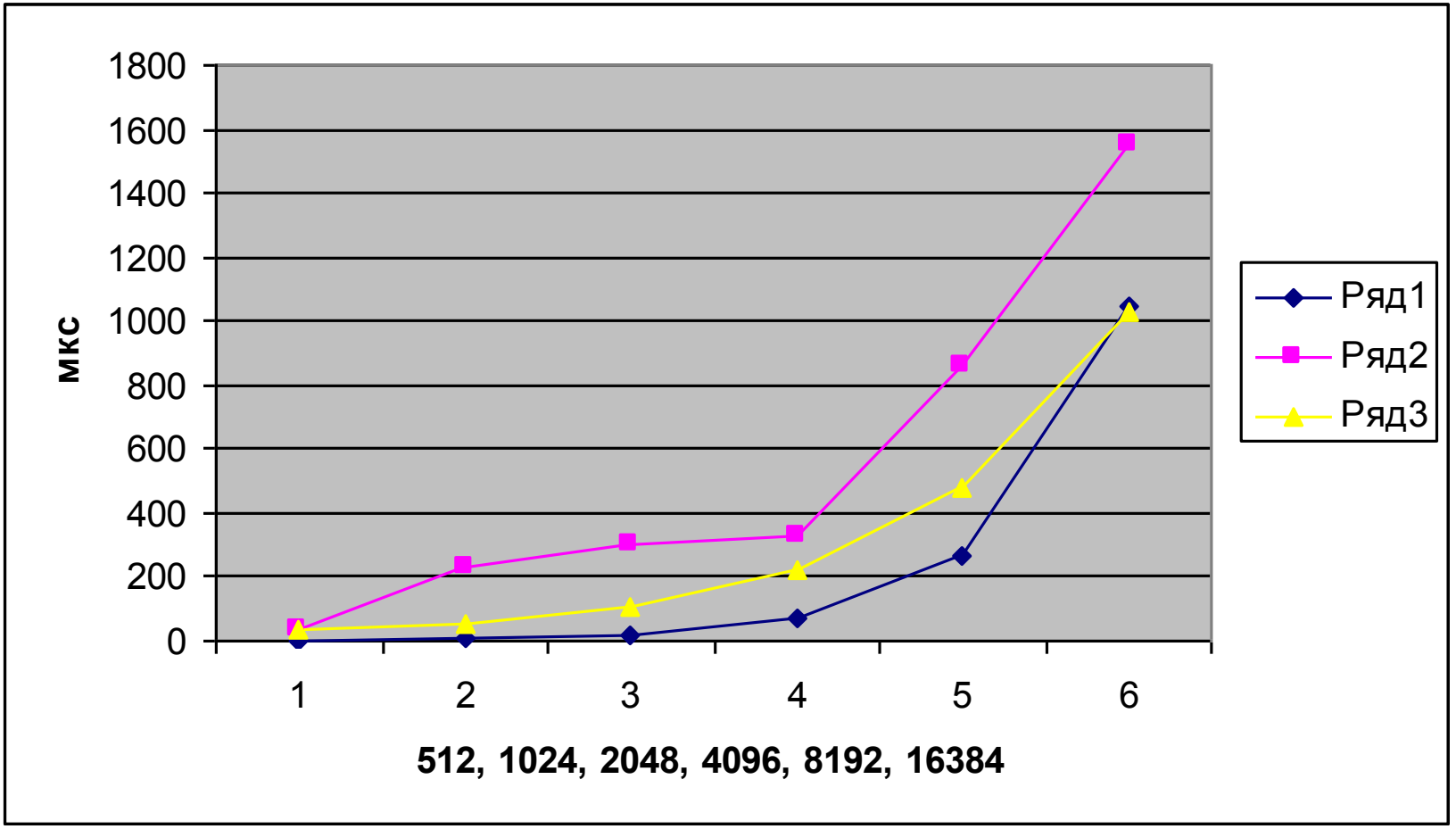
- Для всех простых чисел
  - приведение первого сомножителя по модулю;
  - формирование преобразования Фурье для 1 сомножителя;
  - приведение второго сомножителя по модулю;
  - формирование преобразования Фурье для 2 сомножителя;
  - покомпонентное умножение по модулю;
  - формирование обратного преобразования Фурье.
- Преобразование для получения результата

Использование БПФ..

Схема параллельных вычислений

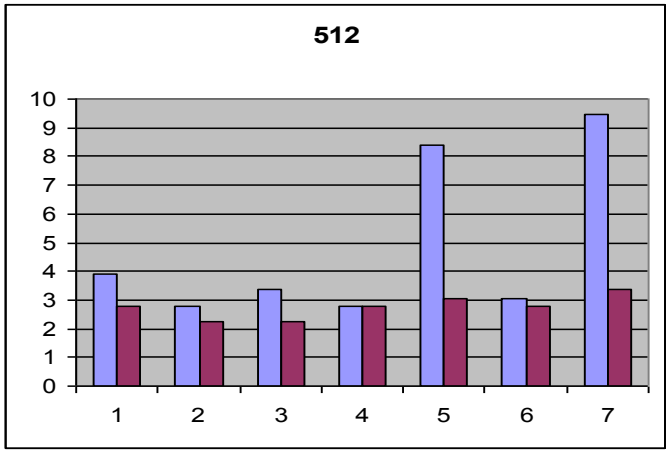
- 1 Получить параллельно прямое преобразование Фурье для каждого сомножителя и каждого простого числа (используется от 6 до 2 ядер в зависимости от возможности процессора).
- 2. Почленно перемножить (3 или 2 ядра)
- 3 Сформировать обратное преобразование Фурье (общая часть) (3 или 2 ядра)
- Учесть изменения для каждого очередного простого числа (число ядер 6..2)
- Преобразование обратного преобразования в число (последовательно)

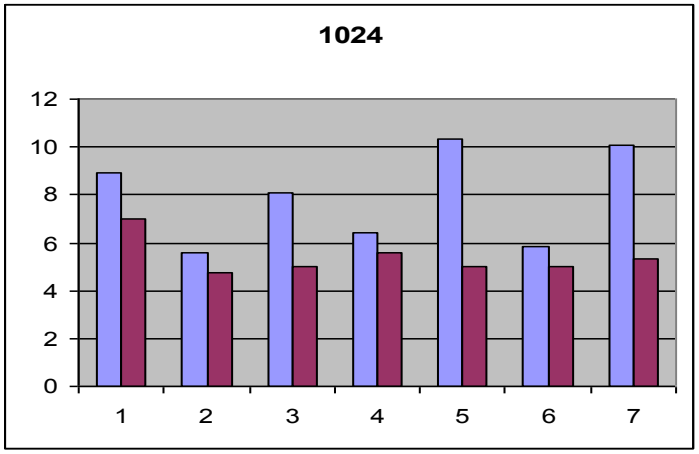
Использование БПФ.
ряд 1 – столбик; ряд 2 – последовательный; ряд 3 параллельный



Рекомендации по использованию методов умножения

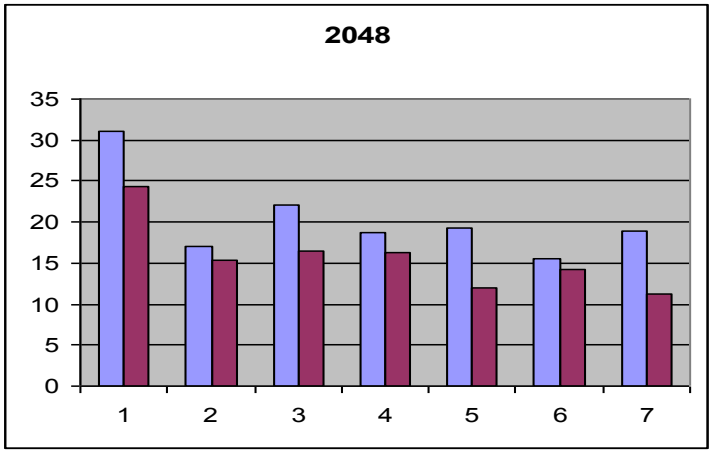
- 1 – Миракла б-ка
- 2 –Послед. Столбик
- 3. Параллельный столбик
- 4 Быстрый столбик
- 5 Быстрый столбик (параллельный режим)
- 6. Карацубы метод
- 7. Карацубы метод (параллельный режим)

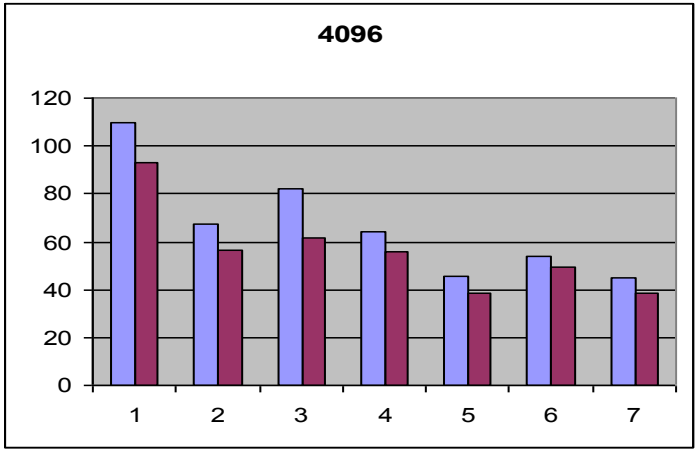




Рекомендации по использованию методов умножения

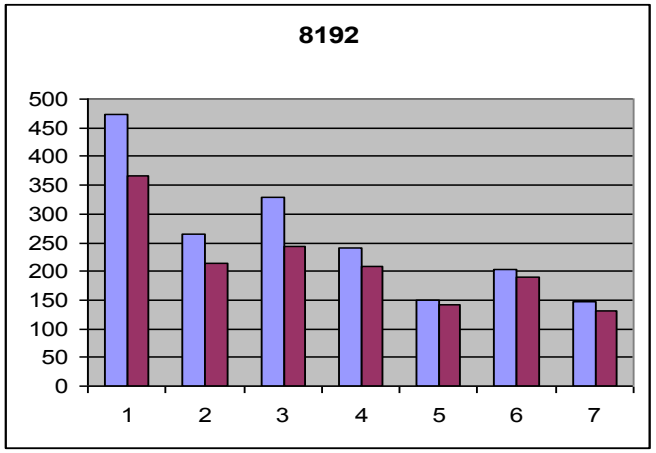
- 1 – Миракла б-ка
- 2 –Послед. Столбик
- 3. Параллельный столбик
- 4 Быстрый столбик
- 5 Быстрый столбик (параллельный режим)
- 6. Карацубы метод
- 7. Карацубы метод (параллельный режим)

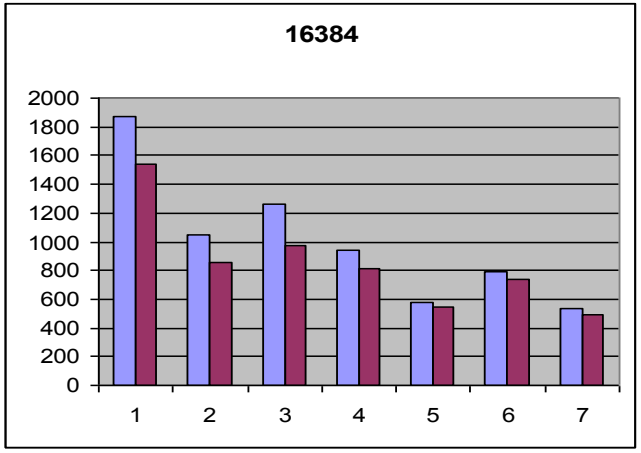




Рекомендации по использованию методов умножения

- 1 – Миракла б-ка
- 2 –Послед. Столбик
- 3. Параллельный столбик
- 4 Быстрый столбик
- 5 Быстрый столбик (параллельный режим)
- 6. Карацубы метод
- 7. Карацубы метод (параллельный режим)





Характеристика ВС и компиляторов

- **Процессор**: Intel (R) Core (TM)2 Duo CPU E6850 @3.00GHz.
- **Среда разработки**: Microsoft Visual C++ 2008.
- **Компиляторы**: Intel(R) C++ Compiler Professional Edition 11.1.
- Microsoft Visual C++ 2008.

Выводы

- Использование преобразования Фурье для сомножителей длиной до 16384 включительно не эффективно.
- Параллельное выполнение для длин сомножителей до 2048 не дает эффекта (гранулярность).
- Начиная с длин сомножителей 2048 бит следует использовать быстрый столбик или метод Карацубы (параллельный вариант) Ускорение составляет 1,38..1.72.
- Код, производимый компилятором Intel, эффективнее кода для Visual Studio при решении этого класса задач (для длины 512 – 33%, 16384 – 13%).
- В связи с увеличением числа ядер производителям процессоров и ОС необходимо стремиться к уменьшению накладных расходов, связанных с использованием потоков.
- Компилятор. При решении сложных задач много вариантов распараллеливания, компиляторы должны иметь средства для перебора этих вариантов (граф и методы прохода этого графа - Фурье)
- Компилятор. Если параллельный вариант не эффективный, то стоит заменять его последовательным (похоже, что Intel компилятор это делает)