

Организация параллельных вычислений в алгоритмах БПФ на процессоре NM6403

Кашкаров В.А., Мушкаев С.В.
НТЦ “Модуль”, г. Москва

Изучаются возможности параллельных вычислений в алгоритмах быстрого преобразования Фурье. Исследуется схема вычислений быстрого преобразования Фурье по основанию 16. Производится сравнительный анализ со стандартным методом вычисления по основанию 2. На примере БПФ-256 описываются принципы построения параллельных вычислений на базе процессора NeuroMatrix® NM6403. Приводятся оценки точности и производительности БПФ процедур для процессора NM6403.

Введение

Значительная часть задач анализа временных рядов связана с преобразованием Фурье и методами его эффективного вычисления. В этих задачах преобразование Фурье играет важную роль как необходимый промежуточный шаг в определении плотностей спектра мощности, кросс-спектральных плотностей, передаточных функций, сверток, корреляционных функций, а также в задачах интерполяции значений.

На практике наиболее широкое распространение получили алгоритмы БПФ по основанию 2 [1], где каждый функциональный узел выполняет базовую операцию – двухвходовую “бабочку”. Эти алгоритмы ориентированы, прежде всего, на сведение к минимуму числа операций умножения. Но с появлением векторных процессоров этот критерий становится несущественным. Напротив, число одновременно выполняемых умножений главным образом определяет производительность процессора. Поэтому возникает вопрос о распараллеливании вычислений и реализации алгоритмов БПФ с более высокими основаниями и их возможными комбинациями

Последовательность вычислений любого БПФ можно описать в виде графа, узлы которого выполняют фактически обычное дискретное преобразование, но с меньшей размерностью входных векторов (меньшим основанием). В зависимости от выбора основания меняется как общее число арифметических операций, так и количество слоев графа (рис. 1).

Таблица 1. Вычислительная сложность БПФ

N	Прямое вычисление ДПФ (основание N)			Вычисление БПФ по основанию 2			Вычисление БПФ с комбинированными основаниями 2,16,32			
	Complex muls	Complex adds	Кол-во слоев графа	Complex muls	Complex Adds	Кол-во слоев графа	Complex muls	Complex adds	Комбинация оснований	Кол-во слоев графа
N	N ²	N ² -N		(N/2)log ₂ N	Nlog ₂ N					
256	65536	65280	1	1024	2048	8	8192	7680	16-16	2
512	262144	261632	1	2304	4608	9	16384	16384	2-16-16	3
1024	1048576	1047552	1	5120	10240	10	49152	49152	32-32	2
2048	4194304	4194304	1	11264	22528	11	131072	131072	2-32-32	3

Complex muls – Число комплексных умножений

Complex adds – Число комплексных сложений

В алгоритмах БПФ по основанию 2 количество таких слоев максимально (табл.1), поэтому при поэтапном поступлении результатов вычислений от слоя к слою происходит большее накопление ошибок округления, нежели в алгоритмах с более высокими основаниями. И чем выше размерность вектора входных данных, тем большим будет количество слоев и в следствие значительнее ошибка. Это особенно критично в случаях, когда вычисления проводятся в целочисленной арифметике (с фиксированной точкой) или при недостаточно широкой разрядности данных. Следует также отметить, что в этом случае для предотвращения переполнения промежуточные результаты после каждого или после группы этапов умножения (слоев графа) необходимо дополнительно нормализовать, применяя операцию сдвига вправо (рис.1). Нормализация помимо сдвига может включать в себя процедуру округления, что также вносит дополнительные вычислительные затраты.

Возможным компромиссным решением может выступать подход, основанный на увеличении основания в алгоритмах БПФ. Ниже рассматривается вариант БПФ-256 по основанию 16. Выбор такого основания с одной стороны дает возможность для организации параллельных вычислений, а с другой снижает количество слоев графа до двух.

Дискретное 256-точечное преобразование Фурье определяется формулой:

$$Y(k) = \sum_{n=0}^{255} W_{256}^{k \cdot n} \cdot X(n), k = 0..255, \text{ где } W_{256}^k = \exp\left(-\frac{2\pi \cdot i \cdot k}{256}\right)$$

Данная формула после тождественных преобразований принимает вид, являющийся опорным для построения БПФ-256 по основанию 16: $Y(k) = \sum_{n=0}^{15} W_{256}^{k \cdot n} \cdot \sum_{i=0}^{15} W_{256}^{16k \cdot i} \cdot X(16 \cdot i + n); k = 0..255$

Конечный граф вычисления БПФ-256 по основанию-16 строится из этой формулы. Структура такого графа показана на рис.1

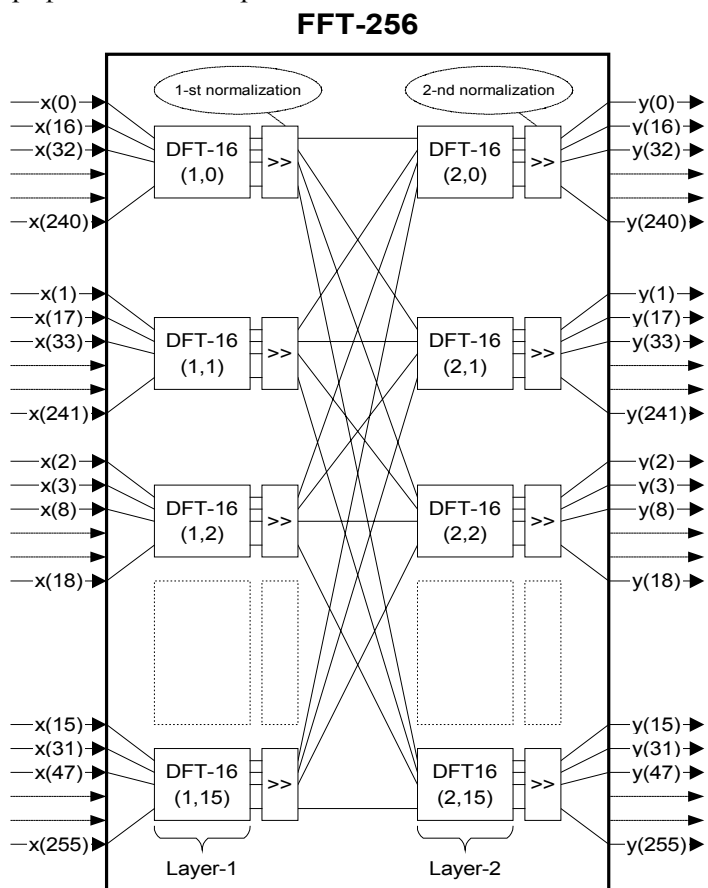


Рис.1 Обобщенный граф вычисления БПФ-256 по основанию 16.

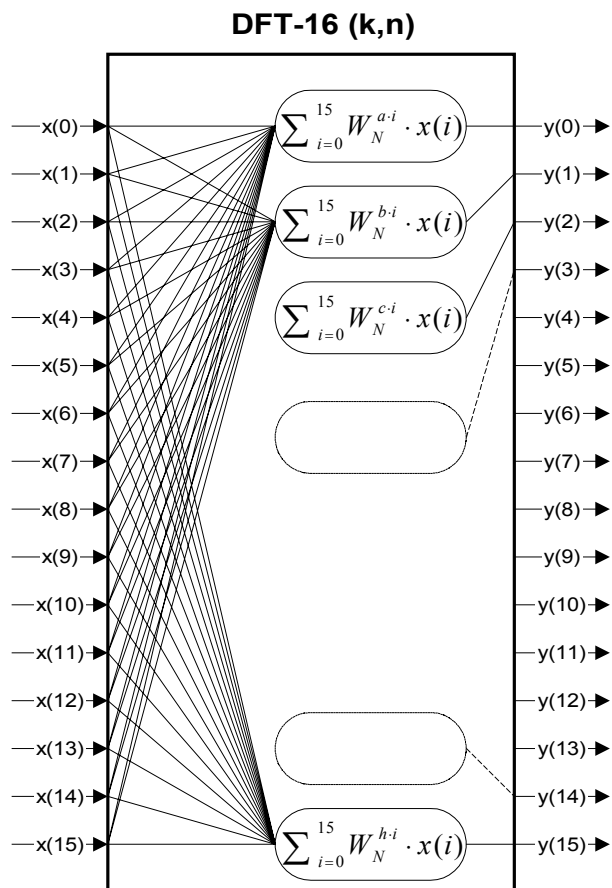


Рис.2 Развернутая схема блока 16-точечного дискретного преобразования Фурье

Граф состоит из двух слоев по 16 блоков. Каждый блок графа имеет 16 комплексных входов и выходов. Как показано на рис.2 каждый блок графа представляет собой 16-точечное дискретное преобразование Фурье и отличается от остальных блоков только комплексными коэффициентами W . Таким образом, распараллеливание алгоритма БПФ фактически сводится к реализации эффективного вычисления ДПФ-16, т.е. к нахождению 16 скалярных произведений различных векторов $[W]$ с одним вектором $[x]$, что эквивалентно умножению матрицы коэффициентов преобразования Фурье $-[W]$ размерностью 16×16 на входной вектор $[x]$.

Организация параллельных вычислений ДПФ-16 на процессоре NM6403.

Эффективное распараллеливание вычислений ДПФ-16 достигается за счет аппаратной поддержки операции векторно-матричного умножения на процессоре NeuroMatrix® NM6403. Все арифметические вычисления, относящиеся непосредственно к вычислению ДПФ-16, производятся на векторном сопроцессоре. Так как векторный сопроцессор позволяет оперировать данными переменной разрядности, то для хранения входных данных и результатов вычислений удобно отводить по 32 разряда на мнимую и действительную часть, а для хранения комплексных коэффициентов W - по 8 бит на действительную и мнимую часть. Таким образом, в одном 64р. слове может содержаться одно комплексное число. Мнимые и действительные части коэффициенты W хранятся так же в упакованном виде, но в разных 64р. словах. Все коэффициенты W вычисляются заранее и поэтому хранятся внутри массива в порядке удобном для последующих вычислений (рис.3).

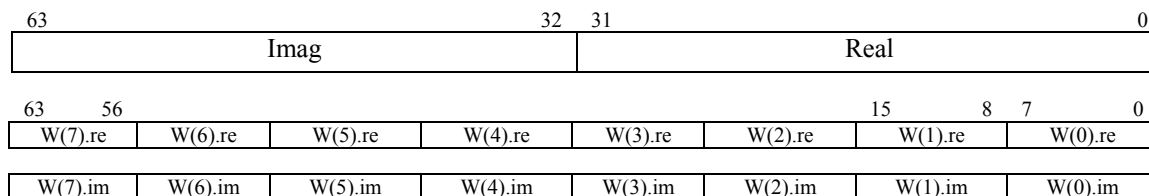


Рис.3 Формат хранения входных данных и коэффициентов преобразования

Вследствие такого представления данных векторный умножитель работает в двух конфигурациях:

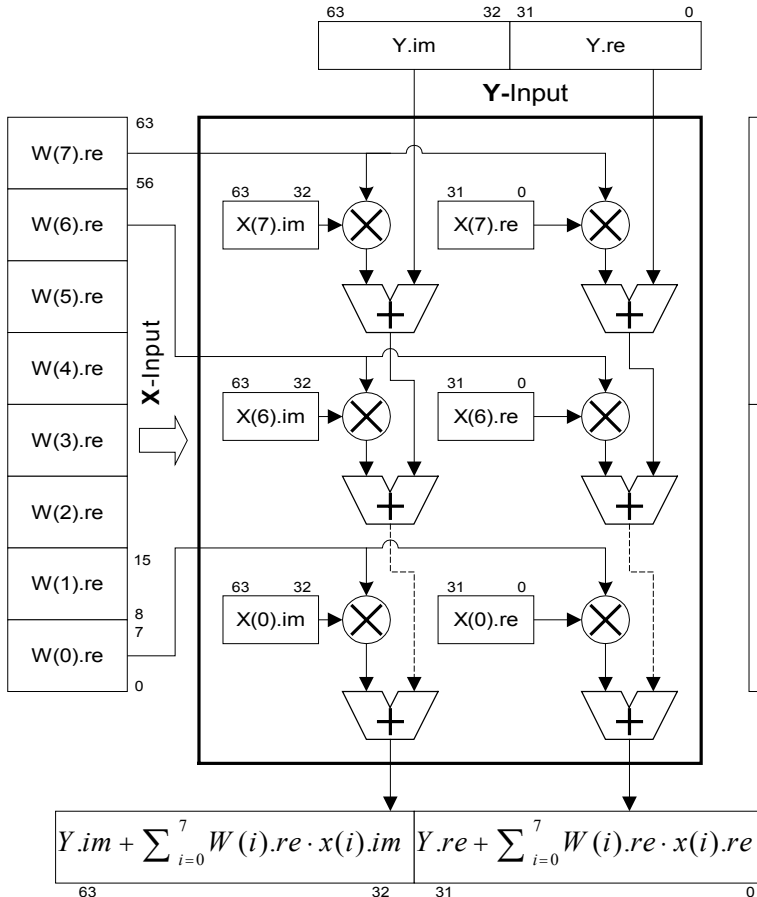


Рис.4 Эквивалентная схема умножителя векторного сопроцессора NM6403 при разбиении матрицы весовых коэффициентов - (2x32бита)/(8x8бит)

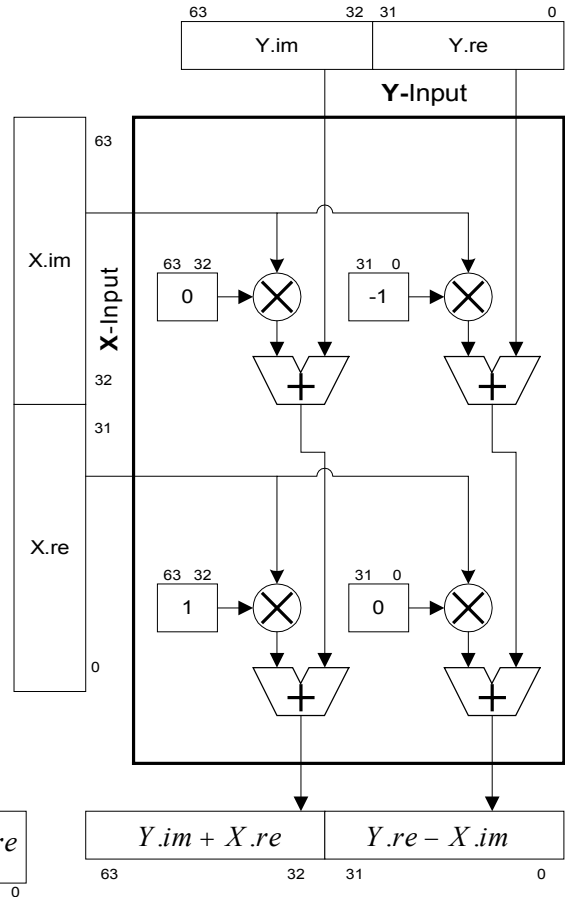


Рис.5 Эквивалентная схема умножителя векторного сопроцессора NM6403 при разбиении матрицы весовых коэффициентов - (2x32бита)/(2x32бита)

По приведенным двум вариантам разбиения матрицы векторного умножителя производится полный процесс скалярного умножения двух комплексных векторов. Первая схема выполняет 16 умножений с накоплением за такт и служит для нахождения сумм попарных произведений мнимых и действительных частей, вторая выполняет 4 умножения с накоплением за такт, но фактически служит только для окончательного сложения полученных частичных сумм. Полная схема умножения двух комплексных векторов длиной 16 элементов отображена на рис.6. Так как за один раз в матрицу весовых коэффициентов можно загрузить только 8 элементов вектора [x], загрузка всего вектора [x] происходит в два этапа.

Весь процесс вычисления скалярного произведения $y(k) = \sum_{i=0}^{15} W_{256}^{a-i} \cdot X(i)$ состоит из трех этапов:

1. В матрицу весовых коэффициентов загружаются 8 комплексных чисел $x(0)..x(7)$. На вход умножителя X поочередно подаются сначала вектор из 8-ми действительных частей комплексных коэффициентов $W(0)..W(7)$ (здесь $W(i) = W_{256}^{a-i}$), а затем вектор из 8-мнимых частей. Умножение производится согласно схеме на рис.4. Результат умножения сохраняется в накопительном FIFO (AFIFO) для последующей обработки.
2. Далее с выхода умножителя (AFIFO) результат произведения в виде двух 64р. слов непосредственно поступает на суммирующий Y-вход умножителя. При этом в матрицу весовых коэффициентов загружаются числа $x(8)..x(15)$, а на вход X умножителя аналогично поступают и умножаются новые коэффициенты $W(8)..W(15)$. Следует отметить, что загрузка чисел $x(8)..x(15)$ сначала производится в теньевую матрицу на фоне вычислений первой стадии, а на второй стадии осуществляется только их быстрое копирование из теньевой матрицы в рабочую. Аналогичным образом загружаются и коэффициенты $x(0)..x(7)$. Результатом вычислений данной стадии являются суммы A,B,C,D:

$A = \sum_{i=0}^{15} W_{256}^{ai} .re \cdot X(i).im$	$B = \sum_{i=0}^{15} W_{256}^{ai} .re \cdot X(i).re$
$C = \sum_{i=0}^{15} W_{256}^{ai} .im \cdot X(i).im$	$D = \sum_{i=0}^{15} W_{256}^{ai} .im \cdot X(i).re$

3. Для получения окончательного результата $-y(k)$, суммы в левых и правых частях двух последних результатов ("3-rd product" и "4-th product") необходимо перекрестно сложить (с учетом знака "-"), т.е. найти $A+D$ и $B-C$. Для этого, как показано на рис.6, в матрицу весовых коэффициентов загружаются числа 0,1 и -1 , суммы A и B подаются на суммирующий вход Y , а суммы C и D – на вход X , и далее, работая по схеме на рис.5, векторный умножитель выдает конечный результат для $y(k)$.

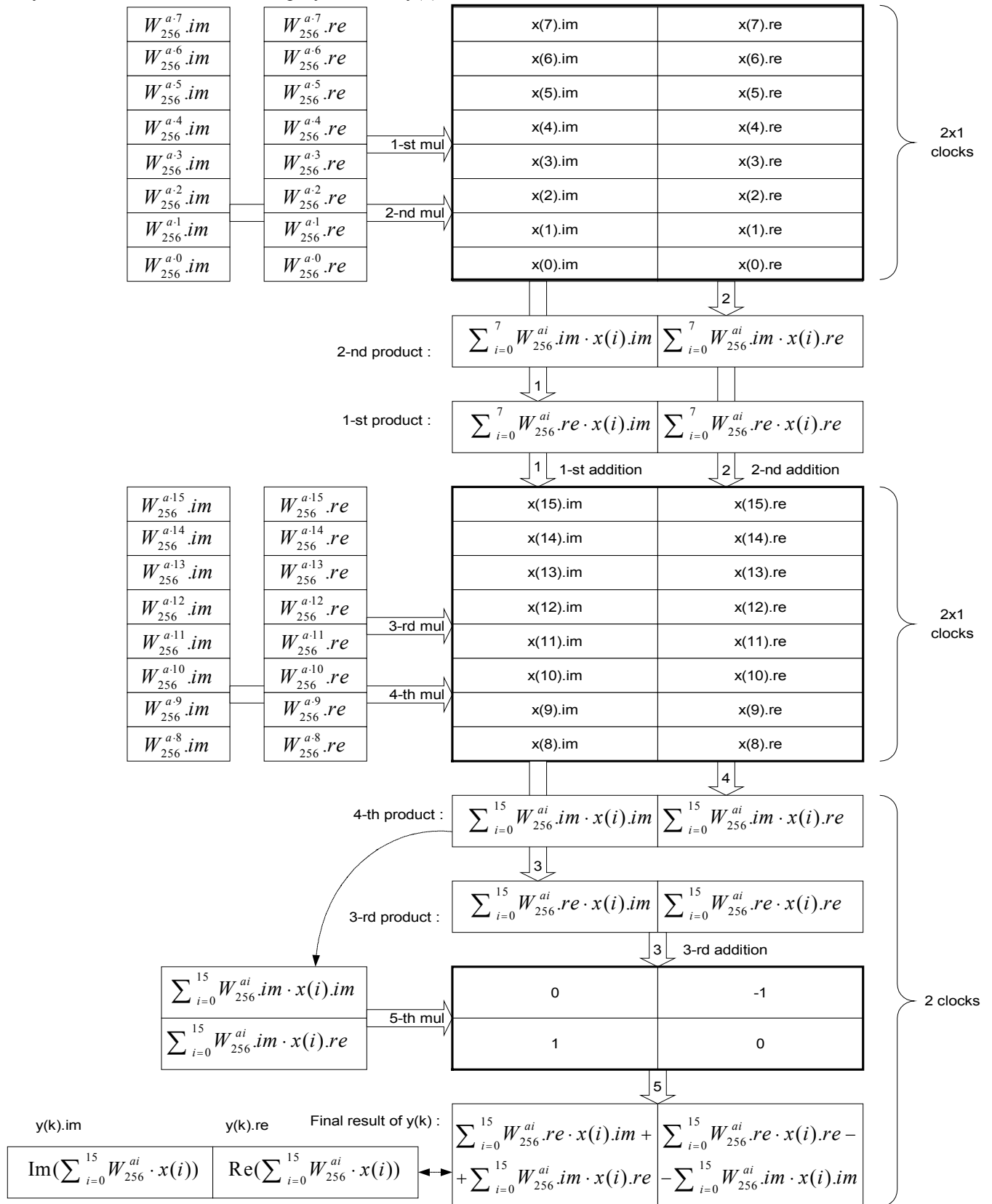


Рис.6 Последовательность вычислений скалярного произведения $y(k) = \sum_{i=0}^{15} W_{256}^{a-i} \cdot X(i)$

Для наглядности на рис.6 проиллюстрирован процесс скалярного умножения только двух векторов. В действительности загрузка входных данных осуществляется пакетами по 32 64-разрядных слова, что

позволяет максимально эффективно использовать векторный сопроцессор. В результате, с учетом времени передачи данных каждый шаг умножения (рис 4, рис 5) практически занимает один процессорный такт, это достигается за счет одновременного использования двух шин данных – подкачка входных данных $x(i)$ по одной шине совмещается с загрузкой коэффициентов $W(i)$ или выгрузкой результатов умножения $y(i)$ по другой. Таким образом, реально вся процедура скалярного умножения двух комплексных 16-мерных векторов в среднем по всему БПФ-256 составляет 7 процессорных тактов.

Производительность и точность вычислений.

Точность вычислений определяется количеством бит отводимых для представления коэффициентов W . Имеется два способа представления значений косинусов и синусов в 8 разрядной сетке:

1. $W = \text{round}(64.0 * \cos(x))$ - (условно 6 бит на единицу)
2. $W = \text{round}(127.0 * \cos(x))$ - (условно 7 бит на единицу)

Первый способ имеет 65 градаций косинуса в диапазоне 0..1, в то время как второй способ – 128 и следовательно обладает более высокой точностью. Однако, выполняя в ходе вычислений только операции целочисленного умножения и сложения, в конце необходимо провести нормализацию результатов, т.е. каждый элемент выходного массива требуется поделить на соответствующий масштабирующий коэффициент. Для первого случая он равен 64^2 , а для второго - 127^2 . В обоих случаях деление целесообразно заменить сдвигом вправо, но в отличие от первого способа замена деления на 127^2 сдвигом на 14 бит вправо привносит небольшую систематическую ошибку. Нормализацию можно проводить либо один раз в конце умножений, либо дважды - после каждого этапа умножения (рис.1) (промежуточная нормализация служит для предотвращения переполнения в ходе вычислений). Количество необходимых этапов нормализации определяется в зависимости от диапазона входных данных см. табл. 3.

Для оценки точности произвольный сигнал обрабатывался прямым и обратным преобразованием Фурье, после чего исходный x сравнивался с восстановленным сигналом x' . В частности, находилось математическое ожидание M и среднее квадратическое отклонение σ относительных ошибок δ .

$$\delta(i) = \frac{x'(i).re - x(i).re}{x(i).re}, i=0..255$$

Таблица 2. Сравнительная характеристика точности восстановленного сигнала после прямого и обратного БПФ с разными основаниями

Преобразование Фурье	Систематическая ошибка-М		СКО - σ	
	6 бит/1.0	7 бит/1.0	6 бит/1.0	7 бит /1.0
FFT-256 по основанию 2	-1%	-12%	2.0%	0.9%
FFT-256 по основанию 16	-0.4%	-3%	1.2%	0.6%

Знак “-” означает, что восстановленный сигнал ослабляется на М процентов по сравнению с исходным.

Как видно из приведенной таблицы 7-битное представление коэффициентов для БПФ по основанию 2 вносит значительную систематическую ошибку и делает такой алгоритм менее эффективным. Алгоритмы БПФ с основаниям 16 и 32 позволяют более эффективно использовать имеющуюся разрядную сетку под коэффициенты W и обладают более высокой точностью за счет в 4-5 раз меньшего числа слоев графа, что также снижает дополнительные затраты на нормализацию и округление. При этом операция округления на процессоре NM6403 выполняется с помощью векторного регистра vr [2] совместно с основными вычислениями, не приводя к дополнительным затратам времени. Также имеется возможность проведения дополнительной оптимизации за счет совмещения процедуры нормализации (арифметического сдвига вправо) с последним этапом вычисления скалярного произведения (рис. 6).

Общая характеристика функций БПФ.

Входные и выходные данные - целые 32-битные комплексные числа, формат хранения показан на рис.3

Диапазон входных данных указан в таблице 3.

Разрядность коэффициентов преобразования – 8 бит (два варианта представления: 6 и 7 бит на единицу)

Работа с данными - арифметика с фиксированной точкой.

Выходные данные расположены в правильном порядке.

Таблица 3. Производительность функций прямого и обратного БПФ на процессоре NM6403

Кол-во комплекс отсчетов	Без нормализации			С одной нормализацией			С двумя нормализациями		
	Тактов	Время мс	Диапазон вх.данных	Тактов	Время, мс	Диапазон вх.данных	Тактов	Время, мс	Диапазон вх.данных
256	3662	0.092	± 512 (7bit) ± 2047 (6bit)	4053	0.1	± 512 (7bit) ± 2048 (6bit)	4429	0.11	$\pm 2^{18}$ (7bit) $\pm 2^{18}$ (6bit)
512	8180	0.2	± 256 (7bit) ± 1023 (6bit)	8930	0.22	± 256 (7bit) ± 1023 (6bit)	9524	0.24	$\pm 2^{18}$ (7bit) $\pm 2^{18}$ (6bit)
1024	18900	0.47	± 128 (7bit) ± 511 (6bit)	20234	0.5	± 128 (7bit) ± 511 (6bit)	22630	0.56	$\pm 2^{17}$ (7bit) $\pm 2^{17}$ (6bit)
2048	47624	1.2	± 64 (7bit) ± 255 (6bit)	50289	1.25	± 64 (7bit) ± 255 (6bit)	52665	1.32	$\pm 2^{17}$ (7bit) $\pm 2^{17}$ (6bit)

NM6403 cycle time=25 ns (40MHz)

Литература

1. Р.Отнес, Л.Эноксен “Прикладной анализ временных рядов”,-М.:Мир,1982
“Нейропроцессор NM6403. Введение в архитектуру” М.: НТЦ ”Модуль”, 1998 <http://www.module.ru/>