

АДАПТАЦИЯ АЛГОРИТМА БПФ ДЛЯ МАТРИЧНО-ВЕКТОРНОГО СОПРОЦЕССОРА NM6407 С ПЛАВАЮЩЕЙ ТОЧКОЙ

инж. Бродяженко А.В., инж. Мушкаев С.В.

ЗАО НТЦ «Модуль» a.brodyazhenko@module.ru

В данной статье рассматривается схема процессорной ячейки векторного сопроцессора NM6407 с плавающей точкой. Описывается операция умножения матрицы на вектор на сопроцессоре NM6407. Рассматриваются схемы БПФ с разными основаниями на примере БПФ-256. Показана организация параллельных вычислений ДПФ-8 на NM6407. Проводится сравнение производительности функций БПФ с другими процессорами.

Введение

В области ЦОС одним из наиболее важных методов анализа сигналов является быстрое преобразование Фурье. Множество задач анализа временных рядов связано с преобразованием Фурье и методами его эффективного вычисления на различных архитектурах. Для решения таких задач хорошо подходят цифровые сигнальные процессоры. В частности, фирмой ЗАО НТЦ «Модуль» разработано семейство таких процессоров с архитектурой NeuroMatrix. Одним из последних процессоров с подобной архитектурой является 1879BM6Я (NM6407) с поддержкой плавающей точки.

Для адаптации алгоритма БПФ под матрично-векторный сопроцессор NM6407 очень важно решить вопрос о том, как уже на первых этапах вычислений БПФ выполнять одновременно по несколько комплексных умножений с накоплением. Также имеет место проблема организации параллельных вычислений БПФ, в силу существования 4 процессорных ячеек у матрично-векторного сопроцессора NM6407. Каждая процессорная ячейка способна выполнять операции сложения, вычитания, умножения (с накоплением и без него) над векторами чисел с плавающей точкой (одинарной, двойной точности, в том числе комплексными).

На рисунке 1 изображена схема процессорной ячейки матрично-векторного сопроцессора NM6407, которая содержит 8 векторных регистров (VREG0-VREG7), емкостью по 32 элемента каждый и операционное устройство (ОУ), состоящее из матрицы умножителей и сумматоров. Под элементом здесь представляется 64-х разрядное слово, которое может интерпретироваться, либо как число double, либо как 2 числа float, либо как комплексное число одинарной точности (действительная и мнимая части комплексного числа упакованы в одно слово). Нужная интерпретация такого элемента задается в ассемблерной команде.



Рисунок 1 Схема процессорной ячейки матрично-векторного сопроцессора

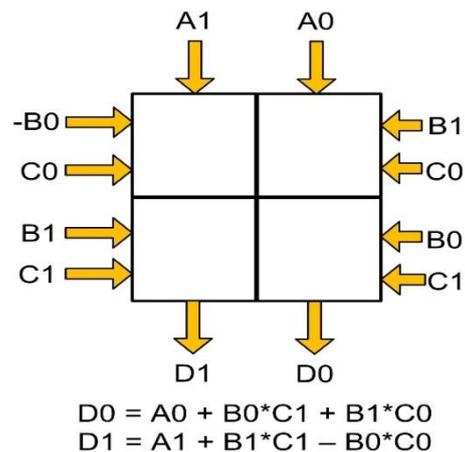


Рисунок 2 Умножение с накоплением комплексных чисел одинарной точности

Базовым действием операционного устройства является умножение с накоплением, которое может выполняться над типами данных, описанными выше. Также ОУ умеет обрабатывать матричные данные типа float (умножение с накоплением матрицы 2x2 на 2-элементный вектор).

Независимо от типа данных умножение с накоплением выполняется за такт. На рисунке 2 схематически изображена операция умножения с накоплением комплексных чисел на NM6407.

Здесь B0 и B1, C0 и C1, A0 и A1, а также D0 и D1 – это действительные и мнимые части (типа float) 4-х комплексных чисел[1].

Операция умножения матрицы на вектор для NM6407

В силу особенностей архитектуры, описанных выше, векторный сопроцессор NM6407 способен быстро перемножать матрицы на вектора различных размеров. Последовательность выполнения подобных операций на сопроцессоре NM6407 можно хорошо представить на примере умножения матрицы комплексных чисел одинарной точности размером 4x4 на 4-элементный вектор комплексных чисел (см. рисунок 3).

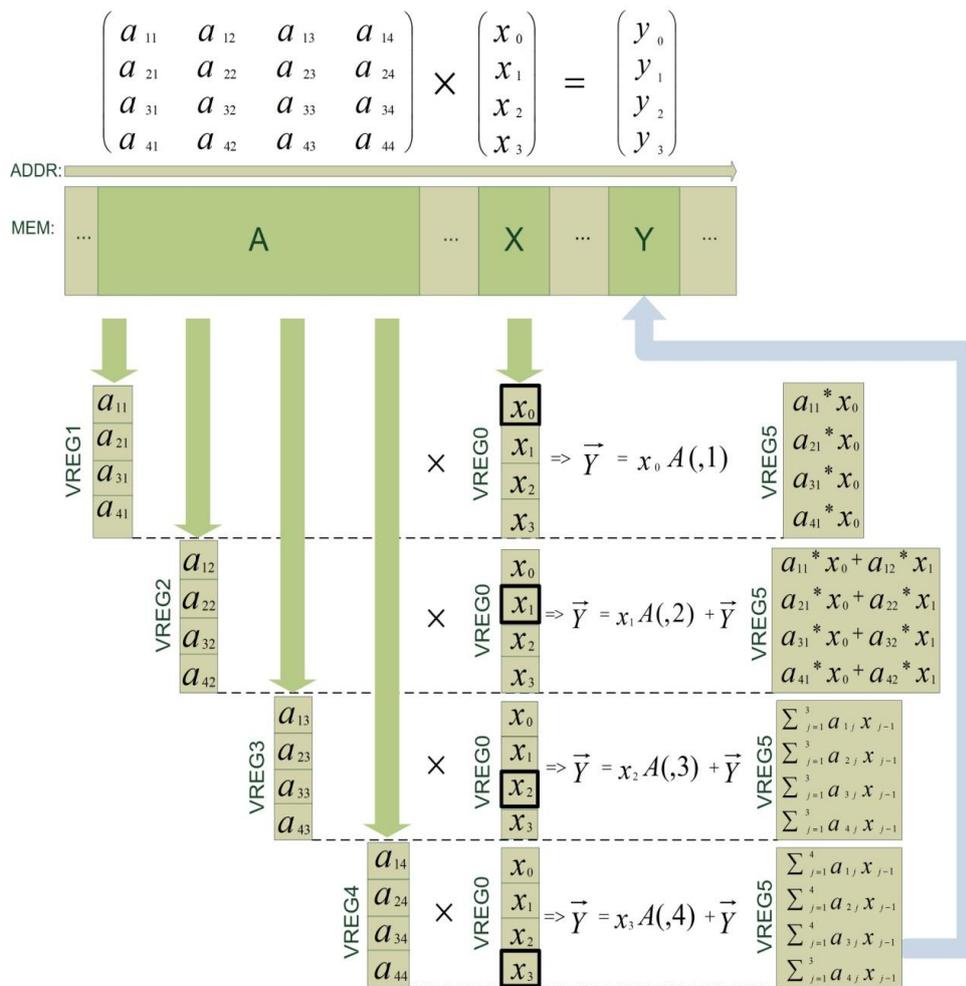


Рисунок 3 Умножение матрицы 4x4 на 4-элементный вектор на NM6407

Столбцы матрицы A : $A(1)$, $A(2)$, $A(3)$ и $A(4)$, загружаются из памяти в векторные регистры: $VREG1$, $VREG2$, $VREG3$ и $VREG4$ соответственно. Аналогично из памяти в регистр $VREG0$ попадает 4-элементный вектор \vec{X} . После чего, происходит поэлементное комплексное умножение с накоплением каждого столбца матрицы, находящегося в векторном регистре процессорной ячейки, на один из элементов вектора \vec{X} , лежащего в $VREG0$. Накопление результата происходит в регистре $VREG5$. Поочередный выбор действующего элемента регистра $VREG0$ на каждом этапе вычислений показан черным квадратом на рисунке 3 и производится автоматически с помощью специальной команды. Таким образом, для умножения матрицы 4x4 на 4-элементный вектор необходимо выполнить 4 команды умножения с накоплением векторного регистра (столбца матрицы A) на скаляр.

Выбор оптимального алгоритма вычисления БПФ на NM6407

Экспериментально было выяснено, что матрично-векторный сопроцессор NM6407 имеет наименьшие задержки в конвейере при глубине заполнения векторного регистра от 4-х элементов и выше. Будем считать, что при вычислении N – точечного ДПФ на NM6407, глубина заполнения векторных регистров равна N (в данном случае $4 \leq N \leq 32$). Тогда возможны 4 варианта алгоритма БПФ, подходящие для реализации на NM6407. Эти варианты

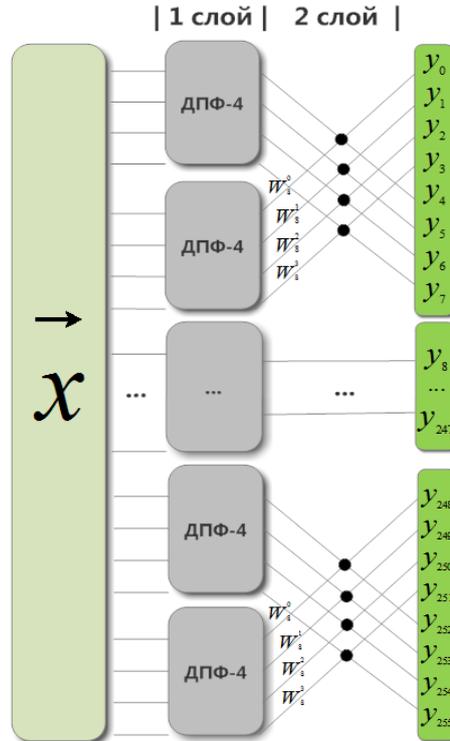


Рисунок 4 Схематическое представление слоя в БПФ-256

проиллюстрированы на рисунке 4. Для примера взято БПФ-256. Варианты отличаются размерностью блоков ДПФ, используемых в 1-ом слое, числом этих блоков и числом слоев во всем БПФ-256. Под слоем здесь и далее понимается набор операций взвешенного умножения с накоплением комплексных чисел. Число операций любого слоя равно размерности БПФ, то есть 256. На рисунке 4 схематически показаны действия, выполняющиеся во 2-ом слое 1-ого варианта алгоритма БПФ. Аналогичные действия выполняются в остальных слоях.

Очень важным шагом в адаптации алгоритма БПФ под NM6407 является выбор основания блока ДПФ для 1-ого слоя. С одной стороны, использование блоков ДПФ-4 позволяет снизить число операций в 1-ом слое (1024 операции для 1-ого варианта против 2048 для 2-ого), с другой, увеличивается общее число слоев во всем алгоритме (5 в 1-ом варианте, против 4-ех во 2-ом). Решающим фактором при выборе оптимального варианта является скорость вычисления блока ДПФ определенной размерности на NM6407. При вычислении ДПФ-4, на обработку 1-ого элемента тратится ~ 3 такта процессорного времени, а в случае ДПФ-8 ~ 1.4 такта. Для ДПФ-16 и ДПФ-32 время обработки 1-ого элемента стремится к такту. Но использование ДПФ-16 или ДПФ-32 ведет к значительному увеличению числа операций в 1-ом слое. Например, для ДПФ-32 число операций в 1-ом слое равно 8192 против 2048 для ДПФ-8.

Оптимальным решением здесь будет выбор блоков ДПФ-8 для 1-ого слоя. Такое решение приводит к уменьшению числа дополнительных слоев, возникающих при использовании ДПФ меньшей размерности. Также уменьшается число блоков ДПФ в 1-ом слое.

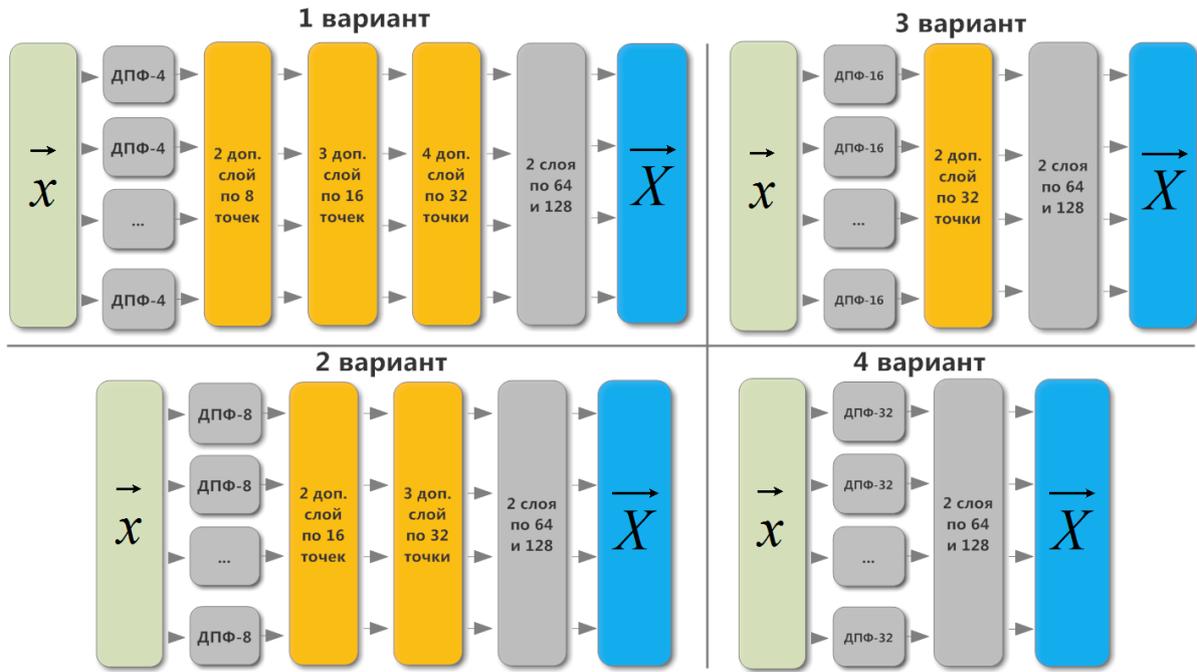


Рисунок 5 Возможные варианты алгоритма на примере БПФ-256

Параллельные вычисления ДПФ-8 на NM6407

Представим вычисление ДПФ-8 как умножение матрицы коэффициентов $W_N^{(n-1)(k-1)}$ размером 8×8 на входной 8-элементный вектор \vec{x} [2]:

$$\vec{X} = W \vec{x}, \text{ где } W_N^{(n-1)(k-1)} = e^{-\frac{2\pi j}{N}(n-1)(k-1)} \text{ коэффициенты, } N = 8;$$

Тогда ДПФ-8 можно вычислить на NM6407 так же, как и умножение матрицы на вектор, описанное выше. Единственное отличие здесь – это изменение глубины заполнения векторных регистров с 4-х на 8 элементов. На двух процессорных ячейках вычисляется один блок ДПФ-8. Соответственно 4 ячейки вычисляют сразу 2 блока ДПФ-8. Матрица коэффициентов W для любого блока ДПФ-8 из 1-ого слоя одна и та же. Поэтому, целесообразно, на этапе вычислений 1-ого слоя хранить матрицу W в 2-х экземплярах на 4-х ячейках.

Идея параллельного вычисления блока ДПФ-8 заключается в том, что первые 4 столбца (32 коэффициента) матрицы W загружаются в 4 векторных регистра (по 8 элементов в каждый) 1-ой процессорной ячейки. Следующие 4 столбца, параллельно предыдущим, загружаются во 2-ую процессорную ячейку. После чего коэффициенты из 1-ой ячейки копируются в 3-ью, а из 2-ой в 4-ую. Копированием здесь заменено чтение из памяти той же матрицы W , которая уже находится в векторных регистрах 1-ой и 2-ой ячеек.

На рисунке 6 в виде блок-схемы показан процесс вычисления двух блоков ДПФ-8 на 4-х процессорных ячейках. Блоки IMU0...IMU7 – это банки статической памяти (SRAM) процессора. Очень важно, чтобы входной вектор \vec{x} , выходной вектор \vec{X} , а также соответствующие коэффициенты матрицы W находились в разных банках памяти процессора. Это необходимо для того, чтобы ячейки могли одновременно обращаться в память, распараллеливая, таким образом, вычисления. Необходимо отметить, что каждая процессорная ячейка способна за такт осуществлять до 2-х операций чтения из памяти и 1 операцию записи в память. А в целом, коммутатор памяти позволяет в одном и том же такте осуществить следующие операции:

1. До 4-х чтений из памяти по 64-разрядным шинам.
2. До 2-х записей в память по 64-разрядным шинам.
3. До 5-и пересылок данных между процессорными ячейками и блоком упаковки и распаковки данных.

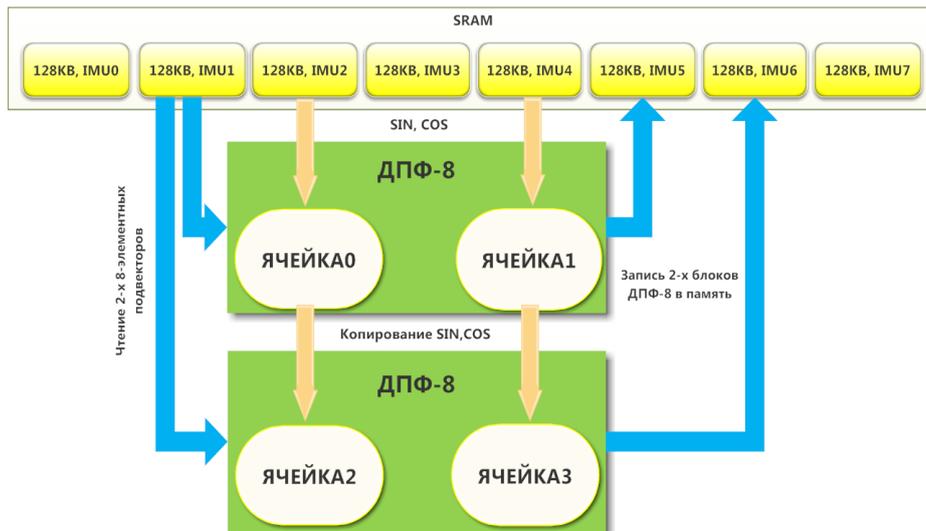


Рисунок 6 Параллельное вычисление ДПФ-8 на матрично-векторном сопроцессоре

Заключение

В таблице 1 представлены значения производительности функций БПФ в тактах для процессоров NM6407, Texas Instruments C674x, Intel Pentium 4 и ARM Cortex A-15. Как видно из результатов, процессор демонстрирует высокую эффективность на уровне ведущих мировых производителей. По сравнению с целочисленным процессором предыдущего поколения NMC3 в новом процессоре NM6407 число тактов на задачах БПФ сократилось на треть. За счет аппаратной поддержки матрично-векторных умножений над данными одинарной и двойной точности, включая комплексные, применение процессора NM6407 особенно эффективно в задачах линейной алгебры и цифровой обработки сигналов.

Таблица 1 – Сравнение производительности функций БПФ на разных процессорах

БПФ	NM6407 (500МГц)		C674x DSPs		Pentium 4	ARM Cortex A-15	
	Такты	Время, мкс	Такты	Время, мкс	Такты	Такты	Время, мкс
128	1290	2.58	-	-	1987	-	-
256	2276	4.55	2401	5.27	4178	8644	8.64
512	5587	11.17	-	-	9725	-	-
1024	14523	29.04	10950	24.01	22382	43916	43.92

Значения производительности для процессора Pentium 4 получены на основе библиотеки Intel IPP. Производительность C674x и ARM процессоров была взята с сайта www.ti.com. Процессоры имеют следующие частоты: 500 МГц (NM6407), 456 МГц (C674x), 3.06 ГГц (Pentium 4), 1 ГГц (ARM Cortex A-15).

Литература

1. Микросхема интегральная NM6407. Руководство по эксплуатации // Москва, НТЦ «Модуль», 2015.
2. Википедия [Электронный ресурс]. Дата обновления: 08.01.2017. – URL: https://ru.wikipedia.org/wiki/Дискретное_преобразование_Фурье