Санкт-Петербургский Государственный Университет
Математико-механический факультет

Кафедра системного программирования

Реализация конфигурируемого аппаратного блока для вычисления быстрого преобразования Фурье переменной длины по смешанному основанию с использованием HLS

Курсовая работа студента 345 группы
Шеина Романа Евгеньевича

Научный руководитель С. И. Салищев

Санкт-Петербург
2011

Оглавление

Введение

Быстрое преобразование Фурье(fast fourier transform, далее FFT) — известный способ ускорения подсчёта дискретного преобразования Фурье, определяемого формулой

$$F_N(k) = \sum_{n=0}^{N-1} f(n)\, e^{\frac{-2\pi k n i}{N}}$$

FFT — представление дискретного преобразования Фурье на N точках в виде множества последовательных дискретных преобразований Фурье меньшего размера. Обычно выделяют 2 основных подхода: расслоение в частотной области(DIF) и расслоение во временной области(DIT), отличающихся способом разбиения большого преобразования на меньшие. Размер меньших преобразований называется основанием FFT, меньшие преобразования организованы в стадии, каждая стадия обрабатывает одну позицию(цифру в системе исчисления по базе основания) из нумерации точек преобразования. FFT, состоящее из стадий с разными основаниями, называется FFT со смешанным основанием.

Для получения явных формул для FFT можно выделить ключевую операцию "бабочка", задаваемую формулой:

$$B(f_1, f_2, \ldots, f_k) = \sum_{j=1}^{k} f_j\, e^{\frac{-2\pi j i}{k}}$$

Тогда стадия номер c в FFT функции f длины $N = r_0 \cdot \ldots \cdot r_{R-1}$ с основаниями $r_0, \ldots, r_{R-1}$ при подходе DIT задаётся следующей формулой:

$$F_{C+1}(d_0, \ldots, d_{R-1}) = B(g_0, \ldots, g_{r_{R-c-1}-1})$$

$$g_j = e^{\frac{d_j \cdot ([d_{R-1}, \ldots, d_{R-c}] \cdot mod(r_{R-1} \cdot \ldots \cdot r_{R-c}))}{r_{R-1} \cdot \ldots \cdot r_{R-1-c}}} \cdot F_C(d_0, \ldots, d_{R-c-2}, j, d_{R-c}, \ldots, d_{R-1})$$

$$F_0(d_0, \ldots, d_{R-1}) = f([d_{R-1}, \ldots, d_0])$$

FFT повсеместно используется для подсчёта преобразования Фурье на практике, например, в реализациях мультиплексирования с ортогональным частотным разделением каналов(OFDM) — методе, применяемом в стандартах IEEE 802.11, 802.16. Активное использование FFT в реализациях устройств передачи сигналов приводит к востребованности специализированных модулей для подсчёта FFT с высокой скоростью.

На настоящий момент существует множество разнообразных аппаратных реализаций FFT. Среди них можно выделить два принципиально различных подхода: основанные на очереди и основанные на памяти с произвольным доступом архитектуры. Как отмечено в [1], основанные на памяти с произвольным доступом архитектуры оказываются намного эффективнее при

реализации FFT больших размеров, и потому им в последнее время уделяется всё больше внимания. В данной работе рассматриваются основанные на памяти с произвольным доступом архитектуры.

В основанных на памяти с произвольным доступом архитектурах память — один из самых дорогостоящих элементов, и поэтому её необходимо минимизировать. Для выполнения операции "бабочка" размера r за один такт необходимо иметь возможность считать r входных значений и записать r выходных значений за один такт. При этом требуется обеспечить отсутствие конфликтов: входы и выходы каждой "бабочки" должны располагаться в различных банках, "бабочка" не должна "портить" входы/выходы других бабочек из своей стадии. Для обеспечения всех этих требований часто используют алгоритмы без использования дополнительной памяти(in-place алгоритмы) на двухпортовой памяти: адреса входов и выходов "бабочки" берутся одинаковыми(иногда с перестановкой). При таком подходе адреса требуется считать только один раз за "бабочку", кроме того, очевидным образом получается отсутствие конфликтов между разными "бабочками" внутри одной стадии.

FFT длины $N=r^R$, где r - основание, подробно описаны в [2], в этой же работе описывается способ распространения предложенного алгоритма адресации на FFT произвольной длины, а в [3] предлагается обобщение этого способа на FFT произвольной длины с произвольными основаниями "бабочек". Однако и этот результат может быть улучшен: например, для $N=8^R\cdot 2$ при правильно подобранной адресации можно запускать 3 "бабочки" с $r=2$ за такт.

Постановка задачи

Первоначальной задачей являлась аппаратная реализация FFT, допускающая оптимизацию FFT длины $N = 2^k \cdot (2^l)^{R-1}, l > k$ посредством запуска в стадии с основанием 2^k в одном такте $2^{(l-k)}$ "бабочек" и изменение направления(прямое/обратное преобразование Фурье) и размера FFT .

Стоит отдельно выделить задачу поиска правильной стратегии адресации для FFT процессора, эту задачу условно можно разделить на две: корректное распределение банков(для отсутствия конфликтов между входами/выходами одной "бабочки") и корректное распределение адресов внутри банка(для отсутствия конфликтов между "бабочками" в одной стадии). Надо заметить, что эти задачи ортогональны: любое корректное распределение банков должно сочетаться с любым корректным распределением адресов.

Решение задачи

В процессе исследования вопроса адресации для оснований - степеней 2 возник вопрос об обобщении на произвольные основания. В результате получена формула для описания корректных распределений банков, включающая в себя способы [2], [3] как частные случаи. Также применимость формулы была проверена на примере, не покрываемом [2], [3](см. Приложение 1).

Надо заметить, что полученная общая формула не имеет доказательства и далеко не во всех случаях может дать корректный результат. Для получения из неё итогового распределения банков может потребоваться решение систем в модульной арифметике. Разрешимость получающихся систем и её соотношение с разрешимостью задачи корректной адресации не рассматривались в рамках этой работы.

Получен метод адресации для FFT длины $N = p \cdot r^{R-1}, r = p \cdot q$ (см. Приложение 2), доказана его корректность.

Для распределения адресов применяется подход, описанный в [1].

Для оснований - степеней 2 на языке C++ реализована модель FFT, использующая предлагаемые методы адресации, начата разработка FFT процессора средствами HLS на основании кода модели(ввиду большой трудоёмкости завершить эту работу до написания отчёта не представляется возможным).

В настоящий момент дописывается статья с подробным описанием предлагаемого способа адресации и доказательствами корректности для случая $N = r^p \cdot (r^q)^{R-1}, q > p$.

Приложение 1

Общая формула для распределения банков

В данном приложении приводится только сама формула и примеры иллюстрирующие , её получение не включено в текст, чтобы избежать перенасыщенности техническими деталями.

Гипотеза: нижеуказанная формула хорошо применима для получения корректного распределения банков

$$m(D)=((g_0\cdot D+g_1\cdot\lfloor D/r_0\rfloor+\ldots+g_{R-1}\cdot\lfloor D/(r_0\cdot\ldots\cdot r_{R-2})\rfloor)+x)\,mod\,B$$

где B - количество банков, D - номер точки, для которой определяется банк, r_i - основания, $g_{0,}\cdots,g_{R-1}$ - некие определяемые основаниями константы, а x - произвольная постоянная.

В качестве примеров можно рассмотреть распределение банков [2], а также случай взаимно простых оснований и FFT на 1536 точек.

Распределение [2] получается из формулы при $g_0=0, g_1=1-r_{0,}$ $\ldots g_{R-1}=1-r_{R-2}$.

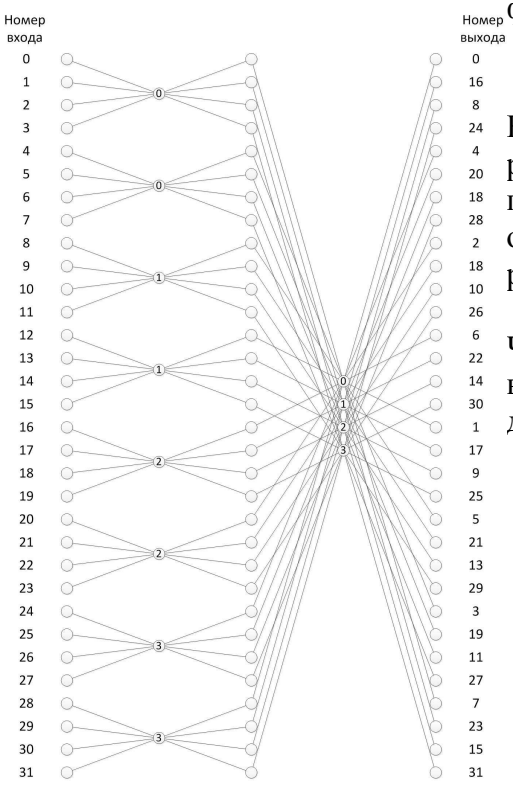
При взаимно простых основаниях может использоваться распределение банков $m(D)=D\,mod\,r$, где r – наибольшее основание.

Распределение банков для FFT на 1536 точек получено из общей формулы подстановкой конкретных значений $r_0=3, r_1=r_2=r_3=8$ и решением получающейся системы. Нетрудно убедиться, что подстановка констант даёт корректное распределение банков.

Приложение2

Распределение банков для случая $N = p \cdot r^R, r = p \cdot q$

Распределение банков для указанного случая получается из общей формулы(см. Приложение 1) при $g_0 = q, g_1 = g_2 = \ldots = g_{R-1} = 1$.



В качестве иллюстрации действия этого распределения прилагается диаграмма потока данных для FFT на 32 точки с основаниями 4 и 8 и таблица с распределением банков.

Числом в круге обозначен номер такта внутри стадии, на котором выполняется данная "бабочка".

номер точки	банк	номер точки	банк	номер точки	банк	номер точки	банк
0	0	8	2	16	4	24	6
1	2	9	4	17	6	25	0
2	4	10	6	18	0	26	2
3	6	11	0	19	2	27	4
4	1	12	3	20	5	28	7
5	3	13	5	21	7	29	1
6	5	14	7	22	1	30	3
7	7	15	1	23	3	31	5

Приложение 3

Сравнение предложенного метода адресации с аналогами

В данной таблице приводится сравнение предложенной стратегии с известными реализациями. Для сравнения приводится 2 разбиения: с оптимальными основаниями и реализуемое рассматриваемыми методами.

	[5]		[2]		[4]		данная работа	
	такты	основания	такты	основания	такты	основания	такты	основания
FFT 512	783	2|4	768	2|4	640	2|4	640	2|4
	--	--	192	8	--	--	192	8
FFT 1024	1305	4	1280	4	1280	4	1280	4
	--	--	896	2|8	--	--	512	2|8
FFT 2048	3609	2|4	3584	2|4	3072	2|4	3072	2|4
	--	--	1280	4|8	--	--	1024	4|8
FFT 4096	6174	4	6144	4	6144	4	6144	4
	--	--	2048	8	--	--	2048	8

Литература

[1] Lihong Jia; Yonghong Gao; Tenhunen, H.; , "A pipelined shared-memory architecture for FFT processors," *Circuits and Systems, 1999. 42nd Midwest Symposium on* , vol.2, no., pp.804-807 vol. 2, 1999
[2] Johnson, L.G.; , "Conflict free memory addressing for dedicated FFT hardware," *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on* , vol.39, no.5, pp.312-316, May 1992
[3] Chen-Fong Hsiao; Yuan Chen; Chen-Yi Lee; , "A Generalized Mixed-Radix Algorithm for Memory-Based FFT Processors," *Circuits and Systems II: Express Briefs, IEEE Transactions on* , vol.57, no.1, pp.26-30, Jan. 2010
[4] Jo, B.G.; Sunwoo, M.H.; , "New continuous-flow mixed-radix (CFMR) FFT Processor using novel in-place strategy," *Circuits and Systems I: Regular Papers, IEEE Transactions on* , vol.52, no.5, pp. 911-919, May 2005
[5] Jacobson, A.T.; Truong, D.N.; Baas, B.M.; , "The design of a reconfigurable continuous-flow mixed-radix FFT processor," *Circuits and Systems, 2009. ISCAS 2009. IEEE International Symposium on* , vol., no., pp.1133-1136, 24-27 May 2009