

среды и модели, заложенной в расчет модельного волнового поля [2].

Модель геологического разреза, созданная для расчета сейсмического волнового поля, должна содержать следующую информацию:

- данные о положении и геометрических свойствах отражающих границ;
- скорости продольной и поперечной волн;
- плотности пород;
- форма сейсмического сигнала, возбуждаемого источником.

На сегодня известно несколько методов моделирования волновых полей и интерпретации дан-

ных в условиях вертикальных скважин. Для случая наклонных скважин и скважин имеющих значительное искривление необходимо развитие и обобщение известных моделей.

Литература

1. Боганик Г.Н. Сейсморазведка: учебник для вузов / И.И. Гурвич. – Тверь: АИС, 2006. – 744 с.
2. Гальперин Е.И. вертикальное сейсмическое профелирование. – М.: Недра, 1971. – 263 с
3. Шевченко А.А. Скважинная сейсморазведка. – М: РГУ нефти и газа, 2002. – 129 с.

ВЫЧИСЛЕНИЕ БПФ НА ПАРАЛЛЕЛЬНОЙ АРХИТЕКТУРЕ С РАСПАРАЛЛЕЛИВАНИЕМ ОПЕРАЦИИ «БАБОЧКА»

Черемнов А.Г.

Научный руководитель: Аврамчук В.С.
Томский политехнический университет
634050, Россия, г. Томск, пр-т Ленина, 30
E-mail: 8xandr@gmail.com

В настоящее время постоянно увеличиваются размеры обрабатываемой информации, а увеличение её объёмов, в свою очередь, приводит к повышенным требованиям к быстродействию компонентов ЭВМ, что влечёт за собой развитие специализированных подходов эффективного использования аппаратных ресурсов ЭВМ.

Разработка узкоспециализированного архитектурного решения для учёта аппаратных особенностей используемого вычислителя и системы в целом является приоритетной задачей. В многопроцессорных системах роль вычислителя играет многоядерные процессоры. Добиться прироста производительности вычислительных операций можно при помощи параллельного выполнения этих операций.

В наши дни корреляционный анализ сигналов используется в акустике, физике, экономике, биологии, геологии и в других областях науки и техники. Быструю корреляцию применяют при обработке больших массивов данных или при необходимости обработки в режиме реального времени. Использование такого подхода основано на применении теоремы о корреляции и быстрого преобразования Фурье (БПФ) [1].

Увеличение числа операций быстрого преобразования Фурье является основным недостатком такого подхода. На основании этого, применение подобного метода в однопроцессорных системах реального времени невозможно.

Алгоритм Кули-Тьюки является наиболее простым в понимании и реализации способом вычисления БПФ [2]. Не смотря на широкую известность алгоритмов БПФ и огромную проработку в этом направлении, каждый год создаются новые реализации БПФ. Это легко объясняется экспоненциальным развитием вычислительной техники, созданием новых архитектур, направленных на

ускорение вычислительных операций. Создавая такие алгоритмы, разработчикам программного обеспечения удаётся добиться максимального быстродействия, компактности и эффективности кода.

Первоначально повышения эффективности вычисления БПФ можно добиться предварительной обработкой входных данных – изменением порядка следования бит в двоичном представлении числа на противоположный (операция бит-реверсирования). Стоит отметить, что преобразованию подвергаются только индексы элементов входных данных, численные значения остаются без изменения.

Пример выполнения бит-реверсирования для 8-ми элементов приведён на рисунке 1.

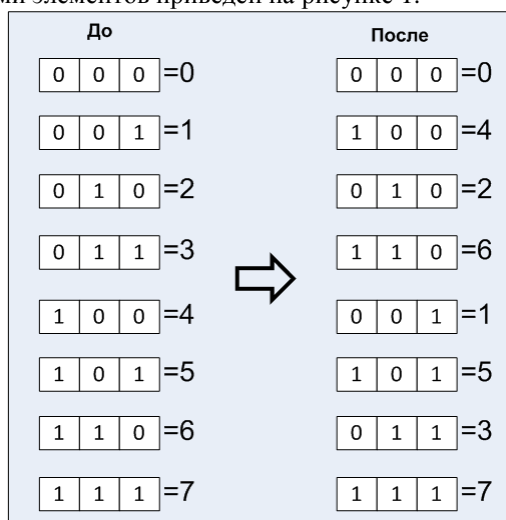


Рис. 1. Пример обработки входных данных для 8-ми элементов

После обработки входных данных производится расчёт БПФ. Разбиение исходной анализируе-

мой последовательности на две равные по размеру части позволяет добиться дополнительной минимизации вычислительных операций (рис. 1). При этом количество операций сократится в два раза [3].

Отдельно отметим, что последующее разбиение полученных последовательностей можно проводить до тех пор, пока число отсчётов в анализируемой выборке кратно 2. При $n=8$ возможное разбиение представлено на рисунке 2 [4].

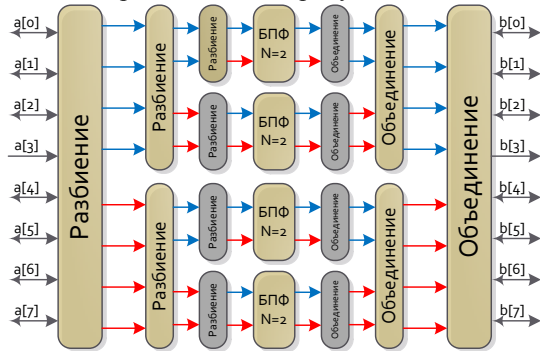


Рис. 2. Разбиение и объединение последовательности при $N=8$

В данной работе использована кроссплатформенная библиотека Intel® ТВВ в качестве инструмента разработки многопоточного приложения. На её основе в графической среде разработки Microsoft Visual Studio 2012 создано программное обеспечение расчета частотно-временной корреляционной функции [5, 6]. Определение времени выполнения операции бит-реверсирования и «бабочка» осуществлено в Visual Studio Team System Profiler, который входит в состав среды разработки Visual Studio Professional [6]. С его помощью установлено, что время, затраченное на операцию бит-реверсирования составляет не более 5% от общего времени выполнения вычисления БПФ. Для реализации параллельного вычисления БПФ использован шаблон библиотеки Intel ТВВ `tbv::parallel_for`. В него входит параметр указывающий степень детализации задачи – GrainSize. Значение `Grainsize > 1`, означает, что в этом случае выполняется несколько различных задач (БПФ) параллельно. Если `Grainsize = 1`, то распараллеливание происходит в рамках одной задачи.

Экспериментальные исследования проведены на трех процессорах фирмы Intel: Core 2 Quad 6700, Xeon® 5160 и Core i5 750. Параметр «Grainsize» варьировался в пределах от 1 до 262144.

В качестве примера в таблице 1 приведены оптимальное значение Grainsize и временные результаты вычисления частотно-временной корреляционной функции на ЦП Core i5-750.

Таблица 1. Результаты вычисления частотно-временной корреляционной функции (1000 преобразований Фурье)

Intel Core i5 750 – 4 вычислительных ядра,
--

компилятор Visual C++ 2012				
Размер выборки	Время выполнения последовательного алгоритма t_1 , мс	Grain Size	Время выполнения параллельного алгоритма t_2 , мс	t_1/t_2
4	0,166	4	0,403	0,412
8	0,531	8	0,810	0,656
16	1,479	16	1,972	0,75
32	3,894	32	4,105	0,949
64	8,705	64	9,480	0,918
128	21,705	128	21,451	1,012
256	46,239	1	45,795	1,01
512	108,680	1	79,228	1,372
1024	244,298	1	148,798	1,642
2048	534,201	1	280,050	1,908
4096	1168,978	1	548,220	2,132
8192	2576,814	1	1115,220	2,311
16384	5571,835	1	2292,142	2,431
32768	11721,840	1	4760,181	2,462
65536	24751,892	1	9930,143	2,493
131072	53233,938	1	20811,387	2,558
262144	114423,045	1	43936,744	2,604

Графики зависимости относительного времени выполнения БПФ для различных процессоров приведены на рисунке 3.

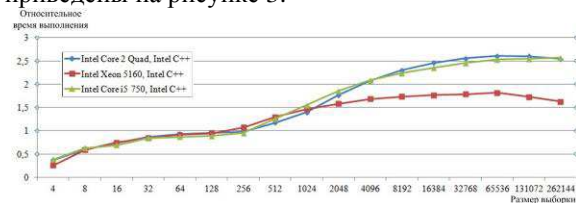


Рис. 3. Зависимость времени выполнения последовательного БПФ к БПФ с распараллеленным циклом, для различных выборок

По данным графиками видно, что время вычисления БПФ с распараллеленным внешним циклом для малых выборок (до 256 элементов) для всех компиляторов больше или равно времени последовательного алгоритма БПФ. Это объясняется затрачиваемым ЦП временем на создание потоков.

Возможные пути повышения эффективности расчета частотно-временной корреляционной функции также рассмотрены в [7-9].

Заключение

Исходя из анализа полученных результатов, можно сделать следующий вывод: использование вычислительных мощностей CPU при вычислении частотно-временной автокорреляционной функции позволяет значительно сократить время вычислений.

Литература

1. Айфичер Э.С., Джервис Б.У. Цифровая обработка сигналов: практический подход. 2-е изд. – М.: Вильямс, 2008. – 992 с.
2. Блейхут Р. Быстрые алгоритмы цифровой обработки сигналов. — М.: Мир, 1989. — 448 с.

3. Теория и практика цифровой обработки сигналов. URL: <http://www.dsplib.ru/index.html> (Дата обращения: 17.09.2013)

4. Сайт компании Microsoft. <http://msdn.microsoft.com/ru-ru/> (Дата обращения: 17.09.2013)

5. Аврамчук В. С. Определение наличия гармонических составляющих и их частот в дискретных сигналах на основе автокорреляционной функции // Известия Томского политехнического университета. - 2012 - Т. 321 – №. 5 – С. 113–116.

6. Аврамчук В.С., Лунева Е.Е., Карабин В.Л. Визуализация частотно-временной корреляционной функции на основе технологии OpenGL // Системы управления и информационные технологии, №1.1(51), 2013. – С. 110–113.

7. Лунева Е.Е., Аврамчук В.С. Анализ путей повышения эффективности расчетов частотно-временной корреляционной функции // Известия Томского политехнического университета. – 2013. – Т. 322 – № 5. – С. 33–36.

8. Лунева Е.Е., Аврамчук В.С. Технологии параллельных вычислений на многопроцессорных системах в задачах корреляционного анализа // Системы управления и информационные технологии, №3.1(49), 2012. – С. 156–159.

9. Аврамчук В.С., Лунева Е.Е., Черемнов А.Г. Способы повышения эффективности вычисления быстрого преобразования Фурье//Интернет-журнал «Наукоедение». 2013 №3 [Электронный ресурс].-М. 2013. – Режим доступа: <http://naukovedenie.ru/PDF/16tvn313.pdf>, свободный – Загл. с экрана.

ВЫЧИСЛЕНИЕ БПФ НА ПАРАЛЛЕЛЬНОЙ АРХИТЕКТУРЕ С РАСПАРАЛЛЕЛИВАНИЕМ ОПЕРАЦИИ «БАБОЧКА» С ИСПОЛЬЗОВАНИЕМ АДАПТИВНОГО РАСЧЁТА СТЕПЕНИ ДЕТАЛИЗАЦИИ

Черемнов А.Г.

Научный руководитель: Аврамчук В.С.

Томский политехнический университет

634050, Россия, г. Томск, пр-т Ленина, 30

E-mail: 8xandr@gmail.com

Тактовая частота работы центрального процессора, а также архитектура конвейерной обработки команд являются определяющими факторами быстродействия ЭВМ. В последнее десятилетие из-за фундаментальных физических ограничений при производстве больших интегральных схем (СБИС) рост тактовой частоты замедлился. Отдельно отметим, что тепловыделение и энергопотребление пропорционально зависит от тактовой частоты. Например, при увеличении тактовой частоты в два раза, тепловыделение увеличивается в 16 раз [1]. В настоящее время рост производительности центральных процессоров достигается за счёт количества вычислительных ядер.

При использовании нескольких слабых процессоров возможно получить большую производительность по сравнению с одним мощным [2].

В наши дни цифровая обработка сигналов применяется практически во всех отраслях науки и техники. Разложение в ряд Фурье является одной из распространённых задач. Математический аппарат преобразования Фурье хорошо известен и описан в литературе [3].

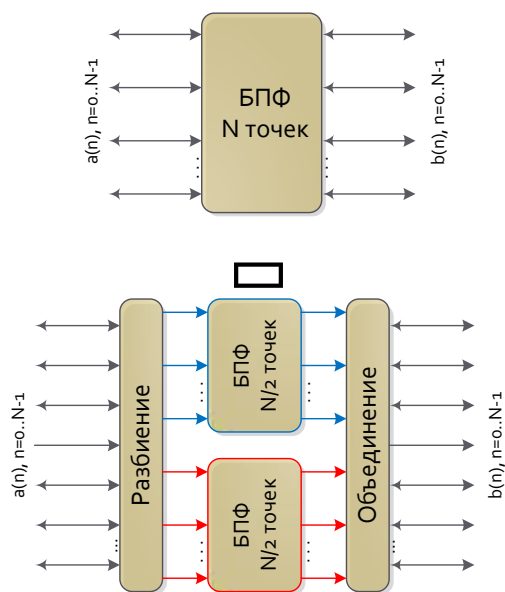


Рис. 1. Замена N-точного БПФ двумя N/2-точными БПФ

Одним из возможных способов вычисления быстрого преобразования Фурье (БПФ) является алгоритм Кули-Тьюки [4]. Этот алгоритм прост в понимании и легко поддается распараллеливанию.

При использовании разбиении исходной анализируемой последовательности на две, более коротких достигается дополнительная минимизация вычислений (рис. 1). При этом количество операций сократится в два раза [3]. Последующее разбиение полученных последовательностей возмож-