

МИНИСТЕРСТВО ВЫСШЕГО И СРЕДНЕГО СПЕЦИАЛЬНОГО ОБРАЗОВАНИЯ УССР
ВИННИЦКИЙ ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

На правах рукописи

Волынец Виктор Иванович

УДК 519.67; 681.32

РАЗРАБОТКА ЭФФЕКТИВНЫХ АЛГОРИТМОВ ОБРАБОТКИ СИГНАЛОВ
ДЛЯ МИКРОПРОЦЕССОРНЫХ АНАЛИЗАТОРОВ СПЕКТРА

Специальность 05.11.16 – информационно-измерительные системы
(в науке и промышленности)

ДИССЕРТАЦИЯ

на соискание ученой степени кандидата технических наук

Научный руководитель
д.т.н., проф. Осадчук В.С.

Винница – 1991

С О Д Е Р Ж А Н И Е

СПИСОК ПРИНЯТЫХ ОСНОВНЫХ СОКРАЩЕНИЙ.....	4
ВВЕДЕНИЕ.....	5
ГЛАВА 1. АНАЛИЗ МЕТОДОВ И СРЕДСТВ ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ (ЦОС).....	13
1.1. Дискретные преобразования Фурье и Хартли - математическая основа ИИС ЦОС.....	13
1.2. Алгоритмы выполнения основных вычислитель- ных операций ЦОС.....	17
1.3. Алгоритмы быстрого вычисления дискретных преобразований Фурье и Хартли.....	20
1.4. Точность алгоритмов быстрого преобразования Фурье и Хартли.....	39
1.5. Микропроцессорная реализация ИИС ЦОС.....	44
Выводы.....	48
ГЛАВА 2. РАЗРАБОТКА ЭФФЕКТИВНЫХ АЛГОРИТМОВ ВЫЧИСЛЕНИЯ ДИСКРЕТНОГО ПРЕОБРАЗОВАНИЯ ХАРТЛИ И ЭНЕРГЕ- ТИЧЕСКОГО СПЕКТРА.....	50
2.1. Алгоритмы быстрого преобразования Хартли по основанию 4.....	50
2.2. Алгоритмы быстрого преобразования Хартли по основанию 8.....	64
2.3. Алгоритмы быстрого преобразования Хартли с расщепленным основанием.....	83
2.4. Сравнительный анализ алгоритмов быстрого преобразования Хартли по количеству арифме- тических операций и объему требуемой памяти.....	91
2.5. Рекуррентные алгоритмы вычисления дискретно- го преобразования Хартли и энергетического	

спектра.....	95
Выводы.....	103
ГЛАВА 3. ИССЛЕДОВАНИЕ ТОЧНОСТИ АЛГОРИТМОВ БЫСТРОГО ПРЕ- ОБРАЗОВАНИЯ ФУРЬЕ И ХАРТЛИ С ФИКСИРОВАННОЙ ЗАПЯТОЙ.....	105
3.1. Основные положения исследования точности алго- ритмов быстрого преобразования Фурье.....	105
3.2. Исследование точности алгоритмов быстрого пре- образования Фурье.....	108
3.3. Исследование точности алгоритмов быстрого пре- образования Хартли.....	122
3.4. Сравнительный анализ точности алгоритмов быст- рого преобразования Фурье и Хартли.....	126
Выводы.....	130
ГЛАВА 4. АППАРАТУРНЫЕ И ПРОГРАММНЫЕ СРЕДСТВА МИКРОПРО- ЦЕССОРНЫХ АНАЛИЗАТОРОВ СПЕКТРА.....	132
4.1. Программная реализация алгоритмов быстрого преобразования Фурье и Хартли.....	132
4.2. Микропроцессорный анализатор спектра на базе микропроцессора КР580ИК80А.....	138
4.2.1. Модуль ввода исходной информации.....	141
4.2.2. Модуль вычислителя энергетического спектра...	147
4.2.3. Модуль вывода результатов обработки.....	148
Выводы.....	153
ЗАКЛЮЧЕНИЕ.....	155
ЛИТЕРАТУРА.....	157
ПРИЛОЖЕНИЯ.....	170

СПИСОК ПРИНЯТЫХ ОСНОВНЫХ СОКРАЩЕНИЙ

- АЦП - аналого-цифровой преобразователь
БИС - большая интегральная схема
БПФ - быстрое преобразование Фурье
БПФД - быстрое преобразование Фурье вещественной последовательности
БПХ - быстрое преобразование Хартли
ДОП - дискретные ортогональные преобразования
ДПФ - дискретное преобразование Фурье
ДПХ - дискретное преобразование Хартли
МП - микропроцессор
МПК - микропроцессорный комплект
ПДП - прямой доступ к памяти
СВИС - сверхбольшая интегральная схема
СКЗ - среднеквадратическое значение
ТЧП - теоретико-числовое преобразование
ЦАП - цифро-аналоговый преобразователь
ЦПС - цифровой процессор сигналов
ЦОС - цифровая обработка сигналов
ЭВМ - электронно-вычислительная машина
ЭЛТ - электронно-лучевая трубка

В В Е Д Е Н И Е

Актуальность проблемы. Широкая автоматизация процессов производства и научных исследований является одним из основных направлений научно-технического прогресса [1, 2]. С этой целью разрабатываются разнообразные информационно-измерительные системы (ИИС) и комплексы на базе средств вычислительной техники, выполняющие задачи сбора, хранения, обработки и передачи информации [1 - 7]. Важное место среди них занимают ИИС, предназначенные для решения задач цифровой обработки сигналов (ЦОС) [8 - 10], к числу которых относятся спектрально-корреляционный анализ, цифровая фильтрация, кодирование и сжатие информации, классификация и распознавание образов [11 - 15]. Данные системы находят применение во многих областях науки и техники, таких как управление, связь, радио- и гидролокация, акустика, астрономия, сейсмология, биомедицина, вибродиагностика, цифровая голография, обработка изображений [16 - 22].

Эффективность ИИС ЦОС во многом определяется эффективностью используемых алгоритмов ЦОС, многие из которых базируются на разнообразных дискретных ортогональных преобразованиях (ДОП) [22 - 26], наибольшее распространение среди которых находит дискретное преобразование Фурье (ДПФ) [14, 15].

В последнее время большое внимание уделяется дискретному преобразованию Хартли (ДПХ) [35, 36], тесно связанному с ДПФ. В отличие от ДПФ ДПХ имеет ряд преимуществ, к числу которых относятся отсутствие комплексной арифметики для его вычисления и тождественность его прямого и обратного преобразований с точностью до фиксированного множителя.

Однако, если для вычисления ДПФ разработано большое количество быстрых алгоритмов вычисления [15, 30, 31, 37 - 42],

получивших название алгоритмов быстрого преобразования Фурье (БПФ), то для вычисления ДПХ разработаны и исследованы лишь эффективные алгоритмы быстрого преобразования Хартли (БПХ) по основанию 2 [43, 44]. В связи с этим представляется целесообразным разработать и исследовать эффективные алгоритмы БПХ по другим более высоким основаниям, в частности основаниям 4, 8 и с расщепленным основанием.

Важным критерием выбора того или иного алгоритма для практической реализации является точность вычислений, обеспечиваемая данным алгоритмом. Несмотря на большое количество публикаций по этой теме [14, 15, 36, 45 - 51], анализ вычислительных ошибок проводился лишь для некоторых частных случаев при реализации алгоритмов БПФ по основаниям 2, 4 и алгоритмов БПХ по основанию 2.

Особым случаем является вычисление ДПФ и ДПХ на скачущих (скользящих) интервалах. Известны рекуррентные алгоритмы вычисления ДПФ на скачущих интервалах [52, 53], а также рекуррентные алгоритмы вычисления энергетического спектра на базе ДПХ на скользящих интервалах [54]. Однако отсутствуют рекуррентные алгоритмы вычисления ДПХ и энергетического спектра на базе ДПФ и ДПХ на скачущих (скользящих) интервалах.

Наряду с разработкой и исследованием эффективных алгоритмов ЦОС большое внимание должно быть уделено разработке ИИС ЦОС на их основе, для реализации которых используется современная элементная база. Применение микропроцессоров для разработки ИИС ЦОС [55 - 57] позволило существенно расширить область применения последних. Однако использование известных микропроцессорных комплектов ставит ряд проблем, к числу которых относится разработка устройств, выполняющих обмен информации между составными частями ИИС ЦОС и обеспечивающих максимальную производительность

системы в целом. Кроме того, важное место отводится эффективной программной реализации алгоритмов ЦОС, учитывающей особенности используемых алгоритмов и микропроцессоров.

Разработка и исследование эффективных алгоритмов, программ и микропроцессорных структур ИИС ЦОС позволит разработать высокопроизводительные ИИС ЦОС, в частности, автономные, функционально законченные микропроцессорные анализаторы спектра, предназначенные для работы в реальном масштабе времени или близком к нему.

Целью данной работы является разработка и исследование эффективных алгоритмов, программ и структурных схем микропроцессорных анализаторов спектра.

Для достижения поставленной цели необходимо решить следующие задачи:

- разработать эффективные алгоритмы БПХ по основаниям 4, 8 и с расщепленным основанием с прореживанием по времени и частоте и провести сравнительный анализ по количеству арифметических операций и объему требуемой памяти;
- разработать рекуррентные алгоритмы вычисления ДПХ и энергетического спектра на скачущих (скользящих) интервалах, провести сравнительный анализ по количеству арифметических операций и объему требуемой памяти;
- исследовать точность алгоритмов БПФ и БПХ с фиксированной запятой и провести их сравнительный анализ;
- разработать эффективные программные реализации алгоритмов ЦОС, учитывающие особенности используемых алгоритмов и оборудования;
- разработать структурные и функциональные схемы устройств, выполняющих обмен информации между составными частями микропроцессорных ИИС ЦОС, которые обеспечивают максимальную производительность последних;

- реализовать структурные и функциональные схемы устройств обмена информации, а также эффективные программные реализации алгоритмов ЦОС в микропроцессорных анализаторах спектра.

Методы исследований базируются на использовании теории ЦОС, аппарата матричной алгебры, теории вероятностей и математической статистики, теоретических основах проектирования цифровой вычислительной техники.

Научная новизна. В результате проведенных исследований получены следующие результаты:

- разработаны эффективные алгоритмы БПХ по основаниям 4, 8 и с расщепленным основанием с прореживанием по времени и частоте, которые по комплексному критерию превосходят аналогичные алгоритмы БПФ для обработки вещественных последовательностей в 3 раза, а алгоритмы БПФ для обработки комплексных последовательностей - в 4,1 раза;

- разработаны рекуррентные алгоритмы вычисления ДПХ и энергетического спектра, которые могут быть использованы для эффективного вычисления ДПХ и энергетического спектра на скачущих (скользящих) интервалах;

- исследована точность алгоритмов БПФ и БПХ по основаниям 2, 4, 8 и с расщепленным основанием с прореживанием по времени и частоте с фиксированной запятой для случаев входного и автоматического масштабирования, представления чисел в прямом, обратном и дополнительном коде, аппроксимации результатов умножения путем округления и усечения.

Практическая ценность работы заключается в следующем:

- разработаны эффективные программные реализации алгоритмов ЦОС, в частности, программы БПФ и БПХ на языке ассемблера микропроцессора КР580ИК80А с временем выполнения для 1024-точечной последовательности соответственно $2,6 \div 3$ и $1,25 \div 1,5$ сек;

- разработаны структурные и функциональные схемы устройств

прозрачного прямого доступа к памяти, отображения информации на экране электронно-лучевой трубки, сопряжения с периферийными устройствами, которые входят в состав ИИС ЦОС и позволяют повысить их производительность в 2 раза;

- на основе структурных и функциональных схем обмена информации, а также эффективных программных реализаций алгоритмов ЦОС разработан автономный микропроцессорный анализатор спектра, позволяющий проводить оперативный спектральный анализ сигналов.

Внедрение результатов работы. Микропроцессорный анализатор спектра на базе микропроцессора КР580ИК80А внедрен в ИРЭ АН УССР (г.Харьков). Годовой экономический эффект от внедрения составил 29,700 тыс.руб.

Материалы диссертационной работы докладывались и обсуждались на следующих конференциях и семинарах:

- XII Всесоюзной научной конференции по микроэлектронике, г.Тбилиси, 1987 г.;
- Республиканской научно-технической конференции "Информатика и автоматизация в регионе", г.Винница, 1988 г.;
- 3-й Всесоюзной школе "Проектирование автоматизированных систем контроля и управления сложными объектами", г.Харьков, 1988 г.;
- научно-техническом семинаре "Применение микропроцессоров, микро- и персональных ЭВМ", г.Челябинск, 1989 г.;
- Международной молодежной научно-технической конференции "Актуальные проблемы информатики, управления, радиоэлектроники и лазерной техники", г.Пушкино Московской области, 1989 г.;
- второй Всесоюзной конференции молодых ученых и специалистов с международным участием "Контроль, управление и автоматизация в современном производстве", г.Минск, 1990 г."

Вклад автора в разработку проблемы. Основные научно-техни-

ческие идеи, изложенные в диссертации, разработаны автором под руководством д.т.н., проф. Осадчука В.С.

Автором разработаны алгоритмы БПХ по основаниям 4, 8 и с расщепленным основанием с прореживанием по времени и частоте; рекуррентные алгоритмы вычисления ДПХ и энергетического спектра на скачущих (скользящих) интервалах. Исследована точность алгоритмов БПФ и БПХ с фиксированной запятой. Разработаны эффективные программные реализации алгоритмов ЦОС. Разработаны структурные и функциональные схемы устройств, выполняющих обмен информации в ИИС ЦОС, которые обеспечивают максимальную производительность последних.

При личном участии автора в хозяйственной научно-исследовательской работе, выполненной кафедрой микроэлектроники Винницкого политехнического института, разработан и внедрен в ИРЭ АН УССР (г. Харьков) микропроцессорный анализатор спектра на базе микропроцессора КР580МК80А, предназначенный для оперативного спектрального анализа сигналов.

Публикации. По результатам проведенных исследований опубликовано 17 печатных работ, в том числе 4 авторских свидетельства.

Структура работы. Диссертация состоит из введения, четырех глав, заключения, перечня использованных источников из 125 наименований и приложений. Объем работы составляет 121 страниц основного текста, 32 рисунков, 13 таблиц.

Во введении обоснована актуальность работы, сформулирована цель и задачи исследования, отражена научная новизна и практическая ценность диссертации, приведена ее структура.

В первой главе рассмотрены ДПФ и ДПХ и их использование в ИИС ЦОС. Проведен анализ алгоритмов выполнения основных вычислительных операций ЦОС. Рассмотрены наиболее эффективные алгоритмы БПФ и проведен их сравнительный анализ. Показана целесообразность

использования алгоритмов БПХ. Рассмотрены вопросы определения точности алгоритмов БПФ и БПХ. Представлены известные рекуррентные алгоритмы вычисления ДПФ и энергетического спектра. Проведен обзор путей микропроцессорной реализации ИИС ЦОС.

Во второй главе разработаны алгоритмы БПХ по основаниям 4,8 и с расщепленным основанием с прореживанием по времени и частоте. Проведен сравнительный анализ данных алгоритмов по количеству арифметических операций и объему требуемой памяти. Разработаны рекуррентные алгоритмы вычисления ДПХ и энергетического спектра и проведен их сравнительный анализ при вычислении на скользящих интервалах.

В третьей главе на основе статистического метода проведен анализ вычислительных ошибок алгоритмов БПФ и БПХ по основаниям 2, 4, 8 и с расщепленным основанием с прореживанием по времени и частоте с фиксированной запятой для случаев входного и автоматического масштабирования, представления чисел в прямом, обратном и дополнительном коде, аппроксимации результатов умножения путем округления и усечения. Проведен сравнительный анализ точности алгоритмов БПФ и БПХ. Рассмотрены вопросы повышения точности и связи динамического диапазона входного сигнала с точностью алгоритмов.

В четвертой главе описана организация выполнения алгоритмов БПФ и БПХ и разработаны их эффективные программные реализации. Описан автономный микропроцессорный анализатор спектра на базе микропроцессора КР580ИК80А и рассмотрены структурные и функциональные схемы устройств, выполняющих обмен информации в его составе.

В заключении изложены основные результаты проведенных исследований.

В диссертации защищаются:

- эффективные алгоритмы БПХ по основаниям 4, 8 и с расщепленным основанием с прореживанием по времени и частоте;
- рекуррентные алгоритмы вычисления ДПХ и энергетического спектра;
- точностные характеристики алгоритмов БПФ и БПХ с фиксированной запятой;
- структурные и функциональные схемы устройств, входящих в состав микропроцессорных анализаторов спектра.

ГЛАВА I. АНАЛИЗ МЕТОДОВ И СРЕДСТВ ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ (ЦОС)

1.1. Дискретные преобразования Фурье и Хартли - математическая основа ИИС ЦОС

В основе построения ИИС ЦОС лежит функциональная схема ЦОС [10], представленная на рис.1.1.

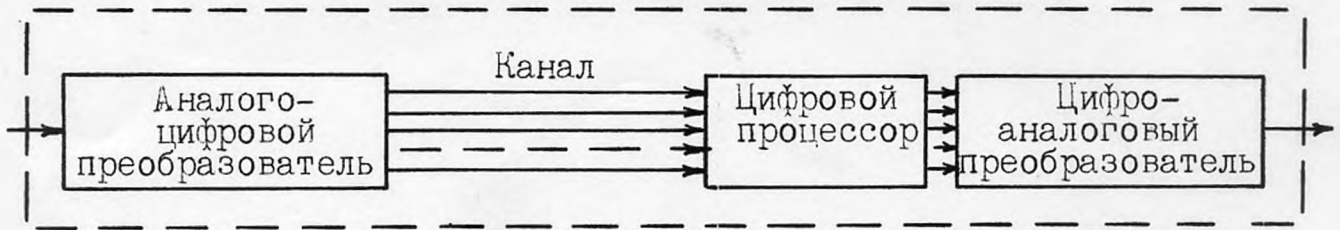


Рис.1.1. Функциональная схема ЦОС

При прохождении сигнала со входа на выход цифрового процессора над ним выполняется некоторое основное преобразование, которое реализуется алгоритмически.

В настоящее время для решения многих задач ЦОС широко используется дискретное преобразование Фурье (ДПФ), которое задается парой соотношений [14, 15]

$$F(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk}, \quad k = \overline{0, N-1}, \quad (1.1)$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} F(k) W_N^{-nk}, \quad n = \overline{0, N-1}, \quad (1.2)$$

где $x(n)$ - последовательность во временной области (вещественная или комплексная), $F(k)$ - последовательность в частотной области Фурье, $W_N = \exp(-j2\pi/N)$ и $j = \sqrt{-1}$.

Соотношение (1.1) называется прямым ДПФ, а соотношение (1.2) - обратным ДПФ. Основные свойства ДПФ (линейность, циклический сдвиг и др.) подробно описаны в [14, 15].

В последнее время большое внимание специалистов привлекает дискретное преобразование Хартли (ДПХ), определяемое парой соотношений [35, 36]

$$H(k) = \sum_{n=0}^{N-1} x(n) \operatorname{cas}(2\pi nk/N), \quad k = \overline{0, N-1}, \quad (1.3)$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} H(k) \operatorname{cas}(2\pi nk/N), \quad n = \overline{0, N-1}, \quad (1.4)$$

где $x(n)$ - вещественная последовательность во временной области, $H(k)$ - вещественная последовательность в частотной области Хартли, $\operatorname{cas}(\cdot) = \cos(\cdot) + \sin(\cdot)$.

В отличие от ДПФ ДПХ носит исключительно вещественный характер, вследствие чего для его вычисления не требуется использование комплексной арифметики. Кроме того, прямое и обратное ДПХ, определяемые соответственно выражениями (1.3) и (1.4), совпадают друг с другом с точностью до постоянного множителя. ДПХ обладает основными свойствами (линейность, циклический сдвиг и др.), аналогичными ДПФ. Их описание можно найти в [35, 36].

Из соотношений (1.1) и (1.3) легко установить связь между ДПФ и ДПХ в случае, когда последовательность $x(n)$ является вещественной. Эта связь задается следующим образом:

$$\begin{aligned} \operatorname{Re} [F(k)] &= [H(k) + H(N-k)] / 2, \\ \operatorname{Im} [F(k)] &= [H(N-k) - H(k)] / 2, \end{aligned} \quad (1.5)$$

где $\operatorname{Re} [F(k)]$ и $\operatorname{Im} [F(k)]$ — соответственно действительная и мнимая части $F(k)$.

Одной из основных задач спектрального анализа является нахождение энергетического спектра [26], который определяется как

$$|F(k)|^2 = \{\operatorname{Re} [F(k)]\}^2 + \{\operatorname{Im} [F(k)]\}^2. \quad (1.6)$$

С учетом (1.5) энергетический спектр можно вычислить следующим образом:

$$|F(k)|^2 = [H^2(k) + H^2(N-k)] / 2. \quad (1.7)$$

Основной задачей цифровой фильтрации является задача вычисления линейной свертки, которая адекватно описывает цифровой фильтр с конечной импульсной характеристикой. С понятием линейной свертки тесно связано понятие линейной корреляции. На практике линейную свертку (линейную корреляцию) можно вычислить через циклическую свертку (циклическую корреляцию). Для двух заданных последовательностей $x(n)$ и $g(n)$ циклическая свертка определяется как новая последовательность [58]

$$s(m) = \sum_{n=0}^{N-1} g \langle m-n \rangle x(n), \quad m = \overline{0, N-1}, \quad (1.8)$$

где $\langle m-n \rangle = m-n$ по модулю N .

Аналогично циклическая корреляция определяется как

$$r(m) = \sum_{n=0}^{N-1} g \langle m+n \rangle x(n), \quad m = \overline{0, N-1}. \quad (1.9)$$

Циклические свертки и корреляцию можно вычислить через ДФ, учитывая их свойства [15]:

$$\begin{aligned} S(k) &= F(k) \cdot G(k), \quad k = \overline{0, N-1}, \\ R(k) &= F^*(k) \cdot G(k), \quad k = \overline{0, N-1}, \end{aligned} \quad (1.10)$$

где $S(k)$, $R(k)$, $F(k)$, $G(k)$ - соответственно ДПФ последовательностей $S(m)$, $\varepsilon(m)$, $x(n)$, $g(n)$; $F^*(k)$ - комплексно-сопряженное ДПФ последовательности $x(n)$.

Аналогичные соотношения можно получить, используя ДПХ. Эти соотношения имеют следующий вид [36]:

$$\begin{aligned} S_H(k) &= H(k) \cdot G_{He}(k) + H(N-k) \cdot G_{Ho}(k), \quad k = \overline{0, N-1}, \\ R_H(k) &= H(N-k) \cdot G_{He}(k) + H(k) \cdot G_{Ho}(k), \quad k = \overline{0, N-1}, \end{aligned} \quad (1.11)$$

где $S_H(k)$, $R_H(k)$, $H(k)$ - соответственно ДПХ последовательностей $S(m)$, $\varepsilon(m)$, $x(n)$; $G_{He}(k)$ и $G_{Ho}(k)$ - соответственно четная и нечетная части ДПХ последовательности $g(n)$.

Таким образом, основные задачи ЦОС могут быть решены, используя ДПФ и ДПХ. Данное обстоятельство легло в основу разработки микропроцессорной системы ЦОС [59], предназначенной для определения спектра, корреляции и свертки.

Анализируя рис.1.1, можно сделать вывод, что эффективность ИИС ЦОС во многом определяется эффективностью используемых алгоритмов. Эффективность алгоритмов ЦОС можно определить по некоторому комплексному критерию

$$\mathcal{E} = \prod_i \frac{P_i \text{ баз.}}{P_i \text{ исп.}}, \quad (1.12)$$

где $P_i \text{ баз.}$ и $P_i \text{ исп.}$ - параметры, характеризующие быстродействие, точность вычислений, объем требуемой памяти и ряд других соответственно базового и используемого алгоритмов.

Кроме того, как видно из рис.1.1, эффективность ИИС ЦОС зависит от эффективности каналов связи, обеспечивающих обмен информации между составными частями ИИС. При этом одним из наиболее важных показателей эффективности каналов связи является максимальная пропускная способность, обеспечиваемая ими.

Таким образом, разработка эффективных ИИС ЦОС сводится к разработке эффективных алгоритмов ЦОС и каналов связи.

1.2. Алгоритмы выполнения основных вычислительных операций ЦОС

Сложность вычислительных алгоритмов во многом определяется количеством и видом требуемых вычислительных операций. Из анализа (1.1) – (1.4) и (1.8) – (1.9) видно, что основными вычислительными операциями являются операции умножения и сложения, которые в общем случае являются комплексными. Ввиду того, что операция умножения является существенно более сложной, чем операция сложения, рассмотрим алгоритмы ее выполнения.

Классический алгоритм заключается в том, что в результате умножения двух комплексных чисел $a+jb$ и $c+jd$ получается комплексное число

$$e+jf = (a+jb) \cdot (c+jd) = (ac-bd) + j(ad+bc). \quad (1.13)$$

Следовательно, операция комплексного умножения в данном случае требует выполнения четырех операций вещественного умножения и двух операций вещественного сложения. Данный алгоритм допускает распараллеливание выполнения операций вещественного умножения и вещественного сложения, что может ускорить его выполнение при реализации на параллельных структурах.

Другой алгоритм комплексного умножения задается следующими равенствами [60]:

$$\begin{aligned} e &= (a-b) \cdot d + a \cdot (c-d), \\ f &= (a-b) \cdot d + b \cdot (c+d). \end{aligned} \quad (1.14)$$

В такой форме алгоритм содержит три вещественных умножения и пять вещественных сложений. Если в серии комплексных умножений

величины c и d суть константы, то члены $c+d$ и $c-d$ можно вычислить заранее. Тогда для вычисления произведения комплексных чисел потребуется три вещественных умножения и три вещественных сложения. Этот алгоритм также допускает распараллеливание выполнения отдельных операций, но быстродействие его в этом случае ниже, чем для классического алгоритма. Он требует также большего объема памяти для хранения констант. Кроме того, в случае распределения величины c и d в диапазоне $|c| < 1$ и $|d| < 1$ реализация алгоритма с фиксированной запятой оказывается затруднительной, так как величины $c-d$, $c+d$ могут выходить за пределы разрядной сетки представления чисел. Для решения этой проблемы можно промасштабировать (например, сдвинуть вправо на один разряд) величины c и d , выполнять вычисления по основным выражениям, а затем полученные величины привести к правильному результату (сдвинуть влево на один разряд). Однако, в этом случае снижается точность результатов умножения за счет более низкой точности представления величин c и d .

В [61] предложен алгоритм умножения комплексных чисел, одно из которых есть $\cos \varepsilon - j \sin \varepsilon$. В этом случае в результате комплексного умножения получается комплексное число

$$e + jf = (a + jb) \cdot (\cos \varepsilon - j \sin \varepsilon) = (a \cdot \cos \varepsilon + b \cdot \sin \varepsilon) + j(b \cdot \cos \varepsilon - a \cdot \sin \varepsilon).$$

Данный результат может быть получен, используя следующий алгоритм:

$$\begin{aligned} z &= a + \operatorname{tg} \frac{\varepsilon}{2} \cdot b, \\ f &= b - z \cdot \sin \varepsilon, \quad e = z - f \cdot \operatorname{tg} \frac{\varepsilon}{2}. \end{aligned} \tag{1.15}$$

В справедливости данного алгоритма можно убедиться, если учесть, что $\operatorname{tg} \frac{\varepsilon}{2} = (1 - \cos \varepsilon) / \sin \varepsilon$. Таким образом, данный алгоритм требует три вещественных умножения и три вещественных сложения и по арифметической сложности эквивалентен предыдущему. По объему требуемой памяти для хранения констант $\operatorname{tg} \frac{\varepsilon}{2}$ и $\sin \varepsilon$ данный алгоритм эквивалентен классическому алгоритму. Недостатком данного алгоритма является невозможность какого-либо распараллеливания процесса вычисления и его реализации с фиксированной запятой, когда $|\operatorname{tg} \frac{\varepsilon}{2}| \gg 1$.

Большое внимание привлекает алгоритм умножения комплексных чисел, выполняемый в поле Галуа [62], способ возможной реализации которого представлен в [63]. Данный алгоритм имеет следующий вид:

$$\begin{aligned} A &= 1/2 (a + jb) \cdot (c + jd), \\ B &= 1/2 (a - jb) \cdot (c - jd), \\ e &= A + B, \\ f &= j \cdot (B - A), \end{aligned}$$

где все операции выполняются в кольце вычетов по некоторому модулю M . При выборе модуля необходимо соблюдать два условия. Во-первых, необходимо, чтобы в данном кольце вычетов существовал элемент $j = \sqrt{-1}$, откуда следует, что в качестве M может быть выбрано любое число вида $M = K^2 + 1$, где K — целые положительные числа. Для исключения умножения на j выбирается $M = (2^p)^2 + 1$, что приводит к замене операций умножения операциями сдвига на p разрядов влево. Во-вторых, необходимо, чтобы действительная и мнимая части комплексного произведения были меньше M . При соблюдении этих условий данный алгоритм требует два вещественных умножения и шесть вещественных сложений. Однако,

данный алгоритм оперирует с величинами, разрядность которых, например, при использовании в качестве модуля M чисел Ферма в два раза больше, чем разрядность исходных чисел. Поэтому необходимо либо наличие умножителей и сумматоров повышенной разрядности, либо ограничение динамического диапазона исходных чисел, что в свою очередь приводит к пониженной точности вычислений.

К алгоритмическому методу умножения относится также алгоритм "цифра за цифрой" и его модификации [64 - 66]. Однако, он составляет отдельную область исследований и здесь не рассматривается.

Каждый из описанных выше алгоритмов разбивается на множество других, зависящих от системы счисления, формы представления и кодирования чисел [67 - 69]. Выбор этих алгоритмов должен производиться в каждом конкретном случае, исходя из множества противоречивых требований.

1.3. Алгоритмы быстрого вычисления дискретных преобразований Фурье и Хартли

Анализ (1.1) показывает, что для вычисления ДПФ требуется в общем случае порядка N^2 комплексных умножений и $N \cdot (N-1)$ комплексных сложений. Такой объём вычислений долгое время затруднял применение ДПФ для решения многих задач и лишь с появлением алгоритма быстрого преобразования Фурье (БПФ) Кули - Тьюки [70] в 1965 г. интерес к нему резко возрос. Данная работа стимулировала разработку большого разнообразия алгоритмов БПФ, лучшие из которых описаны в [14, 15, 37 - 42, 71, 72].

В основе алгоритмов БПФ лежат две стратегии. Одна из них состоит в сведении ДПФ к свёртке, для вычисления которой разработаны эффективные алгоритмы [37], другая - в переходе от одномерного ДПФ к двумерному, что приводит к снижению вычислительных

затрат. Во втором случае длина преобразования N представляется в виде произведения некоторого числа сомножителей. Если сомножители взаимно-просты, данные алгоритмы носят название алгоритмов Гуда - Томаса, разновидностью которых являются алгоритмы Винограда [37]. В противном случае имеют место алгоритмы Кули - Тьюки.

Выбор алгоритма БПФ следует из компромисса между многими критериями, такими как объём вычислений, объём требуемой памяти, точность, повторяемость, модульность, расположение данных по ходу вычислений, регулярность, лёгкость и др., порядок важности которых зависит от особенностей применения, размерности ДПФ и используемого оборудования. В [71] сделан вывод, что первенствовать будут критерии, относящиеся к лёгкости применения алгоритмов, которым наиболее полно удовлетворяют алгоритмы Кули - Тьюки.

В случае представления $N = \varepsilon^m$ имеет место алгоритм БПФ Кули - Тьюки с фиксированным основанием ε ; если $N = \varepsilon_1 \cdot \varepsilon_2 \cdot \dots \cdot \varepsilon_k$ - алгоритм со смешанным основанием [73]. В зависимости от порядка вычислений алгоритмы делятся на алгоритмы с прореживанием по времени и частоте.

Наибольшее распространение получили алгоритмы БПФ с фиксированным основанием, равным 2, 4 и 8. Представим результаты сравнительного анализа данных алгоритмов по количеству арифметических операций и объёму требуемой памяти [74].

Пусть $M_R(N)$ - число вещественных умножений и $A_R(N)$ - число вещественных сложений N - точечного алгоритма БПФ. Тогда для алгоритмов БПФ по основанию 2 (БПФ-2) [70, 75]

$$M_R(N) = \frac{3}{2} N \log_2 N - 5N + 8,$$

$$A_R(N) = \frac{7}{2} N \log_2 N - 5N + 8.$$

Здесь и далее принято, что комплексное умножение содержит три вещественных умножения и три вещественных сложения, тривиаль-

ные умножения на $W_N^0 = 1$, $W_N^{N/4} = -j$ не учитываются, а умножения на $W_N^{N/8}$ и $W_N^{3N/8}$ содержат два вещественных умножения и два вещественных сложения.

Арифметическая сложность алгоритмов БПФ по основанию 4 (БПФ-4) [37] и по основанию 8 (БПФ-8) [76] определяется соответственно следующим образом:

$$M_R(N) = \frac{9}{8} N \log_2 N - \frac{43}{12} N + \frac{16}{3},$$

$$A_R(N) = \frac{25}{8} N \log_2 N - \frac{43}{12} N + \frac{16}{3}$$

и

$$M_R(N) = \frac{25}{24} N \log_2 N - \frac{25}{8} N + 4,$$

$$A_R(N) = \frac{73}{24} N \log_2 N - \frac{25}{8} N + 4.$$

В последнее время большое внимание уделяется алгоритмам БПФ с расщеплённым основанием (БПФРО), в которых вычисления проводятся одновременно по основаниям 2 и 4. Арифметическая сложность алгоритмов БПФРО [77, 78] определяется следующим образом:

$$M_R(N) = N \log_2 N - 3N + 4,$$

$$A_R(N) = 3N \log_2 N - 3N + 4.$$

Арифметическая сложность рассмотренных алгоритмов БПФ, предназначенных для обработки комплексных последовательностей, представлена в табл. I.1 для некоторых длин N .

Объём памяти, требуемый для реализации алгоритмов БПФ, включает в себя объём памяти для хранения входных, промежуточных и выходных данных, объём памяти для хранения поворачивающих

множителей и объем памяти для хранения кода программы. Поскольку все описанные выше алгоритмы могут быть выполнены на месте (с замещением), то для хранения данных они требуют $2N$ ячеек оперативного запоминающего устройства (ОЗУ) (для действительной и мнимой частей). Для хранения поворачивающих множителей требуется в случае использования алгоритмов комплексного умножения (I.15) и (I.14) соответственно $N-8$ или $3N/2-12$ ячеек постоянного запоминающего устройства (ПЗУ) для алгоритмов БПФ-2 и БПФ0, $3N/2-16$ или $9N/4-24$ для алгоритмов БПФ-4 и $7N/4-24$ или $21N/8-36$ - для алгоритма БПФ-8. Объем памяти для хранения кода программы зависит от особенностей программы, языка программирования, используемого оборудования и некоторых других факторов и может быть оценен в каждом конкретном случае на основе экспериментальных данных. Объем памяти для некоторых длин N приведен в табл. I.2.

Сравнительный анализ алгоритмов БПФ для обработки комплексных последовательностей показывает, что с ростом основания арифметическая сложность алгоритмов уменьшается, а объем требуемой памяти увеличивается. Наименьшие арифметическую сложность и объем требуемой памяти имеют алгоритмы БПФ0.

В случае, когда входная последовательность $x(n)$ является вещественной, вычисление ДПФ по одному из выше приведенных алгоритмов БПФ является избыточным и неэффективным, поскольку половина входных данных в этом случае равна нулю (все мнимые части), и половина результатов вычислений не содержит никакой новой информации (в силу комплексной сопряженности ДПФ вещественной последовательности).

Существует два основных класса алгоритмов БПФ вещественной последовательности. Первый из них основан на учете комплексно-сопряженного характера ДПФ вещественной последовательности.

Арифметическая сложность алгоритмов БПФ комплексной последовательности

N	БПФ-2		БПФ-4		БПФ-8		БПФ-16	
	$M_R(N)!$	$A_R(N)$	$M_R(N)!$	$A_R(N)$	$M_R(N)!$	$A_R(N)$	$M_R(N)!$	$A_R(N)$
2	0	4					0	4
4	0	16	0	16			0	16
8	4	52			4	52	4	52
16	24	152	20	148			20	148
32	88	408					68	388
64	264	1032	208	976	204	972	196	964
128	712	2504					516	2308
256	1800	5896	1392	5488			1284	5380
512	4360	13576			3204	12420	3076	12292
1024	10248	30728	7856	28336			7172	27652
2048	23560	68616					16388	61444
4096	53256	151560	40624	138928	38404	136708	36868	135172

Таблица 1.2

Объём памяти алгоритмов БФ комплексной последовательности

N	! ОЗУ	! БФ-2 (ПЗУ) !(1.15)	! БФ-4 (ПЗУ) !(1.14)!(1.15)	! БФ-8 (ПЗУ) !(1.14)!(1.15)	! БФ-16 (ПЗУ) !(1.14)!(1.15)!(1.14)
16	32	8	12	8	12
32	64	24	36	24	36
64	128	56	84	88	132
128	256	120	180	120	180
256	512	248	372	552	372
512	1024	504	756	872	1308
1024	2048	1016	1524	2280	3528
2048	4096	2040	3060	4560	7056
4096	8192	4088	6132	9192	13824

Версия алгоритма БПФ по основанию 2 с прореживанием по времени для обработки вещественной последовательности (БПФД-2) была предложена в [79]. Арифметическая сложность данного алгоритма даётся равенствами:

$$M_R(N) = \frac{3}{4} N \log_2 N - \frac{5}{2} N + 4,$$

$$A_R(N) = \frac{7}{4} N \log_2 N - \frac{7}{2} N + 6.$$

Подобные алгоритмы БПФ вещественной последовательности можно получить также для оснований 4 (БПФД-4) и 8 (БПФД-8) [80]. Арифметическая сложность данных алгоритмов определяется соответственно равенствами:

$$M_R(N) = \frac{9}{16} N \log_2 N - \frac{43}{24} N + \frac{8}{3},$$

$$A_R(N) = \frac{25}{16} N \log_2 N - \frac{21}{8} N + 4$$

и

$$M_R(N) = \frac{25}{48} N \log_2 N - \frac{167}{112} N + \frac{10}{7},$$

$$A_R(N) = \frac{73}{48} N \log_2 N - \frac{37}{16} N + 2.$$

Алгоритм БПФ вещественной последовательности с расщеплённым основанием (БПФДРО) был предложен в [81] для случая прореживания по частоте и в [82, 83] для случая прореживания по времени. Арифметическая сложность данного алгоритма даётся равенствами

$$M_R(N) = \frac{N}{2} \log_2 N - \frac{3}{2} N + 2,$$

$$A_R(N) = \frac{3}{2} N \log_2 N - \frac{5}{2} N + 4.$$

Арифметическая сложность алгоритмов БПФ вещественной последовательности первого класса для некоторых длин N представлена в табл. I.3.

Данные алгоритмы требуют N ячеек ОЗУ для хранения данных и $N/2 - 4$ или $3N/4 - 6$ ячеек ПЗУ в алгоритмах БПФД-2 и БПФДРО, $3N/4 - 8$ или $9N/8 - 12$ - в алгоритмах БПФД-4, $7N/8 - 16$ или $21N/16 - 24$ - в алгоритмах БПФД-8 для хранения поворачивающих множителей соответственно в случае использования алгоритмов комплексного умножения (I.15) или (I.14). Объем памяти для некоторых длин N приведен в табл. I.4.

Сравнительный анализ выше описанных алгоритмов БПФ вещественной последовательности показывает, что с ростом основания арифметическая сложность алгоритмов уменьшается, а объем требуемой памяти увеличивается. При этом количество вещественных умножений и объем требуемой памяти для алгоритмов по основаниям 2, 4 и с расщепленным основанием равно вдвое меньше, чем для соответствующих алгоритмов БПФ комплексной последовательности, а количество вещественных сложений более, чем вдвое меньше. Наименьшую арифметическую сложность и объем требуемой памяти имеют алгоритмы БПФДРО.

Несмотря на значительное сокращение объема вычислений и требуемой памяти, данные алгоритмы обладают рядом недостатков: сложность адресации данных в процессе вычислений, необходимость наличия большого количества базовых вычислительных операций и ряд других. От указанных недостатков свободны алгоритмы БПФ вещественной последовательности второго класса, которые на одном из этапов вычислений используют алгоритмы БПФ комплексной последовательности.

Известны два способа реализации данного метода [84]: 1) совме-

Таблица 1.3

Арифметическая сложность алгоритмов БПФ вещественной последовательности
первого класса

N	$! \text{ БПФ-2}$	$! \text{ БПФ-4}$	$! \text{ БПФ-8}$	$! \text{ БПФ-16}$
N	$! M_R(N) ! A_R(N) !$	$! M_R(N) ! A_R(N) !$	$! M_R(N) ! A_R(N) !$	$! M_R(N) ! A_R(N) !$
2	0	2	0	2
4	0	6	0	6
8	2	20	2	20
16	12	62	10	62
32	44	174	34	164
64	132	454	106	420
128	356	1126	258	1028
256	900	2694	642	2436
512	2180	6278	1538	5636
1024	5124	14342	3586	12804
2048	11780	32262	8194	28676
4096	26628	71686	18434	63492

Таблица 1.4

Объём памяти алгоритмов БФ вещественной последовательности первого класса

N	!	ОЗУ	!	БФД-2 (ПЗУ)	!	БФД-4 (ПЗУ)	!	БФД-8 (ПЗУ)	!	БФД-16 (ПЗУ)
	!		!	!(1.15)	!	!(1.14)!(1.15)	!	!(1.14)!(1.15)	!	!(1.14)!(1.15)!(1.16)
16	!	16	!	4	!	6	!	4	!	6
32	!	32	!	12	!	18	!	12	!	18
64	!	64	!	28	!	42	!	28	!	42
128	!	128	!	60	!	90	!	60	!	90
256	!	256	!	124	!	186	!	124	!	186
512	!	512	!	252	!	378	!	252	!	378
1024	!	1024	!	508	!	762	!	508	!	762
2048	!	2048	!	1020	!	1530	!	1020	!	1530
4096	!	4096	!	2044	!	3066	!	2044	!	3066

щённое преобразование двух вещественных последовательностей и
 2) разбиение одной вещественной последовательности с чётным числом членов на две вещественные последовательности, совмещённое преобразование этих последовательностей и восстановление результата преобразования исходной последовательности.

Первый способ состоит в том, что из двух вещественных последовательностей $a(n)$ и $b(n)$, $n = \overline{0, N-1}$, образуют комплексную последовательность

$$z(n) = a(n) + j b(n)$$

и вычисляют её ДПФ с помощью одного из алгоритмов БПФ для обработки комплексных последовательностей. ДПФ исходных последовательностей находят из соотношений

$$\begin{aligned} A(k) &= \frac{z(k) + z^*(N-k)}{2}, \\ B(k) &= -j \frac{z(k) - z^*(N-k)}{2}, \end{aligned} \quad (1.16)$$

где $A(k)$, $B(k)$, $z(k)$ — соответственно ДПФ последовательностей $a(n)$, $b(n)$, $z(n)$; $z^*(N-k)$ — величина, комплексно-сопряжённая $z(N-k)$; $k = \overline{0, N-1}$.

С учётом комплексной сопряженности $A(k)$ и $B(k)$ для вычисления (1.16) требуется только $2N-4$ вещественных сложений.

Второй способ состоит в том, что вещественная последовательность $x(n)$, $n = \overline{0, 2N-1}$, разбивается на две последовательности $a(n)$ и $b(n)$, $n = \overline{0, N-1}$, составленные соответственно из чётных и нечётных членов исходной последовательности, ДПФ которых вычисляется согласно (1.16). ДПФ исходной последовательности вычисляется следующим образом:

$$F(k) = A(k) + B(k) \cdot W_{2N}^k, \quad k = \overline{0, 2N-1}. \quad (1.17)$$

С учётом комплексной сопряженности $F(k)$ для вычисления (1.17) требуется $3N-8$ вещественных умножений и $5N-10$

вещественных сложений.

Объём вычислений можно сократить, если вычисления проводить следующим образом [85]:

$$\begin{aligned} F(k) &= A(k) + B(k) \cdot W_{2N}^k, \\ F(k+N) &= A(k) - B(k) \cdot W_{2N}^k, \quad k = \overline{0, N/2}. \end{aligned} \quad (I.18)$$

Вычисление (I.18) требует $3N/2 - 4$ вещественных умножений и $7N/2 - 6$ вещественных сложений. Кроме того, вдвое уменьшается число поворачиваемых множителей.

Таким образом, для вычислений (I.16) и (I.18) требуется $3N/2 - 4$ вещественных умножений и $11N/2 - 10$ вещественных сложений. Просуммировав данный объём вычислений с объёмом вычислений, требуемым для вычисления БПФ комплексной последовательности, получим арифметическую сложность алгоритмов БПФ для обработки вещественной последовательности длины $2N$, на основании чего можно получить арифметическую сложность этих алгоритмов для обработки вещественной последовательности длины N , которая определяется следующим образом:

$$M_R(N) = \frac{3}{4} N \log_2 N - \frac{5}{2} N + 4,$$

$$A_R(N) = \frac{7}{4} N \log_2 N - \frac{3}{2} N - 2$$

при использовании алгоритмов БПФ-2;

$$M_R(N) = \frac{9}{16} N \log_2 N - \frac{77}{48} N + \frac{4}{3},$$

$$A_R(N) = \frac{25}{16} N \log_2 N - \frac{29}{48} N - \frac{14}{3}$$

при использовании алгоритмов БПФ-4;

$$M_R(N) = \frac{25}{48} N \log_2 N - \frac{4}{3} N,$$

$$A_R(N) = \frac{73}{48} N \log_2 N - \frac{1}{3} N - 6$$

при использовании алгоритмов БПФ-8;

$$M_R(N) = \frac{1}{2} N \log_2 N - \frac{5}{4} N,$$

$$A_R(N) = \frac{3}{2} N \log_2 N - \frac{1}{4} N - 6$$

при использовании алгоритмов БПФ_{FO}.

Данные алгоритмы требуют N ячеек ОЗУ для хранения данных и $N-12$, $3N/2-18$ ячеек ПЗУ при использовании алгоритмов БПФ-2 и БПФ_{FO}, $5N/4-20$ или $15N/8-30$ - при использовании алгоритмов БПФ-4 и $11N/8-28$ или $33N/16-42$ - при использовании алгоритмов БПФ-8 для хранения поворачивающих множителей соответственно в случае использования алгоритмов комплексного умножения (I.15) или (I.14).

Арифметическая сложность, а также объём требуемой памяти данных алгоритмов для некоторых длин N приведены соответственно в табл. I.5 и I.6.

Сравнительный анализ данных алгоритмов показывает, что наилучшими среди них с точки зрения объёма вычислений и требуемой памяти являются алгоритмы, использующие алгоритмы БПФ_{FO}, так как последние являются наилучшими среди алгоритмов для обработки

Таблица 1.5

Арифметическая сложность алгоритмов БПФ вещественной последовательности второго класса

N	При использовании БПФ-2		При использовании БПФ-4		При использовании БПФ-8		При использовании БПФ-16	
	$M_R(N)$	$A_R(N)$	$M_R(N)$	$A_R(N)$	$M_R(N)$	$A_R(N)$	$M_R(N)$	$A_R(N)$
8	2	28	2	28			2	28
16	12	86			12	86	12	86
32	44	230	40	226			40	226
64	132	574					112	554
128	356	1374	300	1318	296	1314	288	1306
256	900	3198					704	3002
512	2180	7294	1772	6886			1664	6778
1024	5124	16382			3968	15226	3840	15098
2048	11780	36350	9388	33958			8704	33274
4096	26628	79870					19456	72698
8192	59396	174078	46764	161446	44544	159226	43008	157690

Таблица 1.6

Объём памяти алгоритмов БПФ вещественной последовательности второго класса

N	! При использовании! При использовании! При использовании! При использовании!		БПФ-2 (ПЗУ) ! БПФ-4 (ПЗУ) ! БПФ-8 (ПЗУ) ! БПФ-8 (ПЗУ)		! БПФ-8 (ПЗУ) ! БПФ-8 (ПЗУ)		! БПФ-8 (ПЗУ) ! БПФ-8 (ПЗУ)		
	(I.15) ! (I.14) ! (I.15) ! (I.14) ! (I.15) ! (I.14) ! (I.15) ! (I.14)	(I.15) ! (I.14) ! (I.15) ! (I.14) ! (I.15) ! (I.14) ! (I.15) ! (I.14)	(I.15) ! (I.14) ! (I.15) ! (I.14) ! (I.15) ! (I.14) ! (I.15) ! (I.14)	(I.15) ! (I.14) ! (I.15) ! (I.14) ! (I.15) ! (I.14) ! (I.15) ! (I.14)	(I.15) ! (I.14) ! (I.15) ! (I.14) ! (I.15) ! (I.14) ! (I.15) ! (I.14)	(I.15) ! (I.14) ! (I.15) ! (I.14) ! (I.15) ! (I.14) ! (I.15) ! (I.14)	(I.15) ! (I.14) ! (I.15) ! (I.14) ! (I.15) ! (I.14) ! (I.15) ! (I.14)	(I.15) ! (I.14) ! (I.15) ! (I.14) ! (I.15) ! (I.14) ! (I.15) ! (I.14)	
32	32	20	30	20	30	20	30	20	30
64	64	52	78			52	78		78
128	128	116	174	140	210	148	222	116	174
256	256	244	366	620	930			244	366
512	512	500	750					500	750
1024	1024	1012	1518			1380	2070	1012	1518
2048	2048	2036	3054	2540	3810			2036	3054
4096	4096	4084	6126					4084	6126
8192	8192	8180	12270	10220	15330	11236	16854	8180	12270

комплексной последовательности. В то же время данные алгоритмы уступают алгоритмам БПФ вещественной последовательности первого класса.

Наряду с описанными выше алгоритмами существуют и другие быстрые алгоритмы вычисления ДПФ вещественной последовательности [86, 87]. Однако данные алгоритмы обладают такими существенными недостатками, как нерегулярная структура, произвольный порядок входных и выходных отсчётов данных и ряд других. При этом данные алгоритмы не имеют преимуществ по арифметической сложности и затратам памяти перед лучшими алгоритмами БПФ вещественной последовательности, описанными выше.

От недостатков, присущих алгоритмам БПФ вещественной последовательности, свободны алгоритмы быстрого преобразования Хартли (БПХ), предназначенные для быстрого вычисления ДПХ. Алгоритмы БПХ имеют регулярную и простую структуру, что обеспечивает лёгкость их применения. В [88] Брейсуэлл впервые предложил алгоритм БПХ по основанию 2 с прореживанием по времени. Однако, предложенный алгоритм выполнялся без замещения. В [89] показана возможность выполнения данного алгоритма с замещением. Наконец, в [43] был предложен эффективный алгоритм БПХ по основанию 2 с прореживанием по времени, выполняемый с замещением и позволяющий использовать эффективные вычислительные процедуры. Версия алгоритма БПХ по основанию 2 с прореживанием по частоте была предложена в [44]. Вывод данных алгоритмов можно также найти в [90].

Анализ алгоритмов БПХ по основанию 2 [43, 44] показывает, что каждый этап вычислений в общем случае состоит из двух подэтапов. При этом на одном из них выполняются базовые операции, содержащие лишь операции суммирования - вычитания, а на другом базовые операции, приведенные на рис.1.2.

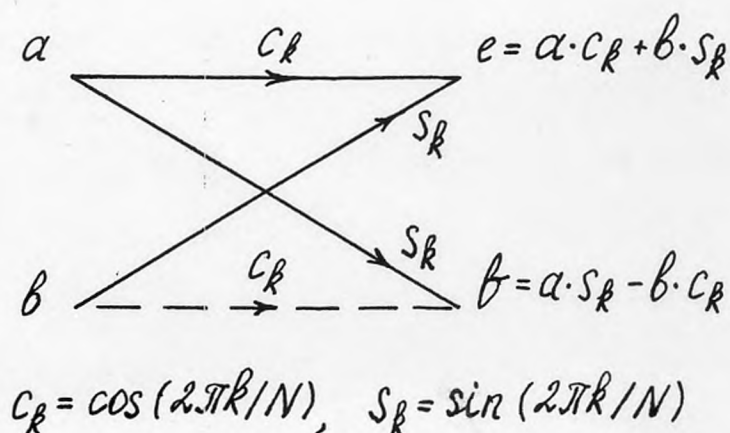


Рис.1.2. Граф-схема базовой операции алгоритмов БПХ по основанию 2 с поворачивающими множителями

Анализ базовой операции показывает, что выходы e и f есть не что иное, как действительная и взятая с обратным знаком мнимая части комплексного произведения чисел $a+jb$ и $\cos(2\pi k/N) - j \sin(2\pi k/N)$. Поэтому для их вычисления может быть использован любой алгоритм умножения комплексных чисел. В частности, в [43] использован алгоритм, определяемый выражениями (1.15).

Арифметическая сложность алгоритмов БПХ по основанию 2 (БПХ-2) определяется равенствами [74]

$$M_R(N) = \frac{3}{4} N \log_2 N - \frac{5}{2} N + 4,$$

$$A_R(N) = \frac{7}{4} N \log_2 N - \frac{5}{2} N + 4.$$

Данные алгоритмы требуют N ячеек ОЗУ для хранения данных и $N/2 - 4$ или $3N/4 - 6$ ячеек ПЗУ для хранения поворачивающих множителей соответственно в случае использования алгоритмов комплексного умножения (1.15) или (1.14).

Сравнительный анализ алгоритмов БПХ по основанию 2 и соответствующих им алгоритмов БПФ показывает следующее:

- алгоритмы БПХ превосходят алгоритмы БПФ комплексной последовательности по количеству арифметических операций и объему требуемой памяти в 2 раза;

- алгоритмы БПХ превосходят алгоритмы БПФ вещественной последовательности второго класса по количеству операций вещественного сложения и объему требуемой памяти, что для $N = 1024$ составляет выигрыш соответственно в 1,07 раза и 1,33 раза, а по количеству операций вещественного умножения имеют одинаковые характеристики;

- алгоритмы БПХ и алгоритмы БПФ вещественной последовательности первого класса имеют одинаковые характеристики по количеству операций вещественного умножения и объему требуемой памяти, а по количеству операций вещественного сложения последние превосходят первые на $(N - 2)$ операций, что для $N = 1024$ составляет выигрыш в 1,07 раза.

Таким образом, алгоритмы БПХ превосходят алгоритмы БПФ, незначительно уступая алгоритмам БПФ вещественной последовательности первого класса. Однако, поскольку алгоритмы БПХ в отличие от алгоритмов БПФ вещественной последовательности легки в применении и свободны от недостатков, присущих последним, их применение является более предпочтительным.

В [91] в общем виде представлен алгоритм БПХ по основанию 4 с прореживанием по времени и показана возможность выполнения его с замещением. В [92] получен алгоритм БПХ с расщепленным основанием с прореживанием по частоте. Недостатком данного алгоритма является повышенное количество вычислительных операций и повышенный объем памяти для хранения поворачивающих множителей. Наконец, в [93, 94] показана возможность получения алгоритмов БПХ с произвольным основанием, алгоритмов БПХ простых множителей и алгоритма Винограда преобразования Хартли. Однако, данные

алгоритмы имеют ряд недостатков, к числу которых относятся невозможность использования эффективных вычислительных процедур, повышенное количество вычислительных операций, сложная индексация входных или выходных отсчетов и выполнение вычислений без замещения [93], наличие дополнительных этапов вычислений [94], что приводит к увеличению времени и снижению точности вычислений. Поэтому с учетом выше изложенного актуальной является задача разработки эффективных алгоритмов БПХ, в частности, алгоритмов БПХ Кули-Тьюки с постоянным основанием, выполняемых с замещением и требующих минимального количества вычислительных операций и объема памяти.

На практике часто приходится вычислять ДПФ и ДПХ на скачущих (скользящих) интервалах, т.е. когда к исходной последовательности добавляется некоторое число новых отсчетов и такое же число начальных отсчетов исходной последовательности отбрасывается. Использование в этом случае алгоритмов БПФ и БПХ для вычисления ДПФ и ДПХ является неэффективным в виду того, что они не используют результаты вычислений на предыдущих интервалах. Поэтому возникает задача поиска эффективных алгоритмов вычисления ДПФ и ДПХ, учитывающих результаты их вычисления на предыдущих интервалах. Такие алгоритмы получили название рекуррентных.

Известны рекуррентные алгоритмы вычисления ДПФ на скачущих (скользящих) интервалах [52, 53], а также рекуррентный алгоритм вычисления энергетического спектра на основе ДПХ на скользящих интервалах [54]. Данные алгоритмы обладают рядом преимуществ по сравнению с алгоритмами БПФ и БПХ, к числу которых относятся существенно меньший объем вычислительных затрат, возможность вычисления произвольного числа гармоник, простота и легкость их применения. Поэтому целесообразным является разработка рекуррентных алгоритмов вычисления ДПХ и энергетического спектра на их основе на скачущих (скользящих) интервалах.

1.4. Точность алгоритмов быстрого преобразования Фурье и Хартли

Важным критерием выбора того или иного алгоритма является точность вычислений, обеспечиваемая данным алгоритмом. При этом в качестве количественной оценки точности принимается отношение среднеквадратического значения (СКЗ) ошибки на выходе алгоритма к СКЗ выходного сигнала, которое физически интерпретируется как отношение мощности шума к мощности сигнала.

В основу аналитического анализа точности алгоритмов БПФ и БПХ положен статистический метод анализа, предложенный Оппенгеймом и Вайнштейном [45]. При статистическом методе анализа каждому источнику элементарной ошибки ставится в соответствие независимый генератор белого шума с дисперсией и математическим ожиданием ошибки, являющейся случайной величиной и принимаются следующие допущения:

- 1) все источники элементарных ошибок не коррелируют с входным сигналом;
- 2) все источники элементарных ошибок не коррелируют друг с другом.

При реализации алгоритмов БПФ и БПХ с фиксированной запятой, использование которой удовлетворяет ряду противоречивых требований по быстродействию и точности вычислений, различают три основных вида вычислительных ошибок [15]:

- 1) ошибки умножения, обусловленные округлением либо усечением результатов произведений;
- 2) ошибки сдвига, обусловленные процедурой масштабирования, вводимой с целью устранения возможного переполнения;
- 3) ошибки, обусловленные квантованием поворачивающих множителей.

Для устранения переполнений используются следующие виды

масштабирования [48]:

1) входное масштабирование, при котором входная последовательность масштабируется таким образом, чтобы в процессе вычислений не возникло переполнений;

2) автоматическое масштабирование, при котором на каждом этапе вычислений осуществляется масштабирование обрабатываемых данных;

3) условное масштабирование, при котором на каждом этапе вычислений масштабирование осуществляется в том случае, если возможно возникновение переполнения.

При анализе вычислительных ошибок в алгоритмах с фиксированной запятой предполагается, что все промежуточные результаты вычислений представляются дробными числами, содержащими $b+1$ разрядов, включая знаковый разряд, а входные данные нормированы таким образом, что $|x(n)| < 1$ для всех n .

Так как результатом перемножения двух $b+1$ - разрядных чисел является $2b+1$ - разрядное число, которое затем округляется либо усекается до $b+1$ - разрядного числа, то возникающая ошибка умножения ε_y представляет собой случайную величину с математическим ожиданием m_{ε_y} , дисперсией $\sigma_{\varepsilon_y}^2$ и СКЗ $M(\varepsilon_y^2)$, значения которых определены в [49].

В результате перемножения двух комплексных чисел возникает комплексная ошибка умножения ε_y^k , статистические характеристики которой зависят от используемого алгоритма умножения. В случае использования алгоритма, требующего четыре вещественные умножения, комплексная ошибка умножения ε_y^k будет иметь математическое ожидание

$$m_{\varepsilon_y^k} = 2 \cdot (1+j) m_{\varepsilon_y}$$

и СКЗ

$$M(|\varepsilon_y^k|^2) = 4 M(\varepsilon_y^2) + 4 m_{\varepsilon_y}^2.$$

Анализ ошибок сдвига, обусловленных процедурой масштабирования, возможен лишь в двух случаях - для входного и автоматического масштабирования.

При входном масштабировании данные на входе алгоритма сдвигаются на $m = \log_2 N$ разрядов вправо, и тем самым устраняются возможные переполнения в ходе выполнения алгоритма. Ошибка сдвига действительного числа на входе алгоритма обладает при этом такими же статистическими характеристиками, как и при усечении результата произведения действительных чисел.

При автоматическом масштабировании на каждом этапе алгоритма происходит сдвиг чисел на $\log_2 \varepsilon$, где ε - основание алгоритма, разрядов вправо. При этом ошибка сдвига ε_c действительного числа характеризуется математическим ожиданием m_{ε_c} , дисперсией $\sigma_{\varepsilon_c}^2$ и СКЗ $M(\varepsilon_c^2)$.

При сдвиге комплексного числа возникает комплексная ошибка сдвига ε_c^k , которая характеризуется математическим ожиданием

$$m_{\varepsilon_c^k} = (1+j)m_{\varepsilon_c}$$

и СКЗ

$$M(|\varepsilon_c^k|^2) = 2M(\varepsilon_c^2).$$

Статистические характеристики ошибок умножения и сдвига приведены соответственно в табл. 1.7 и 1.8

Анализу вычислительных ошибок алгоритмов БПФ посвящено большое количество работ [14, 15, 45 - 50]. Однако, в одних из них [14, 15, 45, 46, 50] проводился упрощенный анализ алгоритмов БПФ, в других [47-49] - проводимый анализ слишком сложен и не нагляден. При этом анализ проводился лишь для некоторых частных случаев при реализации алгоритмов БПФ по основаниям 2 и 4.

Статистические характеристики ошибок умножения

Вид аппроксимации результата произведения	Статистические характеристики			
	m_{cy}	σ_{cy}^2	$M(\epsilon_y^2)$	$M(\epsilon_y ^2)$
Округление прямого, обратного и дополнительного кода	0	$2^{-26}/12$	$2^{-26}/12$	$2^{-26}/3$
усечение прямого и обратного кода	0	$2^{-26}/3$	$2^{-26}/3$	$4 \cdot 2^{-26}/3$
усечение дополнительного кода	$-2/2$	$2^{-26}/12$	$2^{-26}/3$	$7 \cdot 2^{-26}/3$

Статистические характеристики ошибок сдвига

Сдвиг на	Вид сдвига числа	Статистические характеристики			
		$m_{\epsilon c}$	$\sigma_{\epsilon c}^2$	$M(\epsilon c)$	$M(\epsilon c^2)$
Сдвиг на 1 разряд	Прямой и обратный код	0	$2^{-26}/8$	$2^{-26}/8$	$2^{-26}/4$
	Дополнительный код	$-2^{-6}/4$	$2^{-26}/16$	$2^{-26}/8$	$2^{-26}/4$
Сдвиг на 2 разряда	Прямой и обратный код	0	$7.2^{-26}/32$	$7.2^{-26}/32$	$7.2^{-26}/16$
	Дополнительный код	$-3.2^{-6}/8$	$5.2^{-26}/64$	$7.2^{-26}/32$	$7.2^{-26}/16$
Сдвиг на 3 разряда	Прямой и обратный код	0	$35.2^{-26}/128$	$35.2^{-26}/128$	$35.2^{-26}/64$
	Дополнительный код	$-7.2^{-6}/16$	$21.2^{-26}/256$	$35.2^{-26}/128$	$35.2^{-26}/64$

Для алгоритмов БПХ анализ вычислительных ошибок проводился в [36, 51]. При этом рассматривались алгоритмы БПХ по основанию 2 с представлением чисел в дополнительном коде и округлением результатов произведений.

Поэтому представляет интерес задача простого и наглядного анализа вычислительных ошибок алгоритмов БПФ и БПХ по разным основаниям, с прореживанием по времени и частоте с фиксированной запятой для случаев входного и автоматического масштабирования, с представлением чисел в прямом, обратном и дополнительном коде, с аппроксимацией результатов умножения путем округления или усе- чения, который позволил бы в каждом конкретном случае с учетом имеющихся возможностей и требований разрабатывать устройства (программы) для реализации алгоритмов БПФ и БПХ.

1.5. Микропроцессорная реализация ИИС ЦОС

Длительное время задачи ЦОС решались на больших и малых ЭВМ, что крайне сужало область применения ЦОС в силу ограничений по массогабаритным характеристикам и потребляемой мощности используемых ЭВМ. Лишь появление микропроцессоров (МП) позволило преодолеть указанные недостатки и максимально приблизить устройства ЦОС, создаваемые на их основе, к источникам сигналов и объектам управления.

В настоящее время существует большое разнообразие микропроцессорных комплектов (МПК) [96 - 102]. В зависимости от способов реализации микропроцессорные ИИС ЦОС можно подразделить на следующие основные группы [55, 103]:

- основанные на МП и микроЭВМ с фиксированной разрядностью;
- основанные на МП с фиксированной разрядностью со вспомога- тельными устройствами;
- основанные на разрядно-секционированных МП;

- мультимикропроцессорные устройства;
- однокристалльные цифровые процессоры сигналов (ЦПС);
- многокристалльные ЦПС;
- ЦПС на основе систолических структур.

При построении устройств ЦОС на универсальных МП последние выбираются по ряду своих характеристик, основными из которых являются [104]: число микросхем в МПК, число внутренних магистралей, разрядность, система команд, число регистров общего назначения, быстродействие, число буферных регистров ввода-вывода, число уровней прерывания и др.

К числу МПК с фиксированной разрядностью относятся КР580 (*Intel 8080*), КР1801, КР1806, КР1809, КР1810 (*Intel 8086*), КР1821 (*Intel 8085*). Данные МПК имеют большое количество БИС в своем составе, что позволяет строить на их основе функционально законченные узлы в составе ИИС ЦОС. Их разрядность составляет 8 - 16 бит, они имеют 6 - 8 регистров общего назначения, широкую систему команд. Основным недостатком, ограничивающим их применение при создании ИИС ЦОС, является низкое быстрое и программная реализация операции умножения.

С целью повышения быстродействия ИИС ЦОС, создаваемых на основе МП с фиксированной разрядностью они дополняются вспомогательными устройствами, к числу которых относятся умножители КР1802ВР2 - ВР5, умножитель-накопитель К1518ВЖ1, арифметические сопроцессоры КР1810ВМ57 (*Intel 8087*) и КР1801 ВМ4. Первые позволяют существенно ускорить выполнение операции умножения (время умножения составляет 160 - 200 нс), последние - за счет поддержки вычислений с плавающей запятой расширяют динамический диапазон и точность производимых вычислений, позволяют вычислять сложные функции.

Наибольшей производительностью и гибкостью среди универ-

сальных МП обладают ИИС ЦОС, созданные на основе разрядно-секционированных МП, к числу которых относятся МКК К589 (*Intel 3000*), К1800 (*MC 10800*), К1802, К1804 (*Am 2903*). Важнейшими особенностями данных МКК для создания ИИС ЦОС являются: многопортовая структура БИС, позволяющая эффективно выполнять основные операции ЦОС; обеспечение конвейерной обработки информации; возможность построения устройств ЦОС с произвольной разрядностью; система команд, задаваемая пользователем. Основным недостатком ИИС ЦОС на базе разрядно-секционированных МП является высокая стоимость данных устройств, обусловленная сложностью и длительностью их разработки.

В случае, когда быстродействия однопроцессорной ИИС ЦОС недостаточно, создают мультимикропроцессорные ИИС [105].

Новый этап в развитии средств ЦОС наступил с появлением ЦПС. Первый однокристалльный ЦПС *Intel* 2920 (советский аналог К1813ВЕ1) представляет из себя специализированную микроЭВМ со встроенными АЦП и ЦАП, предназначенную для реализации цифровых фильтров невысокого порядка на частоте до десятков кГц. Вслед за ЦПС, предназначенными для цифровой фильтрации, появились ЦПС для широкого круга задач ЦОС, включая БПФ, примером которых может служить ЦПС *TMS* 320010. Основные характеристики однокристалльных ЦПС можно найти в [55].

Дальнейшего повышения быстродействия ИИС ЦОС можно достичь на базе многокристалльных (разрядно-секционированных) ЦПС, представителем которых является семейство СВИС *Am* 29500 фирмы АМД. Реализация 1024-точечного БПФ на данном комплекте требует всего 2 мс [103].

Архитектура ЦПС максимально учитывает специфику алгоритмов ЦОС, в результате чего обеспечивается высокое быстродействие

ИИС ЦОС, создаваемых на их основе.

Высокая степень параллелизма многих алгоритмов ЦОС стимулировали разработку ЦПС на основе систолических структур [106, 107]. Однако на данный момент времени работы в этой области находятся на уровне теоретических исследований.

Сравнительный анализ ИИС ЦОС [55, 103] показывает, что по критерию производительность-стоимость наилучшими характеристиками обладают ИИС ЦОС, реализованные на основе ЦПС и МП с фиксированной разрядностью. Поэтому в случае, когда не предъявляются жесткие требования по быстродействию, ИИС ЦОС целесообразно разрабатывать на основе МП с фиксированной разрядностью, поскольку единственный ЦПС К1813ВЕ1, выпускаемый в нашей стране, пригоден лишь для цифровой фильтрации.

Как отмечалось ранее, при разработке ИИС ЦОС большое внимание должно отводиться каналам связи, обеспечивающим обмен информации и согласование работы составных частей ИИС [57]. Среди известных способов обмена информацией [104]: программно-управляемый, по прерыванию и прямой доступ к памяти (ПДП), наиболее пригодным для ИИС ЦОС является способ ПДП, не требующий затрат процессорного времени на обмен информацией. Для обеспечения режима ПДП в составе различных МПК разработаны специализированные БИС - контроллеры ПДП, например БИС КР580ВТ57. Однако, данные контроллеры ПДП работают с блокировкой МП, что существенно снижает производительность МП системы. Так, например, если использовать их в устройствах для отображения информации на экране электронно-лучевой трубки [108], то они могут приводить к снижению производительности МП в $1,125 \div 1,5$ раза. Поэтому важной задачей является разработка устройств ПДП, а также других устройств, включающих их в свой состав, которые не снижают производительность МП, а, следовательно, и всего ИИС ЦОС в целом.

В случае, когда производительности одного МП оказывается недостаточно, разрабатывают мультимикропроцессорные системы ЦОС. При этом важной задачей является разработка средств арбитража и управления доступом к системным ресурсам, обеспечивающих высокую производительность ИИС.

Важное место занимает программное обеспечение микропроцессорных ИИС ЦОС. Анализ имеющихся публикаций по реализации алгоритмов БПФ [56, 109] на МП с фиксированной разрядностью показывает, что время выполнения 1024-точечного алгоритма БПФ составляет $18 \div 24$ сек, что делает непригодным их применение не только для работы в реальном масштабе времени, но и для оперативного анализа сигналов. Поэтому актуальной является задача разработки эффективных программных реализаций, максимально учитывающих особенности используемых алгоритмов и МП.

В Ы В О Д Ы

На основе анализа известных алгоритмов ИИС ЦОС в данной работе ставятся следующие задачи:

1. Разработать эффективные алгоритмы БПХ по основаниям 4,8 и с расщепленным основанием с прореживанием по времени и частоте и провести сравнительный анализ по количеству арифметических операций и объему требуемой памяти.
2. Разработать рекуррентные алгоритмы вычисления ДПХ и энергетического спектра на скачущих (скользящих) интервалах, провести сравнительный анализ по количеству арифметических операций и объему требуемой памяти.
3. Исследовать точность алгоритмов БПФ и БПХ по основаниям 2, 4, 8 и с расщепленным основанием с прореживанием по времени

и частоте с фиксированной запятой для случаев входного и автоматического масштабирования, представления чисел в прямом, обратном и дополнительном коде, аппроксимации результатов умножения путем округления и усечения.

4. Разработать эффективные программные реализации алгоритмов ЦОС, учитывающие особенности используемых алгоритмов и оборудования.

5. Разработать структурные и функциональные схемы устройств, выполняющих обмен информации между составными частями микропроцессорных ИИС ЦОС, которые обеспечивают максимальную производительность последних.

6. Разработать автономные, функционально законченные микропроцессорные анализаторы спектра, позволяющие проводить оперативный спектральный анализ сигналов.

ГЛАВА 2. РАЗРАБОТКА ЭФФЕКТИВНЫХ АЛГОРИТМОВ ВЫЧИСЛЕНИЯ ДИСКРЕТНОГО ПРЕОБРАЗОВАНИЯ ХАРТЛИ И ЭНЕРГЕТИЧЕСКОГО СПЕКТРА

2.1. Алгоритмы быстрого преобразования Хартли по основанию 4

В [110] автором разработаны алгоритмы БПХ по основаниям 4, 8 и с расщепленным основанием с прореживанием по времени и частоте. В данном разделе показан вывод алгоритмов БПХ по основанию 4, в следующих - по основанию 8 и с расщепленным основанием.

Вывод алгоритма БПХ по основанию 4 с прореживанием по времени заключается в следующем.

Разобьем входную последовательность $x(n)$, $n = \overline{0, N-1}$, на последовательности $x(4n)$, $x(4n+1)$, $x(4n+2)$ и $x(4n+3)$, $n = \overline{0, N/4-1}$. Тогда ДПХ последовательности $x(n)$ можно представить в виде

$$\begin{aligned}
 H(k) &= \sum_{n=0}^{N/4-1} x(4n) \cos \frac{2\pi \cdot 4nk}{N} + \sum_{n=0}^{N/4-1} x(4n+1) \cos \frac{2\pi \cdot (4n+1)k}{N} + \\
 &+ \sum_{n=0}^{N/4-1} x(4n+2) \cos \frac{2\pi \cdot (4n+2)k}{N} + \sum_{n=0}^{N/4-1} x(4n+3) \cos \frac{2\pi \cdot (4n+3)k}{N} = \\
 &= H_0(k) + \cos \frac{2\pi k}{N} H_1(k) + \sin \frac{2\pi k}{N} H_1(N/4-k) + \cos \frac{2\pi \cdot 2k}{N} H_2(k) + \\
 &+ \sin \frac{2\pi \cdot 2k}{N} H_2(N/4-k) + \cos \frac{2\pi \cdot 3k}{N} H_3(k) + \sin \frac{2\pi \cdot 3k}{N} H_3(N/4-k),
 \end{aligned}$$

где

$$H_0(k) = \sum_{n=0}^{N/4-1} x(4n) \cos \frac{2\pi n k}{N/4}, \quad H_1(k) = \sum_{n=0}^{N/4-1} x(4n+1) \cos \frac{2\pi n k}{N/4},$$

$$H_2(k) = \sum_{n=0}^{N/4-1} x(4n+2) \cos \frac{2\pi n k}{N/4}, \quad H_3(k) = \sum_{n=0}^{N/4-1} x(4n+3) \cos \frac{2\pi n k}{N/4}$$

представляют собой $N/4$ - точечные ДПХ последовательностей $x(4n)$, $x(4n+1)$, $x(4n+2)$ и $x(4n+3)$ соответственно. Поскольку полученное выражение определяет ДПХ последовательности $x(n)$ для $k = \overline{0, N/4-1}$, полное ДПХ последовательности $x(n)$ для $k = \overline{0, N-1}$ с учётом свойства периодичности ДПХ будет определяться следующим образом:

$$H(k) = H_0(k) + \cos \frac{2\pi k}{N} H_1(k) + \sin \frac{2\pi k}{N} H_1(N/4-k) + \cos \frac{2\pi \cdot 2k}{N} H_2(k) + \\ + \sin \frac{2\pi \cdot 2k}{N} H_2(N/4-k) + \cos \frac{2\pi \cdot 3k}{N} H_3(k) + \sin \frac{2\pi \cdot 3k}{N} H_3(N/4-k),$$

$$H(k + \frac{N}{4}) = H_0(k) - \sin \frac{2\pi k}{N} H_1(k) + \cos \frac{2\pi k}{N} H_1(N/4-k) - \cos \frac{2\pi \cdot 2k}{N} H_2(k) - \\ - \sin \frac{2\pi \cdot 2k}{N} H_2(N/4-k) + \sin \frac{2\pi \cdot 3k}{N} H_3(k) - \cos \frac{2\pi \cdot 3k}{N} H_3(N/4-k),$$

$$H(k + \frac{N}{2}) = H_0(k) - \cos \frac{2\pi k}{N} H_1(k) - \sin \frac{2\pi k}{N} H_1(N/4-k) + \cos \frac{2\pi \cdot 2k}{N} H_2(k) + \\ + \sin \frac{2\pi \cdot 2k}{N} H_2(N/4-k) - \cos \frac{2\pi \cdot 3k}{N} H_3(k) - \sin \frac{2\pi \cdot 3k}{N} H_3(N/4-k),$$

$$H(k + \frac{3N}{4}) = H_0(k) + \sin \frac{2\pi k}{N} H_1(k) - \cos \frac{2\pi k}{N} H_1(N/4-k) - \cos \frac{2\pi \cdot 2k}{N} H_2(k) - \\ - \sin \frac{2\pi \cdot 2k}{N} H_2(N/4-k) - \sin \frac{2\pi \cdot 3k}{N} H_3(k) + \cos \frac{2\pi \cdot 3k}{N} H_3(N/4-k). \quad (2.1)$$

Выражения (2.1) определяют вид алгоритма БИХ по основанию 4 с прореживанием по времени. В матричной форме оно примет следующий вид:

$$\begin{bmatrix} H(k) \\ H(k + \frac{N}{4}) \\ H(k + \frac{N}{2}) \\ H(k + \frac{3N}{4}) \end{bmatrix} = \begin{bmatrix} 1 & C_R & S_R & C_{2k} & S_{2k} & C_{3k} & S_{3k} \\ 1 & -S_R & C_R & -C_{2k} & -S_{2k} & S_{3k} & -C_{3k} \\ 1 & -C_R & -S_R & C_{2k} & S_{2k} & -C_{3k} & -S_{3k} \\ 1 & S_R & -C_R & -C_{2k} & -S_{2k} & -S_{3k} & C_{3k} \end{bmatrix} \cdot \begin{bmatrix} H_0(k) \\ H_1(k) \\ H_1(\frac{N}{4} - k) \\ H_2(k) \\ H_2(\frac{N}{4} - k) \\ H_3(k) \\ H_3(\frac{N}{4} - k) \end{bmatrix},$$

где $k = \overline{0, N/4-1}$, $C_{[\cdot]} = \cos \frac{2\pi[\cdot]}{N}$, $S_{[\cdot]} = \sin \frac{2\pi[\cdot]}{N}$.

Как видно, данный алгоритм является алгоритмом без замещения, поскольку для определения четырёх выходных точек требуется семь входных точек. Для построения алгоритма, выполняемого с замещением, необходимо вычислять также значения $H(\frac{N}{4} - k)$, $H(\frac{N}{2} - k)$, $H(\frac{3N}{4} - k)$ и $H(N - k)$. Тогда получим алгоритм, имеющий вид

$$\begin{bmatrix} H(k) \\ H(\frac{N}{4} - k) \\ H(k + \frac{N}{4}) \\ H(\frac{N}{2} - k) \\ H(k + \frac{N}{2}) \\ H(\frac{3N}{4} - k) \\ H(k + \frac{3N}{4}) \\ H(N - k) \end{bmatrix} = \begin{bmatrix} 1 & 0 & C_R & S_R & C_{2k} & S_{2k} & C_{3k} & S_{3k} \\ 0 & 1 & C_R & S_R & S_{2k} & -C_{2k} & -C_{3k} & -S_{3k} \\ \hline 1 & 0 & -S_R & C_R & -C_{2k} & -S_{2k} & S_{3k} & -C_{3k} \\ 0 & 1 & S_R & -C_R & -S_{2k} & C_{2k} & S_{3k} & -C_{3k} \\ \hline 1 & 0 & -C_R & -S_R & C_{2k} & S_{2k} & -C_{3k} & -S_{3k} \\ 0 & 1 & -C_R & -S_R & S_{2k} & -C_{2k} & C_{3k} & S_{3k} \\ \hline 1 & 0 & S_R & -C_R & -C_{2k} & -S_{2k} & -S_{3k} & C_{3k} \\ 0 & 1 & -S_R & C_R & -S_{2k} & C_{2k} & -S_{3k} & C_{3k} \end{bmatrix} \cdot \begin{bmatrix} H_0(k) \\ H_0(\frac{N}{4} - k) \\ \hline H_1(k) \\ H_1(\frac{N}{4} - k) \\ \hline H_2(k) \\ H_2(\frac{N}{4} - k) \\ \hline H_3(k) \\ H_3(\frac{N}{4} - k) \end{bmatrix}, \quad (2.2)$$

где $k = 0, N/8$.

Для $k=0$ и $k=N/8$ вычисления по (2.2) могут быть объединены и примут следующий вид:

$$\begin{bmatrix} H(0) \\ H(\frac{N}{8}) \\ H(\frac{N}{4}) \\ H(\frac{3N}{8}) \\ H(\frac{N}{2}) \\ H(\frac{5N}{8}) \\ H(\frac{3N}{4}) \\ H(\frac{7N}{8}) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & \sqrt{2} & 0 & 1 & 0 & 0 \\ \hline 1 & 0 & 1 & 0 & -1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & \sqrt{2} \\ \hline 1 & 0 & -1 & 0 & 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -\sqrt{2} & 0 & 1 & 0 & 0 \\ \hline 1 & 0 & -1 & 0 & -1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & -\sqrt{2} \end{bmatrix} \cdot \begin{bmatrix} H_0(k) \\ H_0(\frac{N}{8}) \\ H_1(0) \\ H_1(\frac{N}{8}) \\ H_2(0) \\ H_2(\frac{N}{8}) \\ H_3(0) \\ H_3(\frac{N}{8}) \end{bmatrix}. \quad (2.3)$$

Центральные матрицы в (2.2) и (2.3) имеют вид

$$\begin{bmatrix} I & A & B & C \\ I & D & -B & E \\ I & -A & B & -C \\ I & -D & -B & -E \end{bmatrix},$$

где I, A, B, C, D, E - матрицы размерности 2×2 , I - единичная матрица. При этом в (2.3) $B=I$.

Эти матрицы могут быть выражены через произведение слабозаполненных матриц

$$\begin{vmatrix} I & O & I & O \\ O & I & O & I \\ I & O & -I & O \\ O & I & O & -I \end{vmatrix} \cdot \begin{vmatrix} I & I & O & O \\ I & -I & O & O \\ O & O & I & O \\ O & O & O & I \end{vmatrix} \cdot \begin{vmatrix} I & O & O & O \\ O & O & B & O \\ O & A & O & C \\ O & D & O & E \end{vmatrix},$$

где O - нулевая матрица размерности 2×2 .

Подставляя данное разложение в (2.2) и (2.3), получим результирующий алгоритм вычисления.

Граф-схема 16-точечного алгоритма БПХ по основанию 4 с прореживанием по времени представлена на рис.2.1. Анализ алгоритма показывает, что на первом этапе вычислений используются базовые операции, представляющие собой 4-точечное ДПХ. Все последующие этапы состоят из двух подэтапов. На первом подэтапе выполняются умножения на поворачивающие множители, при этом вычисления проводятся с использованием базовой операции, представленной на рис.1.2. На втором подэтапе используются базовые операции "а" и "в" для окончательных вычислений по (2.3) и (2.2) соответственно, граф-схемы которых представлены на рис. 2.2. и 2.3, а их общий вид - на рис.2.4, где входные и выходные точки базовой операции являются соответственно элементами преобразованной на первом подэтапе правой и левой матриц-столбцов в (2.2) и (2.3). Алгоритм имеет прямой порядок отсчетов на выходе и четверично-инверсный - на входе.

Для вывода алгоритма БПХ по основанию 4 с прореживанием по частоте разобьем входную последовательность $x(n)$, $n = \overline{0, N-1}$, на последовательности $x(n)$, $x(n + \frac{N}{4})$, $x(n + \frac{N}{2})$ и $x(n + \frac{3N}{4})$, $n = \overline{0, N/4 - 1}$. Тогда ДПХ входной последовательности $x(n)$ можно представить в виде

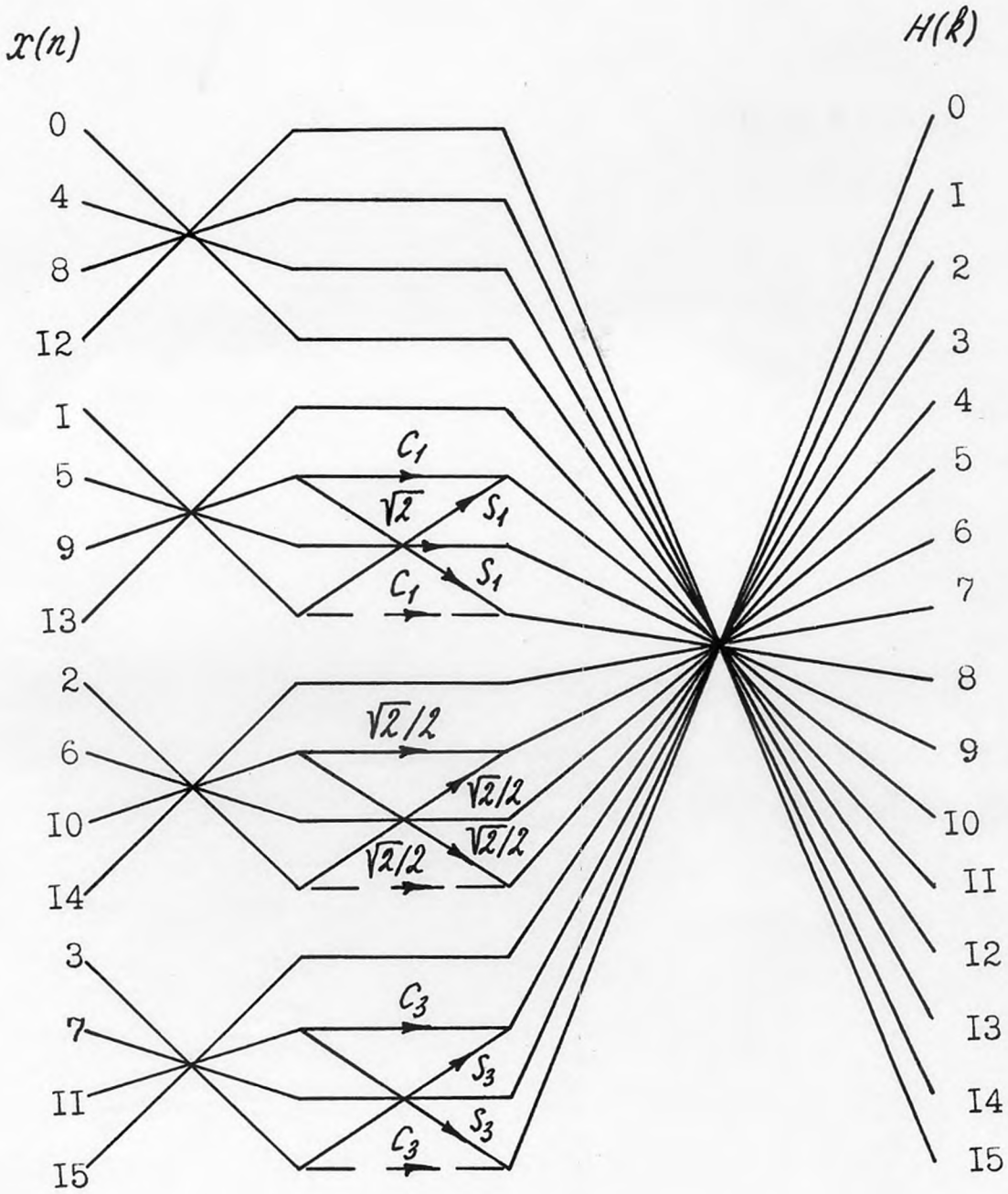


Рис.2.1. Граф-схема алгоритма БПХ по основанию 4 с прореживанием по времени

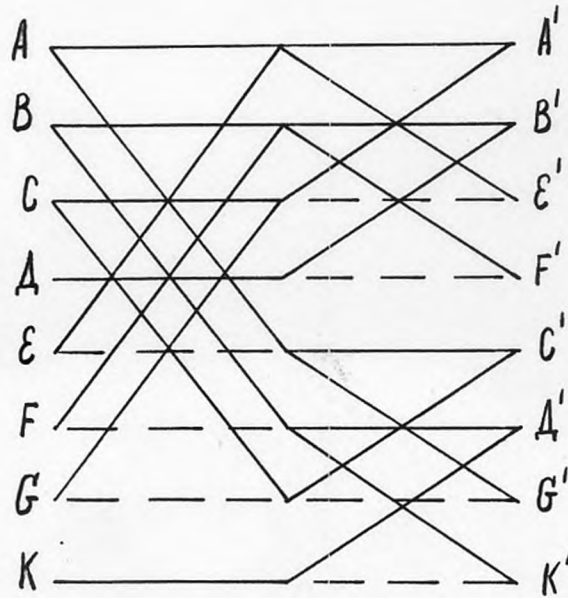


Рис. 2.2. Граф-схема базовой операции "а" в алгоритме БПХ по основанию 4 с прореживанием по времени

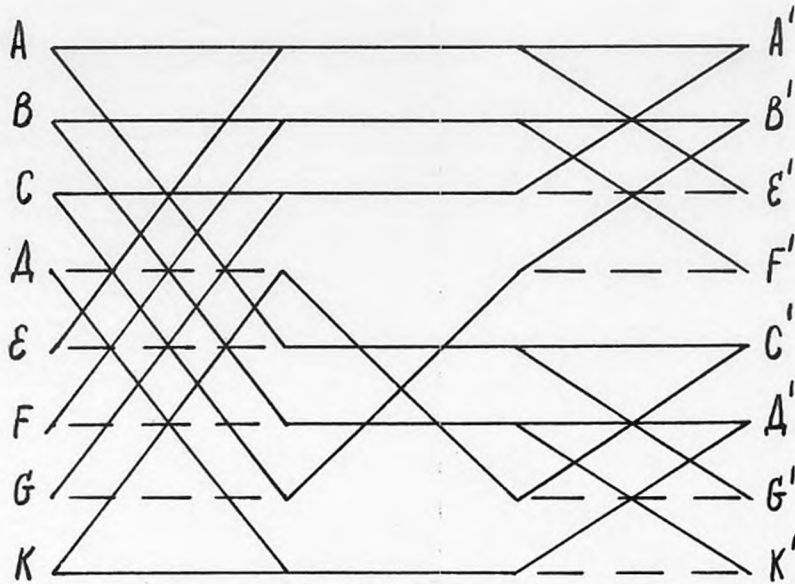


Рис. 2.3. Граф-схема базовой операции "в" в алгоритме БПХ по основанию 4 с прореживанием по времени

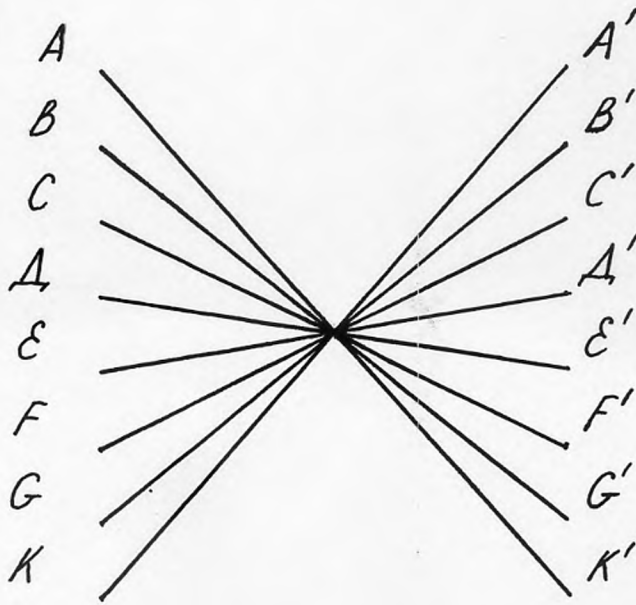


Рис.2.4. Базовые операции "а" и "в" в алгоритмах БПХ по основанию 4

$$\begin{aligned}
 H(k) = & \sum_{n=0}^{N/4-1} x(n) \cos \frac{2\pi nk}{N} + \sum_{n=0}^{N/4-1} x(n + \frac{N}{4}) \cos \frac{2\pi (n + \frac{N}{4})k}{N} + \sum_{n=0}^{N/4-1} x(n + \frac{N}{2}) \cos \frac{2\pi (n + \frac{N}{2})k}{N} \\
 & + \sum_{n=0}^{N/4-1} x(n + \frac{3N}{4}) \cos \frac{2\pi (n + \frac{3N}{4})k}{N} = \sum_{n=0}^{N/4-1} x(n) \cos \frac{2\pi nk}{N} + \\
 & + \cos \frac{\pi k}{2} \sum_{n=0}^{N/4-1} x(n + \frac{N}{4}) \cos \frac{2\pi nk}{N} + \sin \frac{\pi k}{2} \sum_{n=0}^{N/4-1} x(n + \frac{N}{4}) \sin \frac{2\pi nk}{N} + \\
 & + \cos \pi k \sum_{n=0}^{N/4-1} x(n + \frac{N}{2}) \cos \frac{2\pi nk}{N} + \sin \pi k \sum_{n=0}^{N/4-1} x(n + \frac{N}{2}) \sin \frac{2\pi nk}{N} + \\
 & + \cos \frac{3\pi k}{2} \sum_{n=0}^{N/4-1} x(n + \frac{3N}{4}) \cos \frac{2\pi nk}{N} + \sin \frac{3\pi k}{2} \sum_{n=0}^{N/4-1} x(n + \frac{3N}{4}) \sin \frac{2\pi nk}{N},
 \end{aligned}$$

где $\text{sas}(\cdot) = \cos(\cdot) - \sin(\cdot)$.

Введем следующие обозначения:

$$a(n) = x(n) + x(n + \frac{N}{4}) + x(n + \frac{N}{2}) + x(n + \frac{3N}{4}),$$

$$b(n) = x(n) - x(n + \frac{N}{4}) + x(n + \frac{N}{2}) - x(n + \frac{3N}{4}),$$

$$c(n) = x(n) - x(n + \frac{N}{2}), \quad d(n) = x(n + \frac{N}{4}) - x(n + \frac{3N}{4}).$$

Тогда, используя выше приведенное выражение для вычисления $H(k)$, получим:

$$H(4k) = \sum_{n=0}^{N/4-1} a(n) \cos \frac{2\pi nk}{N/4},$$

$$H(4k+1) = \sum_{n=0}^{N/4-1} [c(n) \cos \frac{2\pi n}{N}] \cos \frac{2\pi nk}{N/4} + \sum_{n=0}^{N/4-1} [c(n) \sin \frac{2\pi n}{N}] \text{sas} \frac{2\pi nk}{N/4} +$$

$$+ \sum_{n=0}^{N/4-1} [d(n) \cos \frac{2\pi n}{N}] \text{sas} \frac{2\pi nk}{N/4} - \sum_{n=0}^{N/4-1} [d(n) \sin \frac{2\pi n}{N}] \cos \frac{2\pi nk}{N/4},$$

$$H(4k+2) = \sum_{n=0}^{N/4-1} [b(n) \cos \frac{2\pi \cdot 2n}{N}] \cos \frac{2\pi nk}{N/4} + \sum_{n=0}^{N/4-1} [b(n) \sin \frac{2\pi \cdot 2n}{N}] \text{sas} \frac{2\pi nk}{N/4},$$

$$H(4k+3) = \sum_{n=0}^{N/4-1} [c(n) \cos \frac{2\pi \cdot 3n}{N}] \cos \frac{2\pi nk}{N/4} + \sum_{n=0}^{N/4-1} [c(n) \sin \frac{2\pi \cdot 3n}{N}] \text{sas} \frac{2\pi nk}{N/4} -$$

$$- \sum_{n=0}^{N/4-1} [d(n) \cos \frac{2\pi \cdot 3n}{N}] \text{sas} \frac{2\pi nk}{N/4} + \sum_{n=0}^{N/4-1} [d(n) \sin \frac{2\pi \cdot 3n}{N}] \cos \frac{2\pi nk}{N/4},$$

$$k = \overline{0, N/4-1}.$$

Если в приведенных выше выражениях во всех слагаемых, содержащих множитель $\cos(2\pi nk/(N/4))$, поменять порядок суммирования, то получим:

$$H(4k) = \sum_{n=0}^{N/4-1} [a(n)] \cos \frac{2\pi nk}{N/4},$$

$$H(4k+1) = \sum_{n=0}^{N/4-1} \left\{ [c(n) + c(\frac{N}{4}-n)] \cos \frac{2\pi n}{N} + [d(\frac{N}{4}-n) - d(n)] \sin \frac{2\pi n}{N} \right\} \cos \frac{2\pi nk}{N/4},$$

$$H(4k+2) = \sum_{n=0}^{N/4-1} [b(n) \cos \frac{2\pi \cdot 2n}{N} + b(\frac{N}{4}-n) \sin \frac{2\pi \cdot 2n}{N}] \cos \frac{2\pi nk}{N/4},$$

$$H(4k+3) = \sum_{n=0}^{N/4-1} \left\{ [c(n) - c(\frac{N}{4}-n)] \cos \frac{2\pi \cdot 3n}{N} + [d(n) + d(\frac{N}{4}-n)] \sin \frac{2\pi \cdot 3n}{N} \right\} \cos \frac{2\pi nk}{N/4},$$

$$k = \overline{0, N/4-1}.$$

(2.4)

Выражения (2.4) определяют вид алгоритма БПХ по основанию 4 с прореживанием по частоте.

Вычисления по (2.4) сводятся к вычислению значений функций, стоящих в квадратных скобках под знаками сумм. Обозначив эти функции как $X_i(n)$, $n = \overline{0, N/4-1}$ и $i = \overline{0, 3}$, их можно выразить в матричной форме следующим образом:

$$\begin{bmatrix} X_0(n) \\ X_1(n) \\ X_2(n) \\ X_3(n) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ C_n & C_n & -S_n & S_n & -C_n & -C_n & S_n & -S_n \\ C_{2n} & S_{2n} & -C_{2n} & -S_{2n} & C_{2n} & S_{2n} & -C_{2n} & -S_{2n} \\ C_{3n} & -C_{3n} & S_{3n} & S_{3n} & -C_{3n} & C_{3n} & -S_{3n} & -S_{3n} \end{bmatrix} \cdot \begin{bmatrix} x(n) \\ x(\frac{N}{4}-n) \\ x(n+\frac{N}{4}) \\ x(\frac{N}{2}-n) \\ x(n+\frac{N}{2}) \\ x(\frac{3N}{4}-n) \\ x(n+\frac{3N}{4}) \\ x(N-n) \end{bmatrix},$$

где $n = \overline{0, N/4-1}$.

Как видно, вычисления $X_i(n)$ выполняются без замещения.

Для выполнения вычислений с замещением необходимо вычислять также $X_i(\frac{N}{4}-n)$. Тогда получим следующий алгоритм вычислений:

$$\begin{bmatrix} X_0(n) \\ X_0(\frac{N}{4}-n) \\ X_1(n) \\ X_1(\frac{N}{4}-n) \\ X_2(n) \\ X_2(\frac{N}{4}-n) \\ X_3(n) \\ X_3(\frac{N}{4}-n) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ \hline C_n & C_n & -S_n & S_n & -C_n & -C_n & S_n & -S_n \\ S_n & S_n & C_n & -C_n & -S_n & -S_n & -C_n & C_n \\ \hline C_{2n} & S_{2n} & -C_{2n} & -S_{2n} & C_{2n} & S_{2n} & -C_{2n} & -S_{2n} \\ S_{2n} & -C_{2n} & -S_{2n} & C_{2n} & S_{2n} & -C_{2n} & -S_{2n} & C_{2n} \\ \hline C_{3n} & -C_{3n} & S_{3n} & S_{3n} & -C_{3n} & C_{3n} & -S_{3n} & -S_{3n} \\ S_{3n} & -S_{3n} & -C_{3n} & -C_{3n} & -S_{3n} & S_{3n} & C_{3n} & C_{3n} \end{bmatrix} \cdot \begin{bmatrix} x(n) \\ x(\frac{N}{4}-n) \\ x(n+\frac{N}{4}) \\ x(\frac{N}{2}-n) \\ x(n+\frac{N}{2}) \\ x(\frac{3N}{4}-n) \\ x(n+\frac{3N}{4}) \\ x(N-n) \end{bmatrix}, \quad (2.5)$$

где $n = \overline{0, N/8}$.

Для $n=0$ и $n=N/8$ вычисления по (2.5) могут быть объединены и примут следующий вид:

$$\begin{bmatrix} X_0(0) \\ X_0(\frac{N}{8}) \\ X_1(0) \\ X_1(\frac{N}{8}) \\ X_2(0) \\ X_2(\frac{N}{8}) \\ X_3(0) \\ X_3(\frac{N}{8}) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ \hline 1 & 0 & 1 & 0 & -1 & 0 & -1 & 0 \\ 0 & \sqrt{2} & 0 & 0 & 0 & -\sqrt{2} & 0 & 0 \\ \hline 1 & 0 & -1 & 0 & 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 & 0 & 1 & 0 & -1 \\ \hline 1 & 0 & -1 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & \sqrt{2} & 0 & 0 & 0 & -\sqrt{2} \end{bmatrix} \cdot \begin{bmatrix} x(0) \\ x(\frac{N}{8}) \\ x(\frac{N}{4}) \\ x(\frac{3N}{8}) \\ x(\frac{N}{2}) \\ x(\frac{5N}{8}) \\ x(\frac{3N}{4}) \\ x(\frac{7N}{8}) \end{bmatrix}. \quad (2.6)$$

Центральные матрицы в (2.5) и (2.6) имеют вид

$$\begin{bmatrix} I & I & I & I \\ A & B & -A & -B \\ C & -C & C & -C \\ D & E & -D & -E \end{bmatrix},$$

где I, A, B, C, D, E , - матрицы размерности 2×2 , I - единичная матрица. При этом в (2.6) $C=I$.

Эти матрицы могут быть выражены через произведение слаборазделенных матриц следующим образом:

$$\begin{bmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & C & 0 \\ 0 & 0 & 0 & I \end{bmatrix} \cdot \begin{bmatrix} I & I & 0 & 0 \\ 0 & 0 & A & B \\ I & -I & 0 & 0 \\ 0 & 0 & D & E \end{bmatrix} \cdot \begin{bmatrix} I & 0 & I & 0 \\ 0 & I & 0 & I \\ I & 0 & -I & 0 \\ 0 & I & 0 & -I \end{bmatrix}.$$

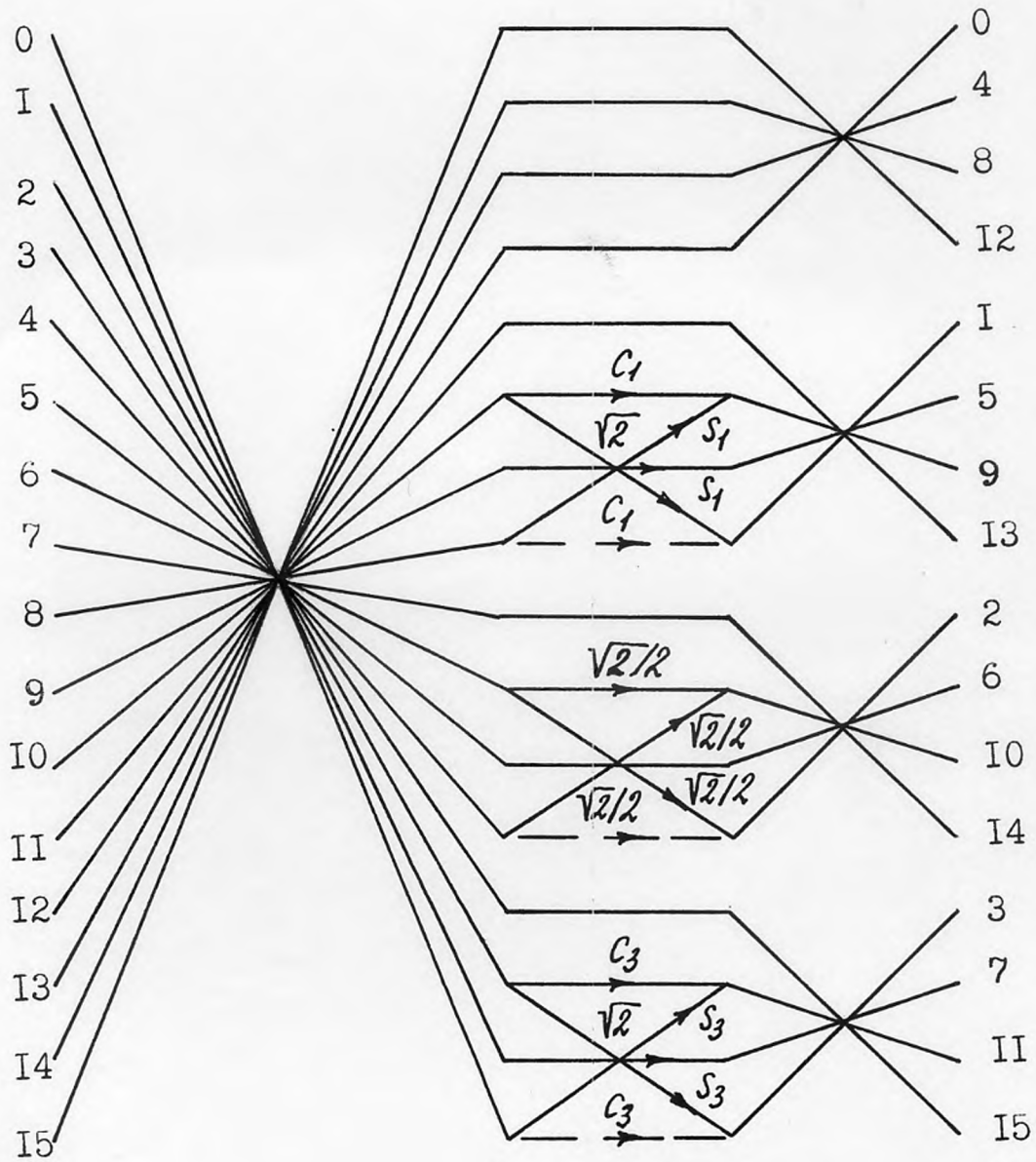


Рис.2.5. Граф-схема алгоритма БПХ по основанию 4 с прореживанием по частоте

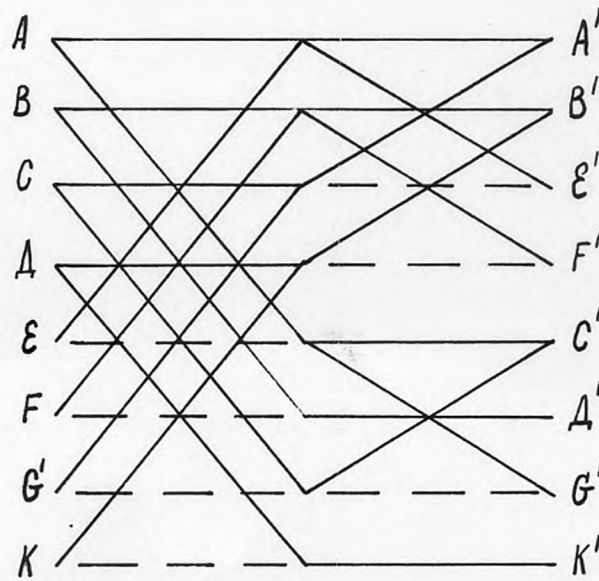


Рис. 2.6. Граф-схема базовой операции "а" в алгоритме БПХ по основанию 4 с прореживанием по частоте

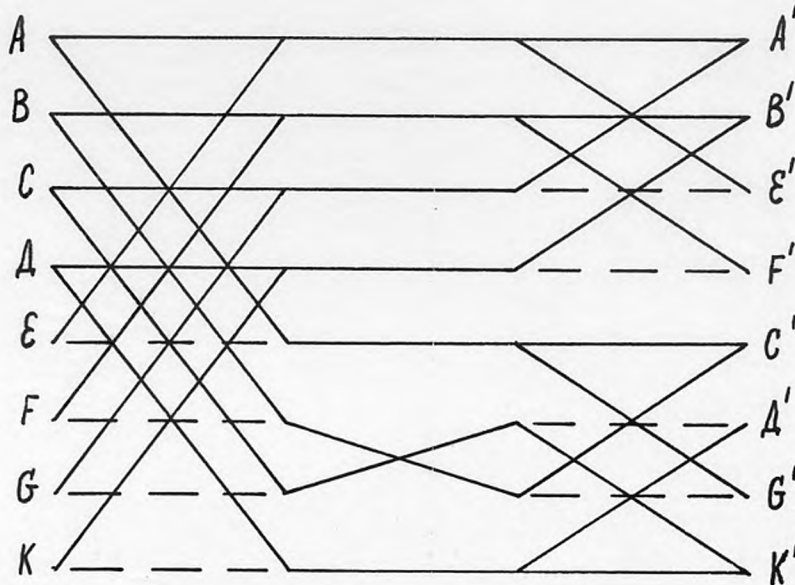


Рис. 2.7. Граф-схема базовой операции "в" в алгоритме БПХ по основанию 4 с прореживанием по частоте

Подставляя данное разложение в (2.5) и (2.6) получим результирующий алгоритм вычисления.

Граф-схема 16-точечного алгоритма БПХ по основанию 4 с прореживанием по частоте представлена на рис.2.5. Анализ алгоритма показывает, что на последнем этапе вычислений используются базовые операции, представляющие собой 4-точечное ДПХ. Все предыдущие этапы состоят из двух подэтапов. На первом подэтапе используются базовые операции "а" и "в" для предварительного вычисления по (2.6) и (2.5) соответственно. Их граф-схемы представлены соответственно на рис.2.6 и 2.7, а общий вид - на рис.2.4, где входные точки базовой операции являются элементами правой матрицы-столбца в (2.5) и (2.6), а выходные - исходными элементами для окончательного вычисления по (2.5) и (2.6) на втором подэтапе, на котором вычисления проводятся с использованием базовой операции, представленной на рис.1.2. Алгоритм имеет прямой порядок отсчетов на входе и четверично-инверсный - на выходе.

2.2. Алгоритмы быстрого преобразования Хартли по основанию 8

Для вывода алгоритма БПХ по основанию 8 с прореживанием по времени входная последовательность $x(n)$, $n = \overline{0, N-1}$, разбивается на последовательности $x(8n)$, $x(8n+1)$, $x(8n+2)$, $x(8n+3)$, $x(8n+4)$, $x(8n+5)$, $x(8n+6)$, $x(8n+7)$, $n = \overline{0, N/8-1}$. Далее осуществляется процедура, аналогичная процедуре вывода алгоритма БПХ по основанию 4 с прореживанием по времени. Опуская громоздкие промежуточные выкладки, окончательно ДПХ последовательности

$x(n)$ в матричной форме может быть выражено в виде (2.7),

где $H_i(k)$ представляют собой $N/8$ - точечные ДПХ последовательностей $x(8n+i)$, $i = \overline{0, 7}$; $k = \overline{0, N/8-1}$, $c_{[.]} = \cos \frac{2\pi[.]}{N}$, $s_{[.]} = \sin \frac{2\pi[.]}{N}$, $e_{[.]} = (c_{[.]} + s_{[.]}) \frac{\sqrt{2}}{2}$, $e'_{[.]} = (s_{[.]} - c_{[.]}) \frac{\sqrt{2}}{2}$.

(2.7)

$$\begin{aligned}
 & \left[\begin{array}{c} H(k) \\ H(k + \frac{N}{8}) \\ H(k + \frac{N}{4}) \\ H(k + \frac{3N}{8}) \\ H(k + \frac{N}{2}) \\ H(k + \frac{5N}{8}) \\ H(k + \frac{3N}{4}) \\ H(k + \frac{7N}{8}) \end{array} \right] = \left[\begin{array}{c} 1 \quad C_k \quad S_k \quad C_{2k} \quad S_{2k} \quad C_{4k} \quad S_{4k} \quad C_{8k} \quad S_{8k} \quad C_{16k} \quad S_{16k} \quad C_{32k} \quad S_{32k} \\ 1 \quad -C_k \quad C_k \quad -S_k \quad S_k \quad -C_{2k} \quad C_{2k} \quad -S_{2k} \quad S_{2k} \quad -C_{4k} \quad C_{4k} \quad -S_{4k} \quad S_{4k} \\ 1 \quad -S_k \quad C_k \quad -C_{2k} \quad -S_{2k} \quad S_{2k} \quad -C_{4k} \quad -S_{4k} \quad C_{4k} \quad -C_{8k} \quad S_{8k} \quad -C_{16k} \quad -S_{16k} \\ 1 \quad -C_k \quad -C_k \quad S_{2k} \quad -S_{2k} \quad -C_{4k} \quad -C_{4k} \quad S_{8k} \quad -S_{8k} \quad -C_{16k} \quad -C_{16k} \quad S_{32k} \quad -S_{32k} \\ 1 \quad C_k \quad -C_k \quad -S_{2k} \quad S_{2k} \quad -C_{4k} \quad C_{4k} \quad -S_{8k} \quad S_{8k} \quad -C_{16k} \quad C_{16k} \quad -S_{32k} \quad S_{32k} \\ 1 \quad S_k \quad -C_k \quad -C_{2k} \quad -S_{2k} \quad S_{2k} \quad -C_{4k} \quad -S_{4k} \quad C_{4k} \quad -C_{8k} \quad S_{8k} \quad -C_{16k} \quad -S_{16k} \\ 1 \quad C_k \quad C_k \quad S_{2k} \quad -S_{2k} \quad -C_{4k} \quad -C_{4k} \quad -S_{8k} \quad -S_{8k} \quad -C_{16k} \quad -C_{16k} \quad -S_{32k} \quad -S_{32k} \end{array} \right] \begin{array}{c} H_0(k) \\ H_1(k) \\ H_2(k) \\ H_3(k) \\ H_4(k) \\ H_5(k) \\ H_6(k) \\ H_7(k) \\ H_8(k) \end{array}
 \end{aligned}$$

Для выполнения алгоритма с замещением необходимо проводить вычисления согласно (2.8) для $k = \overline{0, N/8}$, которое для $k = 0$ и $k = N/16$ сводится к (2.9), где $t_{[.]} = C_{[.]} + S_{[.]}$, $v_{[.]} = S_{[.]} - C_{[.]}$.

Центральные матрицы в (2.8) и (2.9) имеют вид

$$\begin{bmatrix} I & A & B & C & D & E & F & G \\ I & K & L & M & -D & N & Q & P \\ I & R & -B & S & D & T & -F & X \\ I & Y & -L & Z & -D & W & -Q & J \\ I & -A & B & -C & D & -E & F & -G \\ I & -K & L & -M & -D & -N & Q & -P \\ I & -R & -B & -S & D & -T & -F & -X \\ I & -Y & -L & -Z & -D & -W & -Q & -J \end{bmatrix},$$

где $I, A, B, C, D, E, F, G, K, L, M, N, Q, P, R, S, T, X, Y, Z, W, J$ - матрицы размерности 2×2 , I - единичная матрица. При этом в (2.9) $D = I$.

Эти матрицы могут быть представлены в виде произведения слабозаполненных матриц следующим образом:

$$\begin{bmatrix} I & 0 & 0 & 0 & I & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 & I & 0 & 0 \\ 0 & 0 & I & 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & I & 0 & 0 & 0 & I \\ I & 0 & 0 & 0 & -I & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 & -I & 0 & 0 \\ 0 & 0 & I & 0 & 0 & 0 & -I & 0 \\ 0 & 0 & 0 & I & 0 & 0 & 0 & -I \end{bmatrix} \cdot \begin{bmatrix} I & 0 & I & 0 & 0 & 0 & 0 & 0 \\ 0 & I & 0 & I & 0 & 0 & 0 & 0 \\ I & 0 & -I & 0 & 0 & 0 & 0 & 0 \\ 0 & I & 0 & -I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & I \end{bmatrix}^x$$

$H(k)$	C_k	S_k	C_{2k}	S_{2k}	C_{3k}	S_{3k}	C_{4k}	S_{4k}	C_{5k}	S_{5k}	C_{6k}	S_{6k}	C_{7k}	S_{7k}	$H_0(k)$
$H(\frac{N}{8}-k)$	$-z^k$	z^k	C_{2k}	S_{2k}	z^k	z^k	S_{4k}	C_{4k}	z^k	z^k	$-C_{6k}$	S_{6k}	$-z^k$	z^k	$H_0(\frac{N}{8}-k)$
$H(k+\frac{N}{8})$	z^k	$-z^k$	$-S_{2k}$	C_{2k}	$-z^k$	$-z^k$	$-C_{4k}$	S_{4k}	z^k	z^k	S_{6k}	$-C_{6k}$	z^k	z^k	$H_1(k)$
$H(\frac{N}{4}-k)$	C_k	S_k	S_{2k}	C_{2k}	$-C_{3k}$	$-S_{3k}$	$-S_{4k}$	C_{4k}	C_{5k}	S_{5k}	S_{6k}	$-C_{6k}$	$-C_{7k}$	S_{7k}	$H_1(\frac{N}{8}-k)$
$H(k+\frac{N}{4})$	S_k	$-z^k$	$-C_{2k}$	S_{2k}	S_{3k}	C_{3k}	C_{4k}	S_{4k}	$-S_{5k}$	$-C_{5k}$	$-C_{6k}$	S_{6k}	z^k	z^k	$H_2(k)$
$H(\frac{3N}{8}-k)$	z^k	z^k	S_{2k}	C_{2k}	z^k	z^k	$-C_{4k}$	S_{4k}	z^k	z^k	$-S_{6k}$	C_{6k}	z^k	z^k	$H_2(\frac{N}{8}-k)$
$H(k+\frac{3N}{8})$	$-z^k$	$-z^k$	$-S_{2k}$	$-C_{2k}$	$-z^k$	$-z^k$	$-C_{4k}$	$-S_{4k}$	z^k	z^k	z^k	$-C_{6k}$	z^k	z^k	$H_3(k)$
$H(\frac{N}{2}-k)$	S_k	$-C_k$	$-S_{2k}$	C_{2k}	$-C_{3k}$	S_{3k}	$-S_{4k}$	$-C_{4k}$	S_{5k}	$-C_{5k}$	$-S_{6k}$	C_{6k}	S_{7k}	$-C_{7k}$	$H_3(\frac{N}{8}-k)$
$H(k+\frac{N}{2})$	$-z^k$	z^k	C_{2k}	S_{2k}	$-C_{3k}$	$-S_{3k}$	C_{4k}	S_{4k}	$-C_{5k}$	$-S_{5k}$	C_{6k}	S_{6k}	$-C_{7k}$	z^k	$H_4(k)$
$H(\frac{5N}{8}-k)$	z^k	z^k	$-S_{2k}$	$-C_{2k}$	z^k	z^k	$-C_{4k}$	$-S_{4k}$	z^k	z^k	z^k	$-C_{6k}$	z^k	z^k	$H_4(\frac{N}{8}-k)$
$H(k+\frac{5N}{8})$	$-z^k$	$-z^k$	$-S_{2k}$	$-C_{2k}$	$-z^k$	$-z^k$	$-C_{4k}$	$-S_{4k}$	z^k	z^k	z^k	$-C_{6k}$	z^k	z^k	$H_5(k)$
$H(\frac{3N}{4}-k)$	C_k	S_k	C_{2k}	S_{2k}	$-S_{3k}$	$-C_{3k}$	$-S_{4k}$	$-C_{4k}$	C_{5k}	S_{5k}	S_{6k}	$-C_{6k}$	C_{7k}	S_{7k}	$H_5(\frac{N}{8}-k)$
$H(k+\frac{3N}{4})$	S_k	$-z^k$	$-C_{2k}$	C_{2k}	z^k	z^k	$-C_{4k}$	S_{4k}	z^k	z^k	z^k	$-C_{6k}$	z^k	z^k	$H_6(k)$
$H(\frac{7N}{8}-k)$	z^k	z^k	$-C_{2k}$	$-S_{2k}$	z^k	z^k	$-C_{4k}$	$-S_{4k}$	z^k	z^k	z^k	$-C_{6k}$	z^k	z^k	$H_6(\frac{N}{8}-k)$
$H(k+\frac{7N}{8})$	$-z^k$	$-z^k$	$-C_{2k}$	$-S_{2k}$	$-z^k$	$-z^k$	$-C_{4k}$	$-S_{4k}$	z^k	z^k	z^k	$-C_{6k}$	z^k	z^k	$H_7(k)$
$H(N-k)$	S_k	$-C_k$	$-S_{2k}$	C_{2k}	$-S_{3k}$	$-C_{3k}$	$-S_{4k}$	$-C_{4k}$	C_{5k}	S_{5k}	S_{6k}	$-C_{6k}$	C_{7k}	S_{7k}	$H_7(\frac{N}{8}-k)$

(2.8)

$$x \begin{bmatrix} I & I & 0 & 0 & 0 & 0 & 0 & 0 \\ I & -I & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & I \end{bmatrix} \cdot \begin{bmatrix} I & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & D & 0 & 0 & 0 \\ 0 & 0 & B & 0 & 0 & 0 & F & 0 \\ 0 & 0 & L & 0 & 0 & 0 & Q & 0 \\ 0 & A & 0 & C & 0 & E & 0 & G \\ 0 & K & 0 & M & 0 & N & 0 & P \\ 0 & R & 0 & S & 0 & T & 0 & X \\ 0 & Y & 0 & Z & 0 & W & 0 & J \end{bmatrix},$$

где O - нулевая матрица размерности 2×2 .

Подставляя данное разложение в (2.8) и (2.9) получим результирующий алгоритм вычисления.

Граф-схема 64-точечного алгоритма БПХ по основанию 8 с прореживанием по времени представлена на рис.2.8. Анализ алгоритма показывает, что на первом этапе вычислений используются базовые операции, представляющие собой 8-точечное ДПХ. Все последующие этапы состоят из двух подэтапов. На первом подэтапе вычисления проводятся с использованием базовой операции, представленной на рис.1.2. На втором подэтапе используются базовые операции "а" и "в" для вычислений по (2.9) и (2.8) соответственно, граф-схемы которых представлены на рис.2.9 и 2.10, а их общий вид - на рис.2.11, где входные и выходные точки базовой операции являются соответственно элементами преобразованной на первом подэтапе правой и левой матриц-столбцов в (2.8) и (2.9). Алгоритм имеет восьмерично-инверсный порядок отсчетов на входе и прямой - на выходе.

Для вывода алгоритма БПХ по основанию 8 с прореживанием по частоте входная последовательность $x(n), n = \overline{0, N-1}$ разбивается на последовательности $x(n + \frac{N}{8}i), i = \overline{0, 7}$ и $n = \overline{0, N/8-1}$.

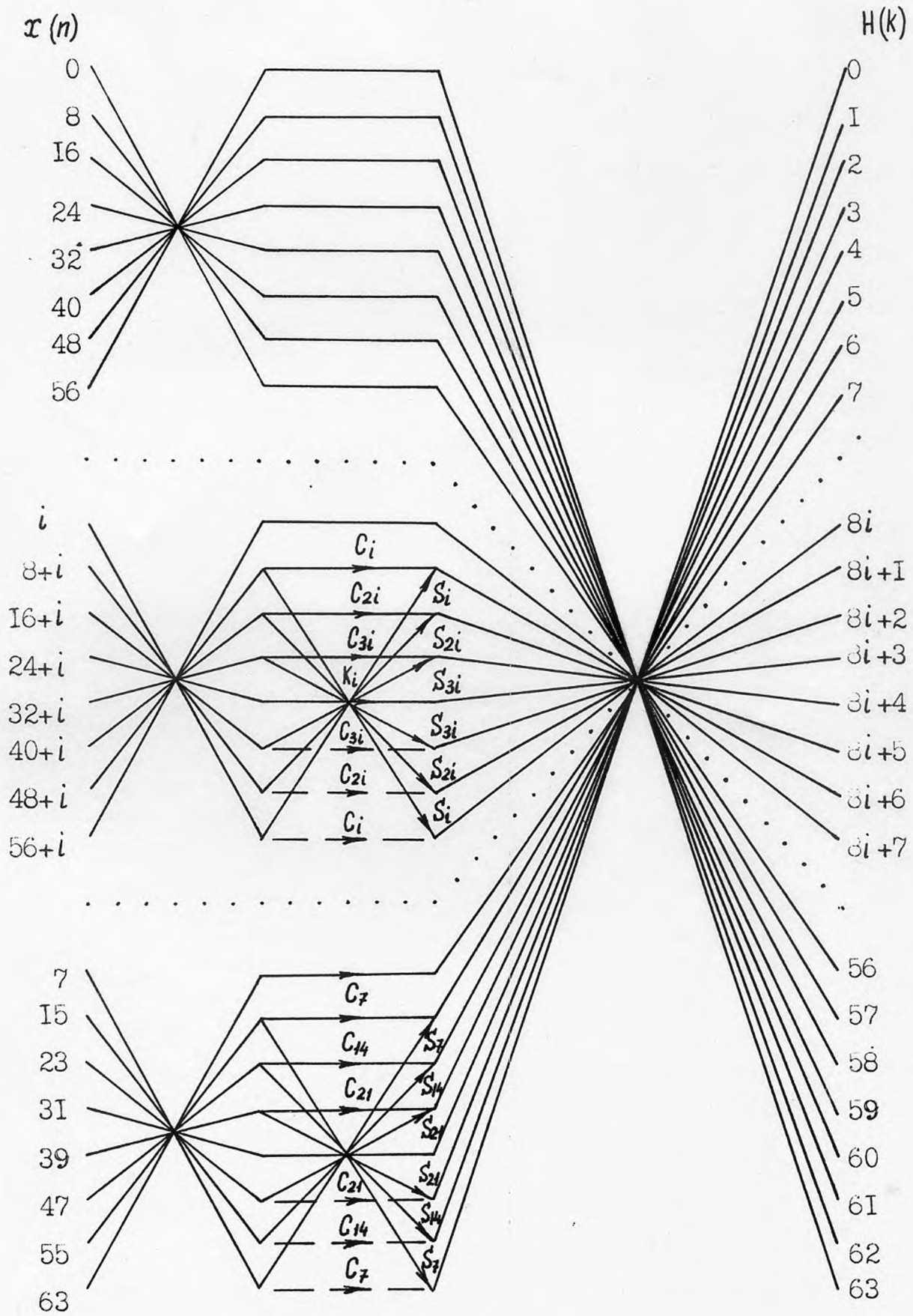


Рис. 2.8. Граф-схема 64-точечного алгоритма БПХ по ос-
нованию 8 с прореживанием по времени $K_i = \begin{cases} 1, i=1,3,5 \\ \sqrt{2}, i=2,6 \end{cases}$

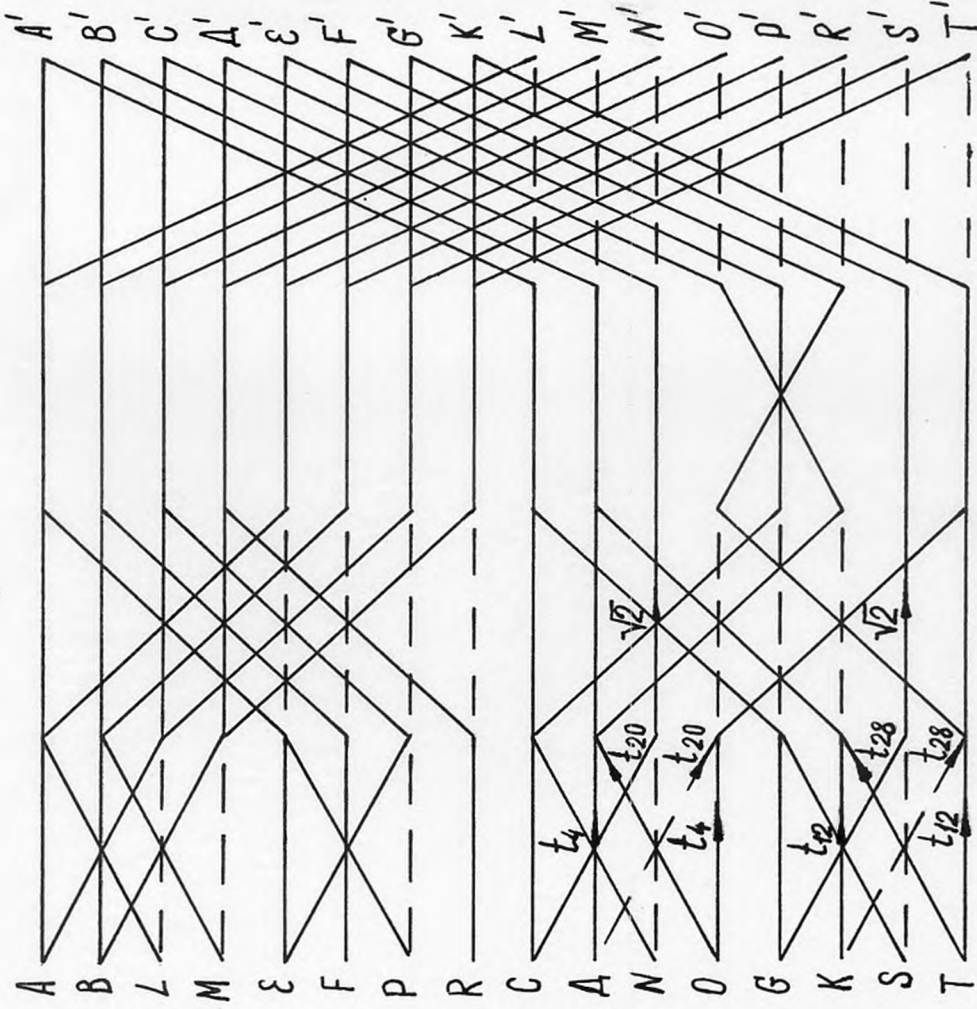


Рис. 2.9. Граф-схема базовой операции "а" в алгоритме БПХ по основанию 8 с прореживанием по времени

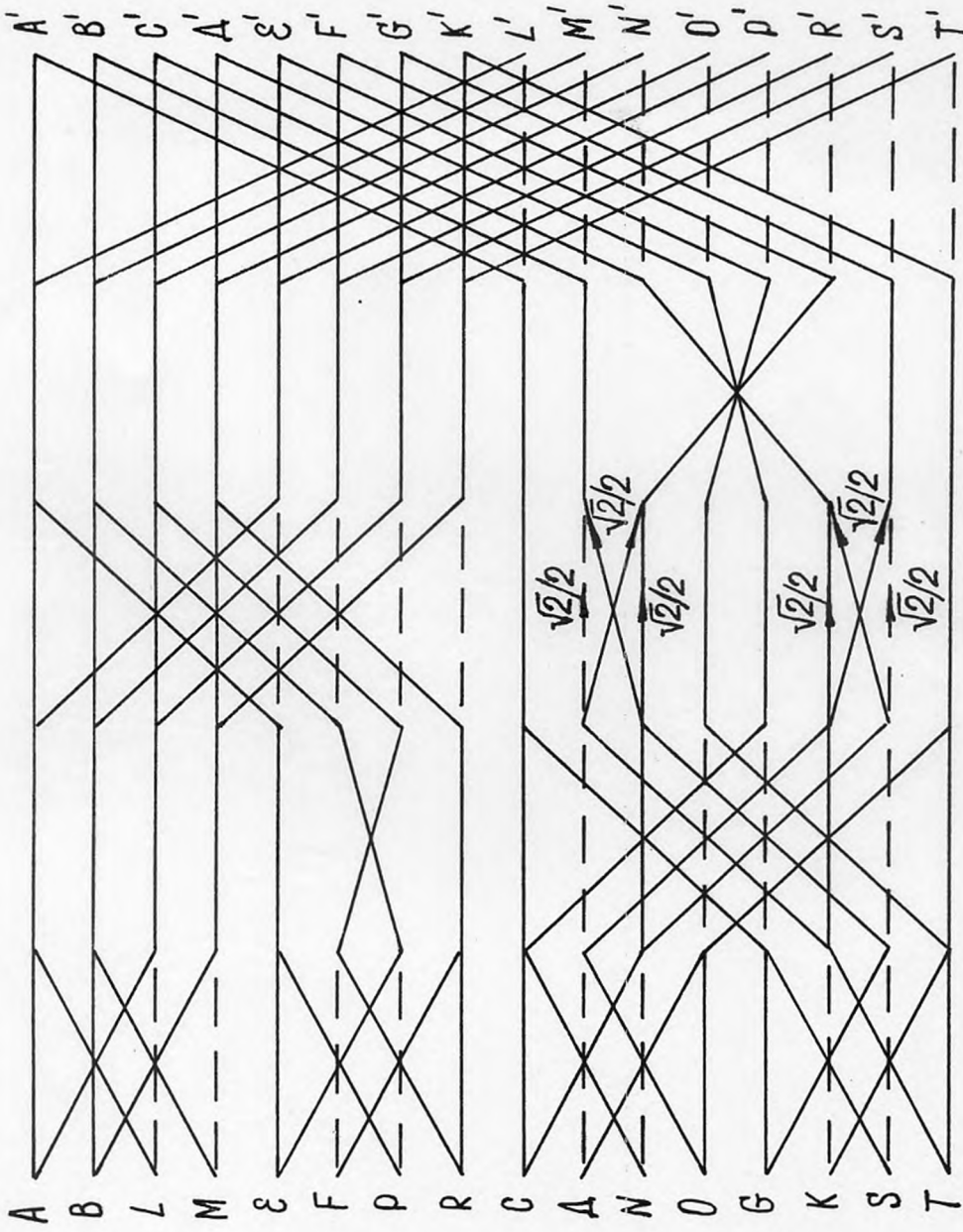


Рис. 2.10. Граф-схема базовой операции "в" в алгоритме БПХ по основанию 8 с прореживанием по времени

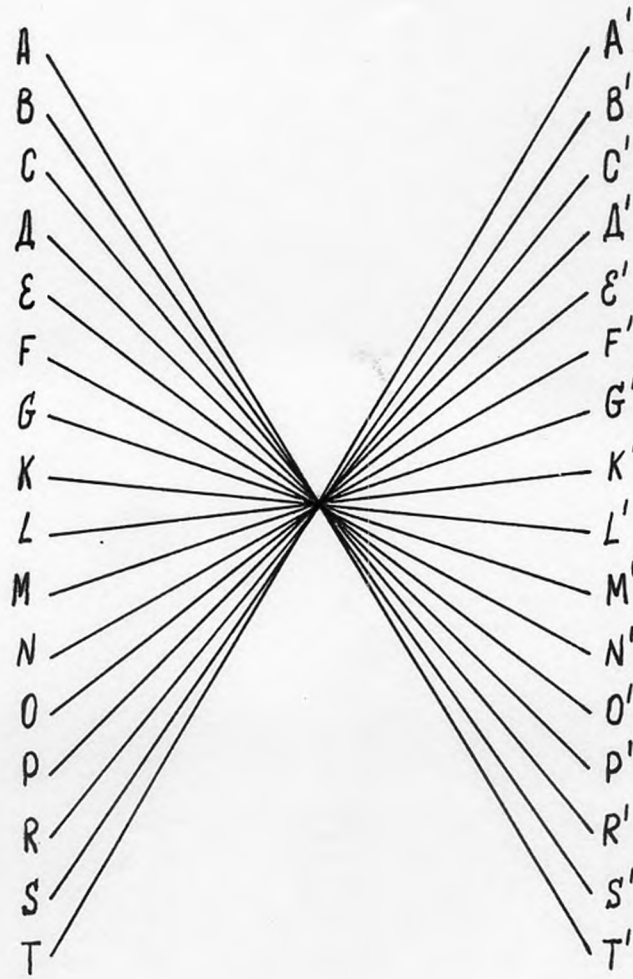


Рис. 2.11. Базовая операция в алгоритмах БПХ по основанию 8

Далее процедура вывода аналогична процедуре вывода алгоритма БПХ по основанию 4 с прореживанием по частоте. Окончательно ДПХ исходной последовательности $x(n)$ имеет следующий вид:

$$\begin{aligned}
 H(8k) &= \sum_{n=0}^{N/8-1} [a(n)] \cos \frac{2\pi nk}{N/8}, \\
 H(8k+1) &= \sum_{n=0}^{N/8-1} \left\{ [e(n)+f(\frac{N}{8}-n)+[f(n)-g(n)+e(\frac{N}{8}-n)+k(\frac{N}{8}-n)]\frac{\sqrt{2}}{2}] \cos \frac{2\pi n}{N} + \right. \\
 &\quad \left. + [[k(\frac{N}{8}-n)-e(\frac{N}{8}-n)-f(n)-g(n)]\frac{\sqrt{2}}{2} + g(\frac{N}{8}-n)-k(n)] \sin \frac{2\pi n}{N} \right\} \cos \frac{2\pi nk}{N/8}, \\
 H(8k+2) &= \sum_{n=0}^{N/8-1} \left\{ [c(n)+c(\frac{N}{8}-n)] \cos \frac{2\pi \cdot 2n}{N} + [d(\frac{N}{8}-n)-d(n)] \sin \frac{2\pi \cdot 2n}{N} \right\} \cos \frac{2\pi nk}{N/8}, \\
 H(8k+3) &= \sum_{n=0}^{N/8-1} \left\{ [e(n)-f(\frac{N}{8}-n)+[e(\frac{N}{8}-n)+k(\frac{N}{8}-n)-f(n)+g(n)]\frac{\sqrt{2}}{2}] \cos \frac{2\pi \cdot 3n}{N} + \right. \\
 &\quad \left. + [[e(\frac{N}{8}-n)-k(\frac{N}{8}-n)-f(n)-g(n)]\frac{\sqrt{2}}{2} + g(\frac{N}{8}-n)+k(n)] \sin \frac{2\pi \cdot 3n}{N} \right\} \cos \frac{2\pi nk}{N/8}, \\
 H(8k+4) &= \sum_{n=0}^{N/8-1} [b(n) \cos \frac{2\pi \cdot 4n}{N} + b(\frac{N}{8}-n) \sin \frac{2\pi \cdot 4n}{N}] \cos \frac{2\pi nk}{N/8}, \\
 H(8k+5) &= \sum_{n=0}^{N/8-1} \left\{ [e(n)+f(\frac{N}{8}-n)-[f(n)-g(n)+e(\frac{N}{8}-n)+k(\frac{N}{8}-n)]\frac{\sqrt{2}}{2}] \cos \frac{2\pi \cdot 5n}{N} + \right.
 \end{aligned}$$

$$+ [g(\frac{N}{8}-n) - k(n) - [k(\frac{N}{8}-n) - e(\frac{N}{8}-n) - f(n) - g(n)] \frac{\sqrt{2}}{2}] \sin \frac{2\pi \cdot 5n}{N} \} \cos \frac{2\pi nk}{N/8},$$

$$H(8k+6) = \sum_{n=0}^{N/8-1} \{ [c(n) - c(\frac{N}{8}-n)] \cos \frac{2\pi \cdot 6n}{N} + [d(\frac{N}{8}-n) + d(n)] \sin \frac{2\pi \cdot 6n}{N} \} \cos \frac{2\pi nk}{N/8},$$

$$H(8k+7) = \sum_{n=0}^{N/8-1} \{ [e(n) - f(\frac{N}{8}-n) - [e(\frac{N}{8}-n) + k(\frac{N}{8}-n) - f(n) + g(n)] \frac{\sqrt{2}}{2}] \cos \frac{2\pi \cdot 7n}{N} +$$

$$+ [g(\frac{N}{8}-n) + k(n) - [e(\frac{N}{8}-n) - k(\frac{N}{8}-n) - f(n) - g(n)] \frac{\sqrt{2}}{2}] \sin \frac{2\pi \cdot 7n}{N} \} \cos \frac{2\pi nk}{N/8}, \quad (2.10)$$

где

$$\begin{aligned} a(n) &= x(n) + x(n + \frac{N}{8}) + x(n + \frac{N}{4}) + x(n + \frac{3N}{8}) + x(n + \frac{N}{2}) + x(n + \frac{5N}{8}) + x(n + \frac{3N}{4}) + x(n + \frac{7N}{8}), \\ b(n) &= x(n) - x(n + \frac{N}{8}) + x(n + \frac{N}{4}) - x(n + \frac{3N}{8}) + x(n + \frac{N}{2}) - x(n + \frac{5N}{8}) + x(n + \frac{3N}{4}) - x(n + \frac{7N}{8}), \\ c(n) &= x(n) - x(n + \frac{N}{4}) + x(n + \frac{N}{2}) - x(n + \frac{3N}{4}), \quad d(n) = x(n + \frac{N}{8}) - x(n + \frac{3N}{8}) + x(n + \frac{5N}{8}) - x(n + \frac{7N}{8}), \\ e(n) &= x(n) - x(n + \frac{N}{2}), \quad f(n) = x(n + \frac{N}{8}) - x(n + \frac{5N}{8}), \quad g(n) = x(n + \frac{3N}{8}) - x(n + \frac{7N}{8}), \\ k(n) &= x(n + \frac{N}{4}) - x(n + \frac{3N}{4}). \end{aligned}$$

Выражения (2.10) определяет вид алгоритма БПХ по основанию 8 с прореживанием по частоте.

Вычисления по (2.10) сводятся к вычислению значений функций, стоящих в квадратных скобках под знаками сумм.

Обозначив эти функции как $X_i(n)$, $n = \overline{0, N/8-1}$ и $i = \overline{0, 7}$, их можно выразить в матричной форме в виде (2.11), где $n = \overline{0, N/8-1}$. Для выполнения вычислений с замещением их следует проводить согласно (2.12) для $n = \overline{0, N/16}$, которое для $n=0$ и $n=N/16$ сводится к (2.13).

$$\begin{bmatrix} X_0(n) \\ X_1(n) \\ X_2(n) \\ X_3(n) \\ X_4(n) \\ X_5(n) \\ X_6(n) \\ X_7(n) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ C_n - E_n - E_n & C_n - S_n - L_n & S_n - C_n & S_n - C_n & E_n & E_n & -C_n & S_n - L_n & S_n \\ C_{2n} & C_{2n} - S_{2n} - C_{2n} & S_{2n} - C_{2n} & S_{2n} - C_{2n} & C_{2n} & C_{2n} & -S_{2n} & S_{2n} - C_{2n} & S_{2n} - S_{2n} \\ C_{3n} & L_{3n} - L_{3n} & -C_{3n} & S_{3n} - E_{3n} & -L_{3n} & L_{3n} & C_{3n} - S_{3n} & E_{3n} & E_{3n} - S_{3n} \\ C_{4n} & S_{4n} - C_{4n} & -S_{4n} & C_{4n} & S_{4n} - C_{4n} & S_{4n} & -S_{4n} & C_{4n} & S_{4n} - C_{4n} - S_{4n} \\ C_{5n} & E_{5n} & E_{5n} & C_{5n} - S_{5n} & L_{5n} & S_{5n} - C_{5n} & -E_{5n} & -C_{5n} & S_{5n} & L_{5n} - S_{5n} \\ C_{6n} & -C_{6n} & S_{6n} & S_{6n} - C_{6n} & C_{6n} & -S_{6n} & C_{6n} & S_{6n} - C_{6n} & C_{6n} & -S_{6n} - S_{6n} \\ C_{7n} & -L_{7n} & L_{7n} & -C_{7n} & S_{7n} & E_{7n} & S_{7n} - C_{7n} & L_{7n} & C_{7n} & -S_{7n} & -E_{7n} & -S_{7n} \end{bmatrix} \begin{bmatrix} x(n) \\ x(\frac{N}{8} - n) \\ x(n + \frac{N}{8}) \\ x(\frac{N}{4} - n) \\ x(n + \frac{N}{4}) \\ x(\frac{3N}{8} - n) \\ x(n + \frac{3N}{8}) \\ x(\frac{N}{2} - n) \\ x(n + \frac{N}{2}) \\ x(\frac{5N}{8} - n) \\ x(n + \frac{5N}{8}) \\ x(\frac{3N}{4} - n) \\ x(n + \frac{3N}{4}) \\ x(\frac{7N}{8} - n) \\ x(n + \frac{7N}{8}) \\ x(N - n) \end{bmatrix}$$

(2.11)

$$\begin{bmatrix}
 \chi_0(0) \\
 \chi_0(\frac{N}{16}) \\
 \chi_1(0) \\
 \chi_1(\frac{N}{16}) \\
 \chi_2(0) \\
 \chi_2(\frac{N}{16}) \\
 \chi_3(0) \\
 \chi_3(\frac{N}{16}) \\
 \chi_4(0) \\
 \chi_4(\frac{N}{16}) \\
 \chi_5(0) \\
 \chi_5(\frac{N}{16}) \\
 \chi_6(0) \\
 \chi_6(\frac{N}{16}) \\
 \chi_7(0) \\
 \chi_7(\frac{N}{16})
 \end{bmatrix}
 =
 \begin{bmatrix}
 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\
 1 & 0 & \sqrt{2} & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & \frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \\
 1 & 0 & 1 & 0 & -1 & 0 & -1 & 0 & 1 & 0 & -1 & 0 & -1 & 0 & 1 & 0 \\
 0 & \sqrt{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 1 & 0 & 0 & 0 & -1 & 0 & \sqrt{2} & 0 & -1 & 0 & 0 & 0 & -\sqrt{2} & 0 & 1 & 0 \\
 0 & \frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} \\
 1 & 0 & -1 & 0 & 1 & 0 & -1 & 0 & 1 & 0 & -1 & 0 & 1 & 0 & -1 & 0 \\
 0 & 1 & 0 & -1 & 0 & 1 & 0 & -1 & 0 & 1 & 0 & -1 & 0 & 1 & 0 & -1 \\
 1 & 0 & -\sqrt{2} & 0 & 1 & 0 & 0 & 0 & -1 & 0 & \sqrt{2} & 0 & -1 & 0 & 0 & 0 \\
 0 & \frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \\
 1 & 0 & -1 & 0 & 1 & 0 & 1 & 0 & -1 & 0 & -1 & 0 & 1 & 0 & -1 & 0 \\
 0 & \frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} \\
 1 & 0 & 0 & 0 & -1 & 0 & -\sqrt{2} & 0 & 1 & 0 & 0 & 0 & -\sqrt{2} & 0 & 1 & 0 \\
 0 & \frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2}
 \end{bmatrix}
 \begin{bmatrix}
 x(0) \\
 x(\frac{N}{16}) \\
 x(\frac{N}{8}) \\
 x(\frac{3N}{16}) \\
 x(\frac{N}{4}) \\
 x(\frac{5N}{16}) \\
 x(\frac{3N}{8}) \\
 x(\frac{7N}{16}) \\
 x(\frac{N}{2}) \\
 x(\frac{9N}{16}) \\
 x(\frac{5N}{8}) \\
 x(\frac{11N}{16}) \\
 x(\frac{3N}{4}) \\
 x(\frac{13N}{16}) \\
 x(\frac{7N}{8}) \\
 x(\frac{15N}{16})
 \end{bmatrix}$$

Центральные матрицы в (2.12) и (2.13) имеют вид

$$\begin{bmatrix} I & I & I & I & I & I & I & I \\ A & B & C & D & -A & -B & -C & -D \\ E & F & -E & -F & E & F & -E & -F \\ G & J & K & L & -G & -J & -K & -L \\ M & -M & M & -M & M & -M & M & -M \\ N & Q & P & R & -N & -Q & -P & -R \\ S & T & -S & -T & S & T & -S & -T \\ X & Y & Z & W & -X & -Y & -Z & -W \end{bmatrix},$$

где $I, A, B, C, D, E, F, G, J, K, L, M, N, Q, P, R, S, T, X, Y, Z, W$ -матрицы размерности 2×2 , I - единичная матрица. Причем в (2.13) $M=I$.

Эти матрицы могут быть представлены в виде произведения слабозаполненных матриц

$$\begin{bmatrix} I & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & M & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & I \end{bmatrix} \cdot \begin{bmatrix} I & I & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & A & B & C & D \\ 0 & 0 & E & F & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & G & J & K & L \\ I & -I & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & N & Q & P & R \\ 0 & 0 & S & T & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & X & Y & Z & W \end{bmatrix}$$

на вторую и первую матрицы в разложении центральной матрицы в алгоритме БПХ по основанию 8 с прореживанием по времени.

Подставляя данное разложение в (2.12) и (2.13), получим результирующий алгоритм вычисления.

Граф-схема 64-точечного алгоритма БПХ по основанию 8 с про-

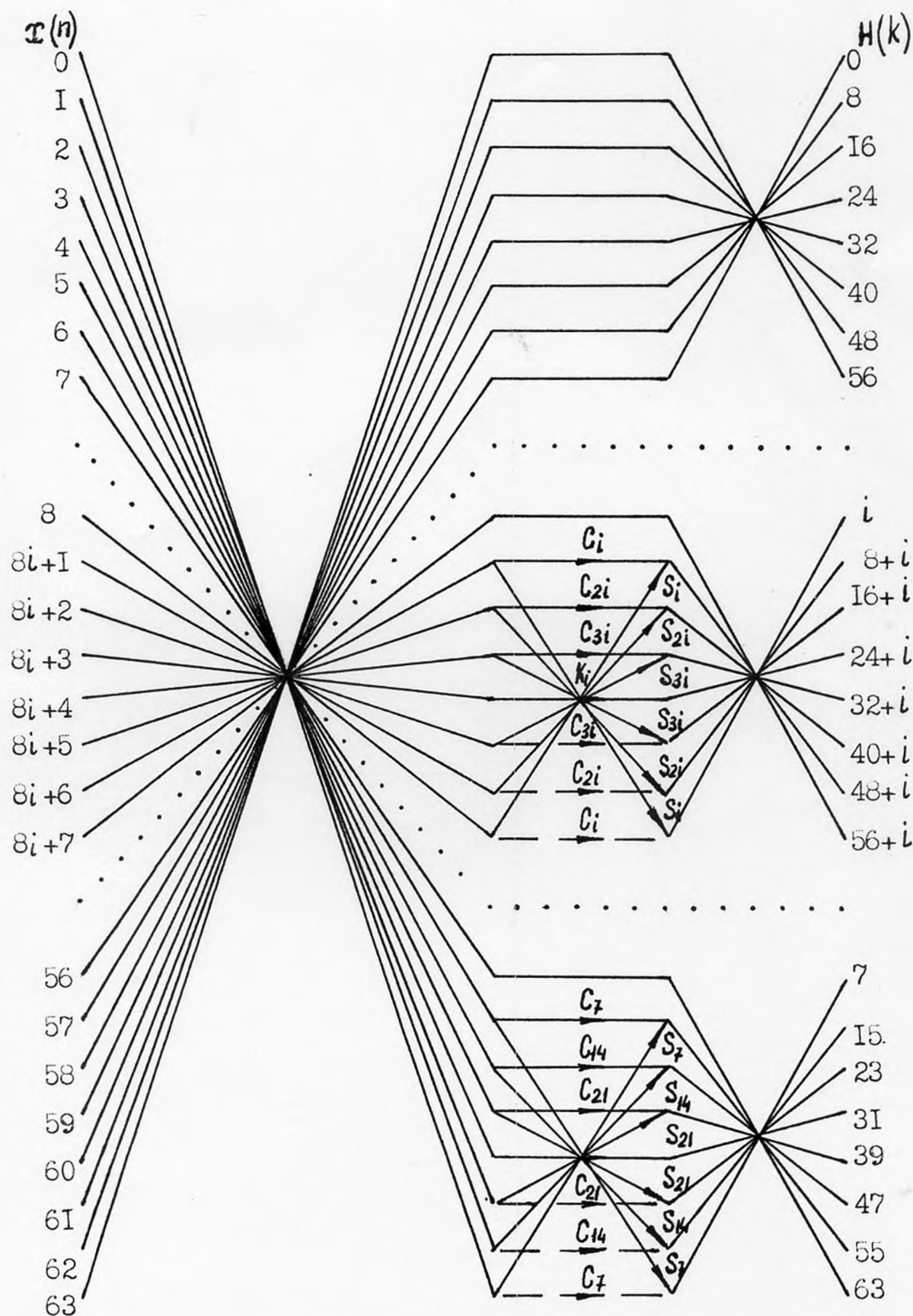


Рис. 2.12. Граф-схема 64-точечного алгоритма БПХ по основанию 8 с прореживанием по частоте $K_i = \begin{cases} 1, & i = 1, 3, 5 \\ \sqrt{2}, & i = 2, 6 \end{cases}$

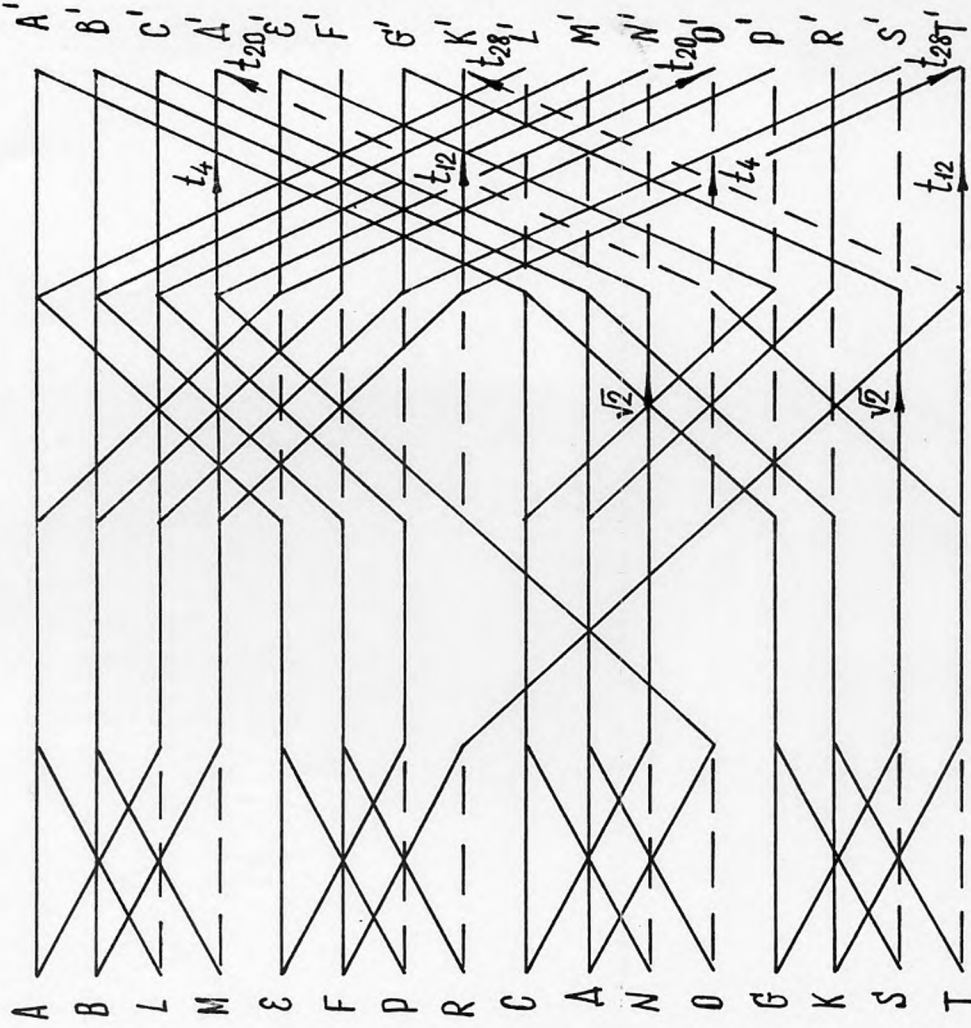


Рис. 2.13. Граф-схема базовой операции "а" в алгоритме БПХ по основанию 8 с прореживанием по частоте

реживанием по частоте представлена на рис.2.12. Анализ алгоритма показывает, что на последнем этапе вычислений используются базовые операции, представляющие собой 8-точечные ДПХ. Все предыдущие этапы состоят из двух подэтапов. На первом подэтапе используются базовые операции "а" и "в" для вычислений по (2.13) и (2.12) соответственно. Их граф-схемы представлены соответственно на рис.2.13 и 2.14, а общий вид - на рис.2.11, где входные точки базовой операции являются элементами правой матрицы-столбца в (2.12) и (2.13), а выходные - исходными элементами для окончательного вычисления по (2.12) и (2.13) на втором подэтапе, на котором вычисления проводятся с использованием базовой операции, приведенной на рис.1.2. Алгоритм имеет прямой порядок отсчетов на входе и восьмерично-инверсный - на выходе.

2.3. Алгоритмы быстрого преобразования Хартли с расщепленным основанием

Разобьем входную последовательность $x(n)$, $n = \overline{0, N-1}$, на последовательности $x(2n)$, $n = \overline{0, N/2-1}$; $x(4n+1)$, $x(4n+3)$, $n = \overline{0, N/4-1}$. Тогда ДПХ исходной последовательности $x(n)$ можно представить в виде

$$H(k) = H_0(k) + \cos \frac{2\pi k}{N} H_1(k) + \sin \frac{2\pi k}{N} H_1\left(\frac{N}{4} - k\right) + \\ + \cos \frac{2\pi \cdot 3k}{N} H_2(k) + \sin \frac{2\pi \cdot 3k}{N} H_2\left(\frac{N}{4} - k\right),$$

где

$$H_0(k) = \sum_{n=0}^{N/2-1} x(2n) \cos \frac{2\pi nk}{N/2}, \quad H_1(k) = \sum_{n=0}^{N/4-1} x(4n+1) \cos \frac{2\pi nk}{N/4},$$

$$H_2(k) = \sum_{n=0}^{N/4-1} x(4n+3) \cos \frac{2\pi nk}{N/4}, \quad k = \overline{0, N/4-1}.$$

Полное ДПХ последовательности $x(n)$ в матричной форме имеет следующий вид:

$$\begin{bmatrix} H(k) \\ H(k + \frac{N}{4}) \\ H(k + \frac{N}{2}) \\ H(k + \frac{3N}{4}) \end{bmatrix} = \begin{bmatrix} 1 & 0 & C_k & S_k & C_{3k} & S_{3k} \\ 0 & 1 & -S_k & C_k & S_{3k} & -C_{3k} \\ 1 & 0 & -C_k & -S_k & -C_{3k} & -S_{3k} \\ 0 & 1 & S_k & -C_k & -S_{3k} & C_{3k} \end{bmatrix} \cdot \begin{bmatrix} H_0(k) \\ H_0(k + \frac{N}{4}) \\ H_1(k) \\ H_1(\frac{N}{4} - k) \\ H_2(k) \\ H_2(\frac{N}{4} - k) \end{bmatrix}, \quad (2.14)$$

где $C_{[\cdot]} = \cos \frac{2\pi[\cdot]}{N}$, $S_{[\cdot]} = \sin \frac{2\pi[\cdot]}{N}$, $k = \overline{0, N/4-1}$.

(2.14) определяет вид алгоритма БПХ с расщеплённым основанием с прореживанием по времени.

Для выполнения вычислений с замещением их необходимо проводить следующим образом:

$$\begin{bmatrix} H(k) \\ H(\frac{N}{4} - k) \\ H(k + \frac{N}{4}) \\ H(\frac{N}{2} - k) \\ H(k + \frac{N}{2}) \\ H(\frac{3N}{4} - k) \\ H(k + \frac{3N}{4}) \\ H(N - k) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & C_k & S_k & C_{3k} & S_{3k} \\ 0 & 1 & 0 & 0 & C_k & S_k & -C_{3k} & -S_{3k} \\ \hline 0 & 0 & 1 & 0 & -S_k & C_k & S_{3k} & -C_{3k} \\ 0 & 0 & 0 & 1 & S_k & -C_k & S_{3k} & -C_{3k} \\ \hline 1 & 0 & 0 & 0 & -C_k & -S_k & -C_{3k} & -S_{3k} \\ 0 & 1 & 0 & 0 & -C_k & -S_k & C_{3k} & S_{3k} \\ \hline 0 & 0 & 1 & 0 & S_k & -C_k & -S_{3k} & C_{3k} \\ 0 & 0 & 0 & 1 & -S_k & C_k & -S_{3k} & C_{3k} \end{bmatrix} \cdot \begin{bmatrix} H_0(k) \\ H_0(\frac{N}{4} - k) \\ H_0(k + \frac{N}{4}) \\ H_0(\frac{N}{2} - k) \\ H_1(k) \\ H_1(\frac{N}{4} - k) \\ H_2(k) \end{bmatrix}. \quad (2.15)$$

Для $k=0$ и $k=N/8$ вычисления по (2.15) могут быть объединены и примут следующий вид:

$$\begin{bmatrix} H(0) \\ H(\frac{N}{8}) \\ H(\frac{N}{4}) \\ H(\frac{3N}{8}) \\ H(\frac{N}{2}) \\ H(\frac{5N}{8}) \\ H(\frac{3N}{4}) \\ H(\frac{7N}{8}) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & \sqrt{2} & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & \sqrt{2} \\ \hline 1 & 0 & 0 & 0 & -1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & -\sqrt{2} & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -\sqrt{2} \end{bmatrix} \cdot \begin{bmatrix} H_0(0) \\ H_0(\frac{N}{8}) \\ H_0(\frac{N}{4}) \\ H_0(\frac{3N}{8}) \\ H_1(0) \\ H_1(\frac{N}{8}) \\ H_2(0) \\ H_2(\frac{N}{8}) \end{bmatrix}. \quad (2.16)$$

Центральные матрицы в (2.15) и (2.16) имеют вид

$$\begin{bmatrix} I & 0 & A & B \\ 0 & I & C & D \\ I & 0 & -A & -B \\ 0 & I & -C & -D \end{bmatrix},$$

где $I, 0, A, B, C, D$ - матрицы размерности 2×2 ,
 I - единичная матрица, 0 - нулевая матрица.

Эти матрицы могут быть представлены в виде произведения слабозаполненных матриц следующим образом:

$$\begin{bmatrix} I & 0 & I & 0 \\ 0 & I & 0 & I \\ I & 0 & -I & 0 \\ 0 & I & 0 & -I \end{bmatrix} \cdot \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & A & B \\ 0 & 0 & C & D \end{bmatrix}.$$

Подставляя данное разложение в (2.15) и (2.16), получим результирующий алгоритм БПХ с расщеплённым основанием с прореживанием по времени, граф-схема которого представлена на рис.2.15.

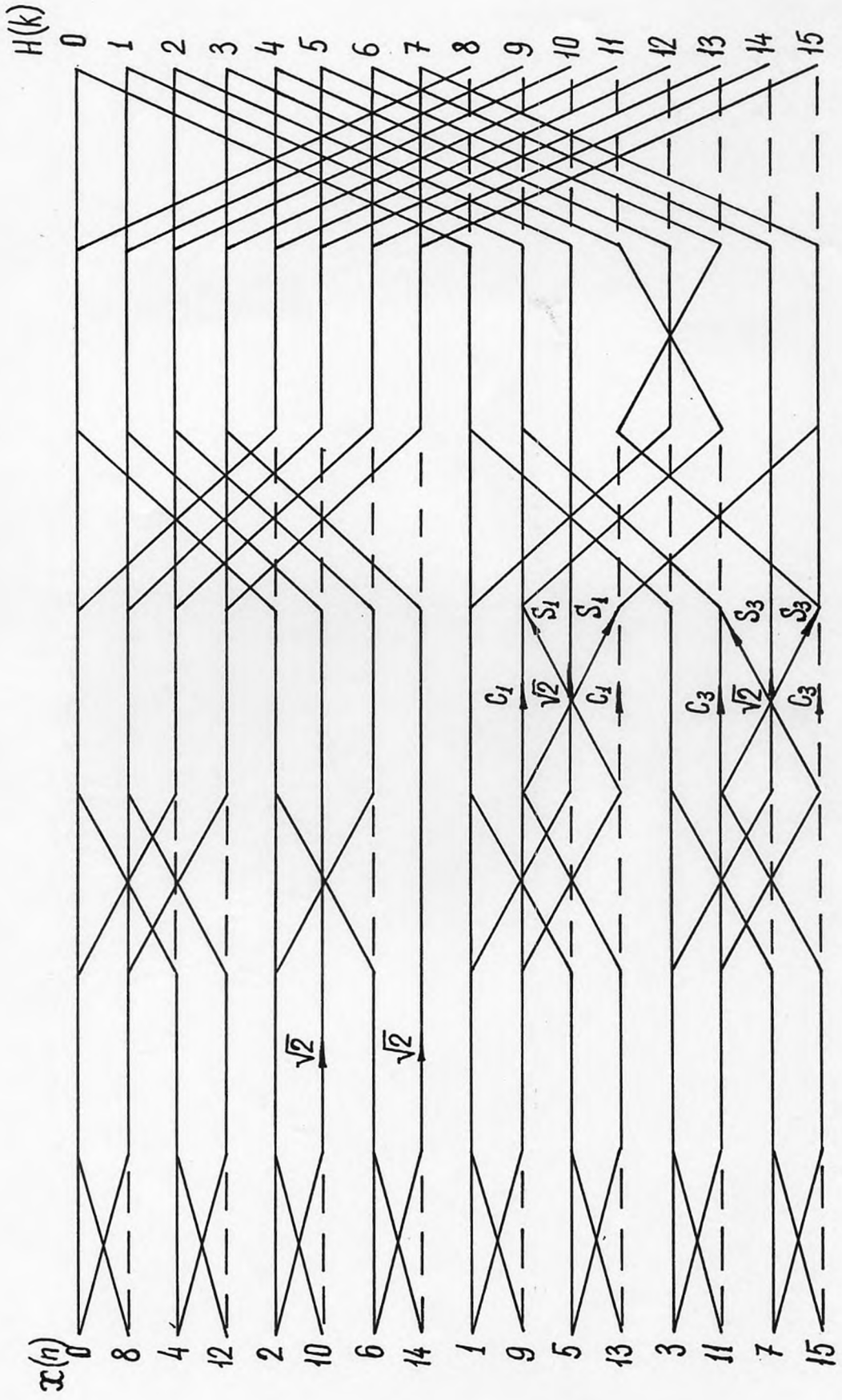


Рис. 2.15. Граф-схема алгоритма БПХ с расщепленным основанием с прореживанием по времени

Алгоритм имеет двоично-инверсный порядок отсчётов на входе и прямой - на выходе.

Для вывода алгоритма БПХ с расщеплённым основанием с прореживанием по частоте воспользуемся результатами вывода алгоритма БПХ по основанию 2 с прореживанием по частоте для определения чётных гармоник и алгоритма БПХ по основанию 4 с прореживанием по частоте для определения нечётных гармоник ДПХ. В итоге алгоритм БПХ с расщеплённым основанием с прореживанием по частоте примет следующий вид:

$$H(2k) = \sum_{n=0}^{N/2-1} [x(n) + x(n + \frac{N}{2})] \cos \frac{2\pi nk}{N/2}, \quad k = \overline{0, N/2-1};$$

$$H(4k+1) = \sum_{n=0}^{N/4-1} \{ [c(n) + c(\frac{N}{4}-n)] \cos \frac{2\pi n}{N} + [d(\frac{N}{4}-n) - d(n)] \sin \frac{2\pi n}{N} \} \cos \frac{2\pi nk}{N/4},$$

$$H(4k+3) = \sum_{n=0}^{N/4-1} \{ [c(n) - c(\frac{N}{4}-n)] \cos \frac{2\pi \cdot 3n}{N} + [d(n) + d(\frac{N}{4}-n)] \sin \frac{2\pi \cdot 3n}{N} \} \cos \frac{2\pi nk}{N/4},$$

где $c(n) = x(n) - x(n + \frac{N}{2})$, $d(n) = x(n + \frac{N}{4}) - x(n + \frac{3N}{4})$, $k = \overline{0, N/4-1}$.

Обозначив через $X_i(n)$, $i = \overline{0, 2}$, функции, стоящие в квадратных скобках под знаками сумм, их вычисление можно осуществить следующим образом:

$$\begin{bmatrix} X_0(n) \\ X_1(n) \\ X_2(n) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ C_n & C_n & -S_n & S_n & -C_n & -C_n & S_n & -S_n \\ C_{3n} & -C_{3n} & S_{3n} & S_{3n} & -C_{3n} & C_{3n} & -S_{3n} & -S_{3n} \end{bmatrix} \cdot \begin{bmatrix} x(n) \\ x(\frac{N}{4}-n) \\ x(n+\frac{N}{4}) \\ x(\frac{N}{2}-n) \\ x(n+\frac{N}{2}) \\ x(\frac{3N}{4}-n) \\ x(n+\frac{3N}{4}) \\ x(N-n) \end{bmatrix}.$$

Для выполнения вычислений с замещением их необходимо проводить следующим образом:

$$\begin{bmatrix} X_0(n) \\ X_0(\frac{N}{4}-n) \\ X_0(n+\frac{N}{4}) \\ X_0(\frac{N}{2}-n) \\ X_1(n) \\ X_1(\frac{N}{4}-n) \\ X_2(n) \\ X_2(\frac{N}{4}-n) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ \hline C_n & C_n & -S_n & S_n & -C_n & -C_n & S_n & -S_n \\ S_n & S_n & C_n & -C_n & -S_n & -S_n & -C_n & C_n \\ \hline C_{3n} & -C_{3n} & S_{3n} & S_{3n} & -C_{3n} & C_{3n} & -S_{3n} & -S_{3n} \\ S_{3n} & -S_{3n} & -C_{3n} & -C_{3n} & -S_{3n} & S_{3n} & C_{3n} & C_{3n} \end{bmatrix} \cdot \begin{bmatrix} x(n) \\ x(\frac{N}{4}-n) \\ x(n+\frac{N}{4}) \\ x(\frac{N}{2}-n) \\ x(n+\frac{N}{2}) \\ x(\frac{3N}{4}-n) \\ x(n+\frac{3N}{4}) \\ x(N-n) \end{bmatrix}. \quad (2.17)$$

Для $n=0$ и $n=N/8$ вычисления по (2.17) могут быть объединены и примут следующий вид:

$$\begin{bmatrix} X_0(0) \\ X_0(\frac{N}{8}) \\ X_0(\frac{N}{4}) \\ X_0(\frac{3N}{8}) \\ X_1(0) \\ X_1(\frac{N}{8}) \\ X_2(0) \\ X_2(\frac{N}{8}) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ \hline 1 & 0 & 1 & 0 & -1 & 0 & -1 & 0 \\ 0 & \sqrt{2} & 0 & 0 & 0 & -\sqrt{2} & 0 & 0 \\ \hline 1 & 0 & -1 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & \sqrt{2} & 0 & 0 & 0 & -\sqrt{2} \end{bmatrix} \cdot \begin{bmatrix} x(0) \\ x(\frac{N}{8}) \\ x(\frac{N}{4}) \\ x(\frac{3N}{8}) \\ x(\frac{N}{2}) \\ x(\frac{5N}{8}) \\ x(\frac{3N}{4}) \\ x(\frac{7N}{8}) \end{bmatrix}. \quad (2.18)$$

Центральные матрицы в (2.17) и (2.18) имеют вид

$$\begin{bmatrix} I & 0 & I & 0 \\ 0 & I & 0 & I \\ A & B & -A & -B \\ C & D & -C & -D \end{bmatrix},$$

где $I, 0, A, B, C, D$ — матрицы размерности 2×2 ,

I — единичная матрица, 0 — нулевая матрица, и могут быть представлены в виде произведения слабозаполненных матриц следующим образом:

$$\begin{bmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & A & B \\ 0 & 0 & C & D \end{bmatrix} \cdot \begin{bmatrix} I & 0 & I & 0 \\ 0 & I & 0 & I \\ I & 0 & -I & 0 \\ 0 & I & 0 & -I \end{bmatrix}.$$

Подставляя данное разложение в (2.17) и (2.18), получим результирующий алгоритм БПХ с расщеплённым основанием с прорежива-

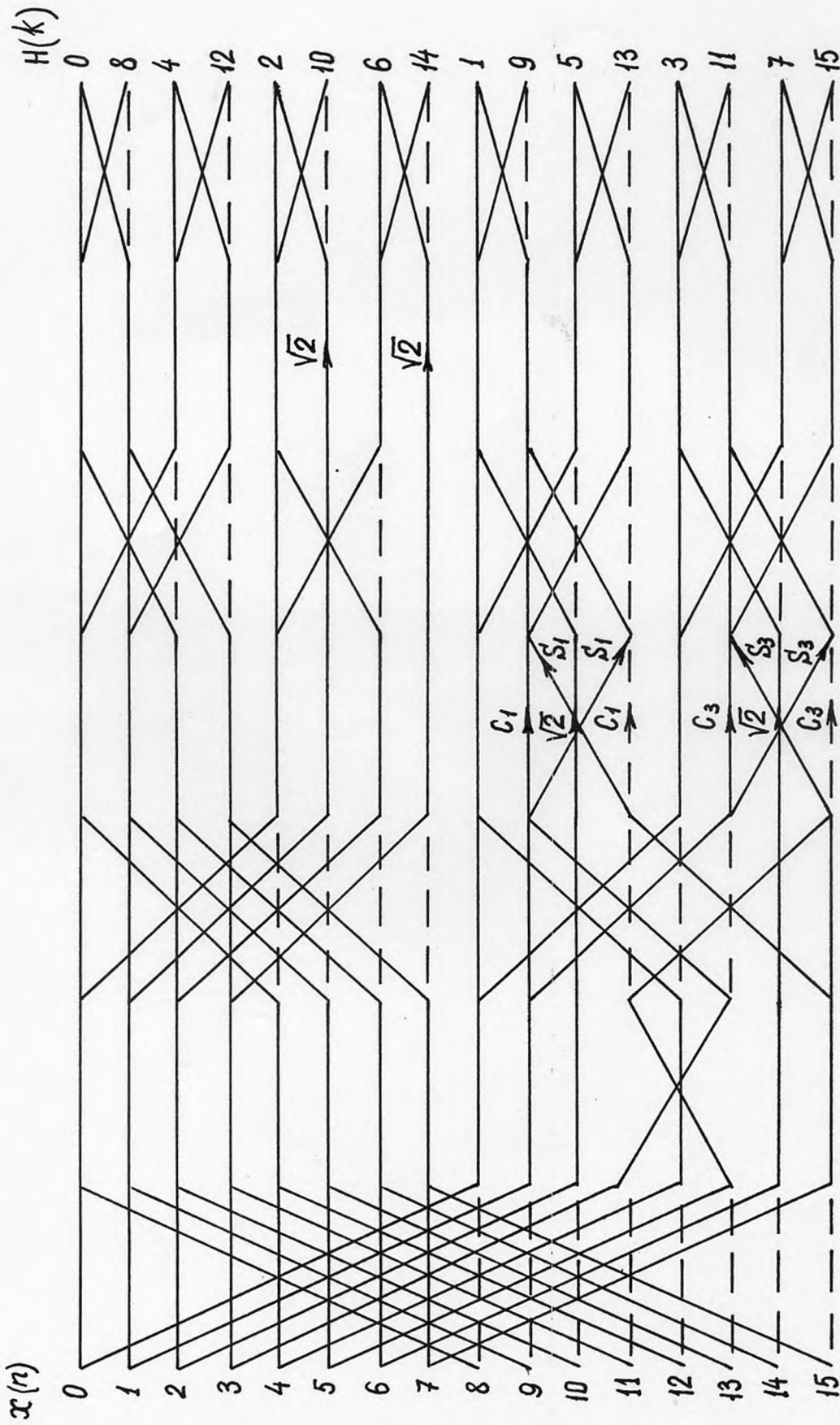


Рис. 2.16. Граф-схема алгоритма БПХ с расщепленным основанием с прореживанием по частоте

нием по частоте, граф-схема которого представлена на рис.2.16. Алгоритм имеет прямой порядок отсчётов на входе и двоично-инверсный - на выходе.

2.4. Сравнительный анализ алгоритмов быстрого преобразования Хартли по количеству арифметических операций и объёму требуемой памяти

Сравнительный анализ алгоритмов БПХ по количеству арифметических операций и объёму требуемой памяти приведен в [74].

Пусть $M_R(N)$ - число вещественных умножений и $A_R(N)$ число вещественных сложений N - точечного алгоритма БПХ. Учтём также, что базовая операция, приведенная на рис.1.2 и используемая в алгоритмах БПХ, требует 3 вещественных умножения и 3 вещественных сложения. Поскольку арифметическая сложность и объём требуемой памяти алгоритмов БПХ с прореживанием по времени и частоте одинаковы, анализ проведём для алгоритмов БПХ с прореживанием по времени.

Тогда арифметическая сложность алгоритмов БПХ по основанию 4 (БПХ-4) определяется следующим образом:

$$M_R(N) = 4M_R(N/4) + \frac{9}{8}N - 8,$$

$$A_R(N) = 4A_R(N/4) + \frac{25}{8}N - 12.$$

С учётом того, что $M_R(4) = 0$ и $A_R(4) = 8$, получим:

$$M_R(N) = \frac{9}{16}N \log_2 N - \frac{43}{24}N + \frac{8}{3},$$

$$A_R(N) = \frac{25}{16}N \log_2 N - \frac{17}{8}N + 4.$$

Арифметическая сложность алгоритмов БПХ по основанию 8 (БПХ-8) определяется следующим образом:

$$M_R(N) = 8M_R(N/8) + \frac{25}{16}N - 16,$$

$$A_R(N) = 8A_R(N/8) + \frac{73}{16}N - 28.$$

С учётом того, что $M_R(8) = 2$ и $A_R(8) = 26$, получим:

$$M_R(N) = \frac{25}{48}N \log_2 N - \frac{179}{112}N + \frac{16}{7},$$

$$A_R(N) = \frac{73}{48}N \log_2 N - \frac{29}{16}N + 4.$$

Арифметическая сложность алгоритмов БПХ с расщеплённым основанием (БПХРО) определяется следующим образом:

$$M_R(N) = M_R(N/2) + 2M_R(N/4) + \frac{3}{4}N - 4,$$

$$A_R(N) = A_R(N/2) + 2A_R(N/4) + \frac{9}{4}N - 8.$$

С учётом того, что $M_R(2) = M_R(4) = 0$ и $A_R(2) = 2$, $A_R(4) = 8$ получим:

$$M_R(N) = \frac{N}{2} \log_2 N - \frac{3}{2}N + 2,$$

$$A_R(N) = \frac{3}{2}N \log_2 N - \frac{13}{6}N + 4 + \frac{2 \cdot (-1)^{\log_2 N}}{3}.$$

Поскольку алгоритмы БПХ выполняются с замещением, то для хранения данных они требуют N ячеек ОЗУ, а для хранения поворачивающих множителей требуется $N/2 - 4$ или $3N/4 - 6$ ячеек

ПЗУ для алгоритмов БПХ-2 и БПХРО, $3N/4-8$ или $9N/8-12$ для алгоритмов БПХ-4 и $7N/8-16$ или $21N/16-24$ - для алгоритмов БПХ-8 соответственно в случае использования алгоритмов (I.15) или (I.14).

Количество арифметических операций алгоритмов БПХ с учётом результатов, приведённых в первой главе для алгоритма БПХ-2, для некоторых N приведено в табл.2.1. Объём требуемой памяти для алгоритмов БПХ совпадает с объёмом требуемой памяти алгоритмов БПФ вещественной последовательности первого класса.

Сравнительный анализ алгоритмов БПХ показывает, что с ростом основания арифметическая сложность алгоритмов уменьшается, а объём требуемой памяти увеличивается. Наименьшие арифметическую сложность и объём требуемой памяти имеют алгоритмы БПХРО, которые по количеству операций вещественного умножения и вещественного сложения обеспечивают выигрыш соответственно в 1,43 и 1,17 раза по сравнению с алгоритмами БПХ-2 для $N = 1024$.

Сравнительный анализ алгоритмов БПХ и соответствующих им алгоритмов БПФ показывает следующее:

- алгоритмы БПХ превосходят алгоритмы БПФ комплексной последовательности по количеству арифметических операций и объёму требуемой памяти в 2 и более раза;

- алгоритмы БПХ превосходят алгоритмы БПФ вещественной последовательности второго класса; при этом алгоритмы с расщеплённым основанием для $N = 1024$ обеспечивают выигрыш в 1,07, 1,15 и 1,33 раза соответственно по количеству операций вещественного умножения, вещественного сложения и объёму требуемой памяти;

- алгоритмы БПХ и алгоритмы БПФ вещественной последовательности первого класса имеют одинаковые характеристики по количеству операций вещественного умножения и объёму требуемой памяти, а по количеству операций вещественного сложения последние

Таблица 2.1

Арифметическая сложность алгоритмов БПХ

N	БПХ-2		БПХ-4		БПХ-8		БПХ-16	
	$MR(N)$	$AR(N)$	$MR(N)$	$AR(N)$	$MR(N)$	$AR(N)$	$MR(N)$	$AR(N)$
2	0	2					0	2
4	0	8	0	8			0	8
8	2	26			2	26	2	22
16	12	76	10	70			10	66
32	44	204					34	174
64	132	516	104	468	100	472	98	442
128	356	1252					258	1070
256	900	2948	696	2660			642	2522
512	2180	6788			1584	6084	1538	5806
1024	5124	15364	3928	13828			3586	13146
2048	11780	34308					8194	29358
4096	26628	75780	20312	68100	19056	67332	18434	64858

превосходят первые, что для алгоритмов с расцепленным основанием составляет $[N + 2 \cdot (-1)^{\log_2 N}] / 3$ операций и для $N = 1024$ - выигрыш в 1,03 раза.

Таким образом, алгоритмы БПХ превосходят алгоритмы БПФ комплексной последовательности, вещественной последовательности второго класса и имеют практически одинаковые характеристики по сравнению с алгоритмами БПФ вещественной последовательности первого класса, что делает их более предпочтительными в применении благодаря регулярности и простоте структуры.

2.5. Рекуррентные алгоритмы вычисления дискретного преобразования Хартли и энергетического спектра

Вывод рекуррентных алгоритмов вычисления ДПХ и энергетического спектра проведем на основе подхода, предложенного в [III].

Из анализа (I.1) и (I.3) видно, что ДПФ и ДПХ могут быть представлены в виде

$$X(k) = \sum_{n=0}^{N-1} x(n) \alpha(nk), \quad k = \overline{0, N-1}, \quad (2.19)$$

где $x(n)$ - последовательность во временной области, $X(k)$ - последовательность в частотной области преобразования, $\alpha(nk)$ - ядро преобразования.

Рассмотрим исходную последовательность во временной области $x_0(n)$, определенную в N точках, которая имеет вид

$$x(0), x(1), x(2), \dots, x(m-1), x(m), x(m+1), \dots, x(N-3), x(N-2), x(N-1).$$

Пусть данная последовательность имеет преобразование $X_0(k)$, которое известно.

При добавлении к последовательности $x_0(n)$ m новых членов и отбрасывании m начальных ее членов получим последовательность $x_1(n)$, которая имеет вид

$x(m), x(m+1), x(m+2), \dots, x(N-1), x(N), x(N+1), \dots, x(N-3+m), x(N-2+m), x(N-1+m)$.

Пусть последовательность $x_1(n)$ имеет преобразование $X_1(k)$, которое следует определить.

Выполним циклический сдвиг последовательности $x_1(n)$ на m позиций вправо, в результате чего получим последовательность $x_1^c(n)$, которая имеет вид

$x(N), x(N+1), x(N+2), \dots, x(N-1+m), x(m), x(m+1), \dots, x(N-3), x(N-2), x(N-1)$.

Последовательности $x_0(n)$, $x_1(n)$ и $x_1^c(n)$ представлены на рис.2.17.

Полученную последовательность $x_1^c(n)$ можно представить как сумму последовательностей

$x(0), x(1), x(2), \dots, x(m-1), x(m), x(m+1), \dots, x(N-3), x(N-2), x(N-1)$

и

$x(N)-x(0), x(N+1)-x(1), x(N+2)-x(2), \dots, x(N-1+m)-x(m-1), 0, 0, \dots, 0, 0, 0$.

В силу свойства линейности преобразования (2.19) преобразование последовательности $x_1^c(n)$ можно представить как сумму преобразований последовательностей, составляющих последовательность $x_1^c(n)$. С учетом того, что первая последовательность есть исходная последовательность $x_0(n)$, которая имеет преобразование $X_0(k)$, преобразование последовательности $x_1^c(n)$ можно определить как

$$X_1^c(k) = X_0(k) + \sum_{n=0}^{m-1} [x(N+n) - x(n)] \alpha(nk), \quad k = \overline{0, N-1}. \quad (2.20)$$

Если преобразование $X_1(k)$ можно выразить через преобразование $X_1^c(k)$, то получим рекуррентный алгоритм вычис-

ления данного преобразования, который в общем виде определяется как

$$X_1(k) = f \left\{ X_0(k) + \sum_{n=0}^{m-1} [x(N+n) - x(n)] d(nk) \right\}, \quad k = \overline{0, N-1}, \quad (2.21)$$

где f - некоторая функция, связывающая $X_1(k)$ и $X_1^c(k)$.

Используя (2.21) и учитывая свойства преобразования, в частности свойства циклического сдвига ДПФ и ДПХ, можно определить конкретный вид рекуррентных алгоритмов. Так рекуррентный алгоритм вычисления ДПФ примет вид

$$F_1(k) = [F_0(k) + \sum_{n=0}^{m-1} [x(N+n) - x(n)] W_N^{nk}] W_N^{-mk}, \quad k = \overline{0, N-1}, \quad (2.22)$$

который для $m = 1$ будет преобразован к виду

$$F_1(k) = [F_0(k) + [x(N) - x(0)]] W_N^{-k}, \quad k = \overline{0, N-1}, \quad (2.23)$$

где $F_1(k)$ - определяемое ДПФ, $F_0(k)$ - ДПФ на предыдущем интервале.

Алгоритмы, определяемые выражениями (2.22) и (2.23), совпадают с алгоритмами полученными в [53] и [52].

Рекуррентные алгоритмы вычисления ДПХ для произвольного m и $m = 1$ примут соответственно следующий вид:

$$H_1(k) = [H_0(k) + \sum_{n=0}^{m-1} [x(N+n) - x(n)] \cos \frac{2\pi nk}{N}] \cos \frac{2\pi mk}{N} - [H_0(N-k) + \sum_{n=0}^{m-1} [x(N+n) - x(n)] \cos \frac{2\pi n(N-k)}{N}] \sin \frac{2\pi mk}{N}, \quad k = \overline{0, N-1} \quad (2.24)$$

и

$$H_1(k) = [H_0(k) + [x(N) - x(0)]] \cos \frac{2\pi k}{N} - \\ - [H_0(N-k) + [x(N) - x(0)]] \sin \frac{2\pi k}{N}, \quad k = \overline{0, N-1}, \quad (2.25)$$

где $H_1(k)$ - определяемое ДПХ, $H_0(k)$ - ДПХ на предыдущем интервале.

Используя (2.22) - (2.23) и учитывая (1.6), можно получить рекуррентные алгоритмы вычисления энергетического спектра на базе ДПФ, которые для произвольного m и $m = 1$ определяются соответственно следующим образом:

$$|F_1(k)|^2 = |F_0(k) + \sum_{n=0}^{m-1} [x(N+n) - x(n)] W_N^{nk}|^2, \quad k = \overline{0, N-1} \quad (2.26)$$

и

$$|F_1(k)|^2 = |F_0(k) + [x(N) - x(0)]|^2, \quad k = \overline{0, N-1}. \quad (2.27)$$

Используя (2.24) - (2.25) и учитывая (1.7), можно получить рекуррентные алгоритмы вычисления энергетического спектра на базе ДПХ, которые для произвольного m и $m = 1$ определяются соответственно следующим образом:

$$|F_1(k)|^2 = \left[(H_0(k) + \sum_{n=0}^{m-1} [x(N+n) - x(n)] \cos \frac{2\pi nk}{N})^2 + \right. \\ \left. + (H_0(N-k) + \sum_{n=0}^{m-1} [x(N+n) - x(n)] \cos \frac{2\pi n(N-k)}{N})^2 \right] / 2, \quad k = \overline{0, N-1} \quad (2.28)$$

и

$$|F_r(k)|^2 = [(H_0(k) + [x(N) - x(0)])^2 + (H_0(N-k) + [x(N) - x(0)])^2] / 2, \quad k = \overline{0, N-1}. \quad (2.29)$$

Аналогичные рекуррентные алгоритмы можно разработать для так называемых теоретико-числовых преобразований (ТЧП), которые имеют вид

$$X(k) = \left\langle \sum_{n=0}^{N-1} x(n) \alpha^{nk} \right\rangle \bmod M, \quad k = \overline{0, N-1},$$

где модуль M и длина последовательности N не имеют общих сомножителей, N является делителем $\varphi(M)$, где φ - функция Эйлера, α - взаимно просто с M , $\langle \alpha^N \rangle \bmod M = 1$ и $\langle N \cdot N^{-1} \rangle \bmod M = 1$.

Рекуррентные алгоритмы вычисления ТЧП для произвольного m и $m = 1$ имеют соответственно следующий вид:

$$X_r(k) = \left\langle [X_0(k) + \sum_{n=0}^{m-1} [x(N+n) - x(n)] \alpha^{nk}] \alpha^{-mk} \right\rangle \bmod M, \quad k = \overline{0, N-1}$$

и

$$X_r(k) = \left\langle [X_0(k) + [x(N) - x(0)]] \alpha^{-k} \right\rangle \bmod M, \quad k = \overline{0, N-1}.$$

Данные алгоритмы можно использовать для вычисления свертки (корреляции), используя свойство циклической свертки (корреляции). При этом отсутствуют какие-либо ошибки вычисления, поскольку все вычисления проводятся в кольце целых чисел. Кроме того, при некоторых α (например $\alpha = 2$ или $\alpha = \sqrt{2}$) операции умножения заменяются операциями сдвига. Рекуррентный алгоритм вычисления ТЧП для $m = 1$ был использован при разработке устройства скользящего

спектрально-корреляционного анализа [112].

В [111] получены выражения для m , при которых вычисление по рекуррентным алгоритмам является более эффективными, чем при непосредственном вычислении или с использованием быстрых алгоритмов и показано, что более эффективными являются рекуррентные алгоритмы при $m = 1$. Легко определить, что в этом случае алгоритм вычисления ДПФ вещественной последовательности ($k = 0, \overline{N/2}$) требует $2N-6$ вещественных сложений и $3N/2-8$ вещественных умножений. Рекуррентный алгоритм вычисления ДПХ с учетом того, что $H_1(k)$ и $H_1(N-k)$ могут вычисляться совместно, требует $5N/2-7$ вещественных сложений, $3N/2-8$ вещественных умножений. В свою очередь рекуррентный алгоритм вычисления энергетического спектра вещественной последовательности по (2.27) требует $N+1$ вещественных сложений и N вещественных умножений, а по (2.29) - $3N/2$ вещественных сложений и N вещественных умножений. Арифметическая сложность рекуррентных алгоритмов для некоторых N приведена в табл.2.2.

Данные рекуррентные алгоритмы могут быть выполнены с замечением, в следствие чего они требуют N ячеек ОЗУ для хранения входных, промежуточных и выходных отсчетов. Для хранения поворачивающих множителей алгоритмы (2.23) и (2.25) требуют $N-8$ или $3N/2-12$ ячеек ПЗУ соответственно в случае использования алгоритмов комплексного умножения (1.15) или (1.14). Кроме того, выполнение алгоритмов (2.23) и (2.25) может быть организовано таким образом, что они будут требовать $N/2-4$ или $3N/4-6$ ячеек ПЗУ соответственно в случае использования алгоритмов комплексного умножения (1.15) или (1.14). Алгоритмы (2.27) и (2.29) не требуют затрат памяти для хранения поворачивающих множителей.

Сравнительный анализ рекуррентных алгоритмов вычисления ДПФ, ДПХ и энергетического спектра, а также алгоритмов БПФ, БПХ

Таблица 2.2

Арифметическая сложность рекуррентных алгоритмов

N	Алгоритм (2.23)		Алгоритм (2.25)		Алгоритм (2.27)		Алгоритм (2.29)	
	$M_R(N)$	$A_R(N)$	$M_R(N)$	$A_R(N)$	$M_R(N)$	$A_R(N)$	$M_R(N)$	$A_R(N)$
8	4	10	4	13	8	9	8	12
16	16	26	16	33	16	17	16	24
32	40	58	40	73	32	33	32	48
64	88	122	88	153	64	65	64	96
128	184	250	184	313	128	129	128	192
256	376	506	376	633	256	257	256	284
512	760	1018	760	1273	512	513	512	768
1024	1528	2042	1528	2553	1024	1025	1024	1536
2048	3064	4090	3064	5113	2048	2049	2048	3072
4096	6136	8186	6136	10233	4096	4097	4096	6144

и вычисления энергетического спектра на их основе показывает, что первые существенно превосходят последние при вычислении на скользящих (скачущих) интервалах. В частности, рекуррентные алгоритмы вычисления ДПФ и ДПХ вещественной последовательности на скользящих интервалах для $N = 1024$ превосходят алгоритмы БПФДО и БПХРО по количеству операций вещественного умножения и вещественного сложения соответственно в 2,34 и 5,15... 6,25 раза, а по объему требуемой памяти имеют одинаковые показатели.

Анализ рекуррентных алгоритмов вычисления ДПХ и энергетического спектра на их основе показывает, что они имеют одинаковые показатели по сравнению с рекуррентными алгоритмами вычисления ДПФ и энергетического спектра на их основе, незначительно уступая по количеству операций вещественного сложения, что для $N = 1024$ составляет проигрыш соответственно в 1,25 и 1,5 раза.

В Ы В О Д Ы

1. Разработаны эффективные алгоритмы БПХ по основаниям 4,8 и с расщепленным основанием с прореживанием по времени и частоте. Данные алгоритмы имеют простую и регулярную структуру, выполняются с замещением и позволяют использовать эффективные вычислительные процедуры, сводящие к минимуму вычислительные затраты, в результате чего они являются максимально пригодными для микропроцессорной реализации.

2. Проведен сравнительный анализ алгоритмов БПХ по количеству арифметических операций и объему требуемой памяти. Показано, что наилучшими среди них являются алгоритмы БПХРО, которые по количеству операций вещественного умножения и вещественного сложения обеспечивают выигрыш соответственно в 1,43 и 1,17 раза по сравнению с алгоритмами БПХ-2 для $N = 1024$. Алгоритмы БПХ превосходят алгоритмы БПФ комплексной последовательности, вещест-

венной последовательности второго класса и имеют практически одинаковые характеристики по сравнению с алгоритмами БПФ вещественной последовательности первого класса, что делает их более предпочтительными в применении благодаря регулярности и простоте структуры.

3. Разработаны рекуррентные алгоритмы вычисления ДПХ и энергетического спектра на скользящих (скачущих) интервалах. Проведен анализ данных алгоритмов по количеству арифметических операций и объему требуемой памяти. Показано, что данные алгоритмы превосходят алгоритмы БПХ и вычисления энергетического спектра на их основе, обеспечивая максимальный выигрыш при вычислении на скользящих интервалах, который для $N = 1024$ составляет 2,34 и 5,15 раза соответственно по количеству операций вещественного умножения и вещественного сложения.

ГЛАВА 3. ИССЛЕДОВАНИЕ ТОЧНОСТИ АЛГОРИТМОВ БЫСТРОГО ПРЕОБРАЗОВАНИЯ ФУРЬЕ И ХАРТЛИ С ФИКСИРОВАННОЙ ЗАПЯТОЙ

3.1. Основные положения исследования точности алгоритмов быстрого преобразования Фурье

В основу исследования точности алгоритмов БПФ положен подход, предложенный в [113].

В случае входного масштабирования ошибка $\varepsilon(k)$, $k = 0, \overline{N-1}$, в произвольной точке на выходе алгоритма БПФ определяется как некоторая сумма произведений ошибок умножения ε_y^k на степени комплексных коэффициентов W_N , т.е.

$$\varepsilon(k) = \sum_{\rho} \varepsilon_y^k(\rho) W_N^{a(\rho)}, \quad (3.1)$$

где $a(\rho)$ – показатель, зависящий от места возникновения ошибки умножения.

Так как

$$|\varepsilon(k)| \leq \sum_{\rho} |\varepsilon_y^k(\rho) W_N^{a(\rho)}| = \sum_{\rho} |\varepsilon_y^k(\rho)|,$$

то СКЗ выходной ошибки

$$M(|\varepsilon(k)|^2) \approx \sum_{\rho} M(|\varepsilon_y^k(\rho)|^2). \quad (3.2)$$

Таким образом, при входном масштабировании СКЗ ошибки в произвольной точке на выходе алгоритма БПФ равно сумме СКЗ ошибок умножения, распространяющихся к данной выходной точке.

В случае автоматического масштабирования ошибка в произвольной точке на выходе алгоритма БПФ определяется как сумма ошибок, обусловленных ошибками умножения и ошибками сдвига. При этом

ошибка, обусловленная ошибками умножения, определяется как

$$\varepsilon_y(k) = \sum_p \varepsilon_y^x(p) W_N^{a(p)} \cdot (1/2)^{b(p)}, \quad (3.3)$$

где $b(p)$ – показатель, зависящий от места возникновения ошибки умножения, 2 – основание алгоритма БПФ. Множитель $(1/2)^{b(p)}$ в выражении (3.3) возникает в связи с тем, что на каждом этапе алгоритма БПФ осуществляется сдвиг чисел на $\log_2 2$ разрядов вправо, что равносильно умножению на $1/2$.

Ошибка в выходной точке алгоритма БПФ, обусловленная ошибками сдвига, определяется аналогично (3.3), а именно,

$$\varepsilon_c(k) = \sum_s \varepsilon_c^x(s) W_N^{c(s)} \cdot (1/2)^{d(s)}, \quad (3.4)$$

где $c(s)$ и $d(s)$ – показатели, зависящие от места возникновения ошибки сдвига.

Таким образом, ошибка в выходной точке алгоритма БПФ будет

$$\begin{aligned} \varepsilon(k) = \varepsilon_y(k) + \varepsilon_c(k) = & \sum_p \varepsilon_y^x(p) W_N^{a(p)} \cdot (1/2)^{b(p)} + \\ & + \sum_s \varepsilon_c^x(s) W_N^{c(s)} \cdot (1/2)^{d(s)}. \end{aligned} \quad (3.5)$$

Тогда

$$\begin{aligned} M(|\varepsilon(k)|^2) \approx & \sum_p M(|\varepsilon_y^x(p)|^2) \cdot (1/2)^{2b(p)} + \\ & + \sum_s M(|\varepsilon_c^x(s)|^2) \cdot (1/2)^{2d(s)}. \end{aligned} \quad (3.6)$$

Таким образом, СКЗ ошибки в выходной точке алгоритма БПФ при

автоматическом масштабировании равно сумме СКЗ ошибок умножения и сдвига, распространяющихся к данной выходной точке, ослабленных соответственно в $(1/2)^{2d(p)}$ и $(1/e)^{2d(s)}$ раза.

Для дальнейшего изложения нам понадобятся следующие утверждения, в справедливости которых нетрудно убедиться.

Утверждение 1. Ошибка, возникающая на l -ой ступени алгоритма $N = e^m$ - точечного алгоритма БПФ, распространяется к e^{m-l} выходным точкам ($l = 0, m-1$).

Утверждение 2. Ошибки сдвига в алгоритмах БПФ с прореживанием по времени и по частоте, а также ошибки умножения в алгоритмах БПФ с прореживанием по времени, возникающие на l -ой ступени, при автоматическом масштабировании ослабляются в $(1/e)^{m-l-1}$ раз.

Утверждение 3. Ошибки умножения в алгоритмах БПФ с прореживанием по частоте, возникающие на l -ой ступени, при автоматическом масштабировании ослабляются в $(1/e)^{m-l}$ раз.

В заключение определим СКЗ сигнала на выходе алгоритма БПФ.

При входном масштабировании сигнал на входе алгоритма БПФ такой, что $|x(n)| < 1/N$ для всех n . При равномерном законе распределения СКЗ входного сигнала равно $1/(3N^2)$. По теореме Парсеваля [15] СКЗ сигнала на выходе алгоритма ДПФ увеличивается в N раз, т.е. равно $1/(3N)$.

При автоматическом масштабировании сигнал на входе алгоритма БПФ такой, что $|x(n)| < 1$ для всех n . При равномерном законе распределения СКЗ входного сигнала равно $1/3$. СКЗ выходного сигнала по теореме Парсеваля должно быть равно $N/3$. Однако, с учётом того, что при автоматическом масштабировании на выходе алгоритма БПФ получаются N -ые части выходных гармоник, СКЗ выходного сигнала равно $1/(3N)$.

Таким образом, как при входном так и автоматическом масшта-

бирования, СКЗ выходного сигнала равно $1/(3N)$.

3.2. Исследование точности алгоритмов быстрого преобразования Фурье

Исследование точности алгоритмов БПФ по основанию 2 проведём с использованием граф-схем, представленных на рис.31 и 32 соответственно для алгоритма с прореживанием по времени и частоте, где приняты следующие обозначения: \otimes - ошибка умножения, \circ - ошибка сдвига.

Как видно из рис.3.1, точки на выходе алгоритма БПФ с прореживанием по времени, объединенные в группы (очерчено сплошными линиями), имеют одинаковые СКЗ ошибки, так как к каждой из точек данной группы распространяется одинаковое количество ошибок умножения и сдвига. В общем случае количество таких групп равно $m-1$, а количество точек в группах (сверху вниз) равно 4, 4, 8, 16, ..., $N/2$.

Рассмотрим сначала случай входного масштабирования, когда ошибки на выходе алгоритма БПФ обусловлены лишь ошибками умножения.

Заметим, что ошибки умножения для нижней группы точек начинают накапливаться со второго этапа алгоритма БПФ, для группы точек над ней - с третьего и т.д. вплоть до самой верхней группы точек, совсем не содержащей ошибок. При этом количество ошибок умножения, распространяющихся к каждой выходной точке внутри группы, пользуясь утверждением I, можно определить следующим образом:

$$\begin{aligned}
 & - \text{для нижней группы точек} - \frac{1}{N/2} \sum_{l=2}^{m-1} (N/2 - N/4) \cdot 2^{m-l}; \\
 & - \text{для группы точек над ней} - \frac{1}{N/4} \sum_{l=3}^{m-1} (N/4 - N/8) \cdot 2^{m-l};
 \end{aligned}$$

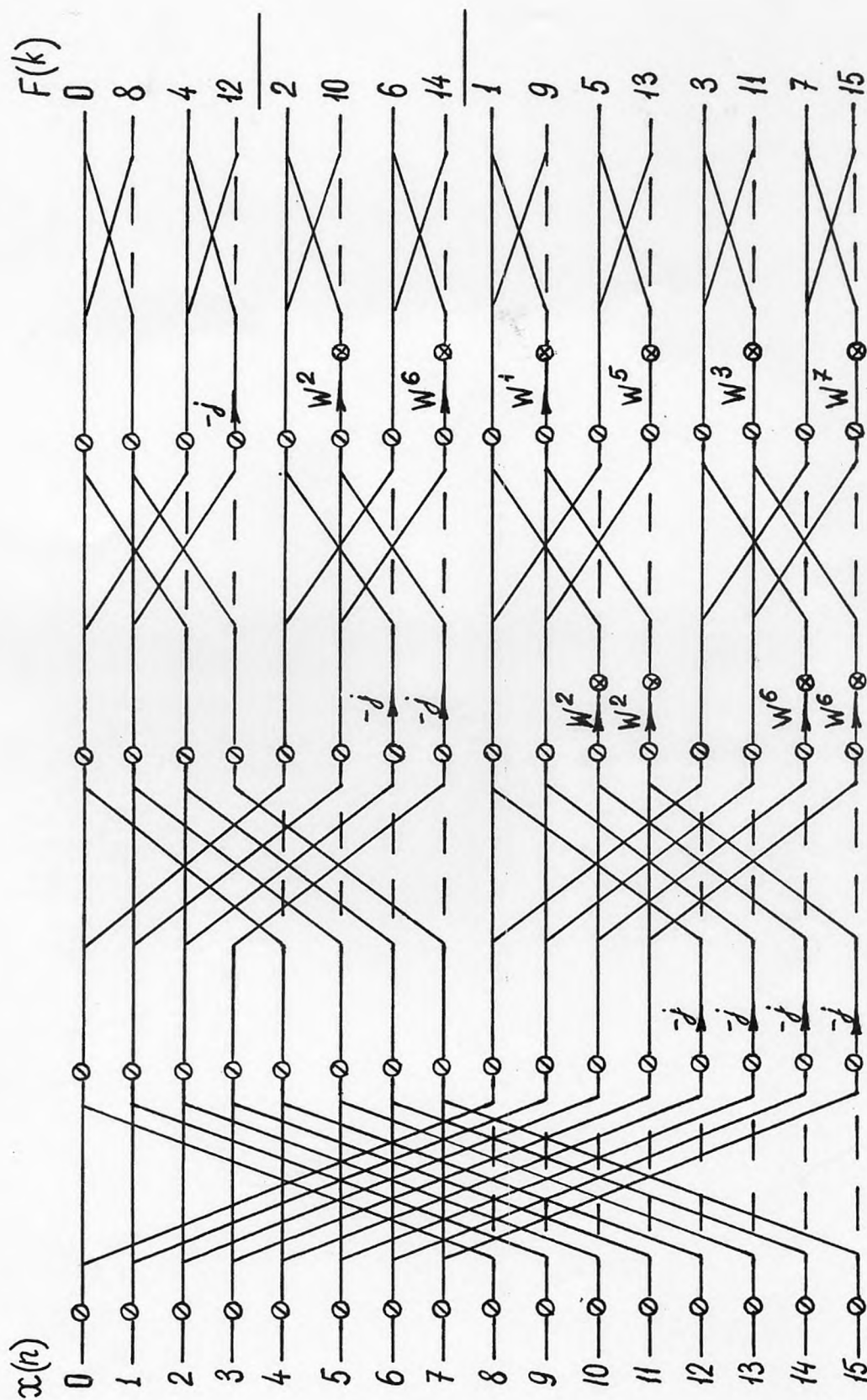


Рис.3.1. Граф-схема алгоритма БПФ по основанию 2 с прореживанием по времени с источниками вычислительных ошибок

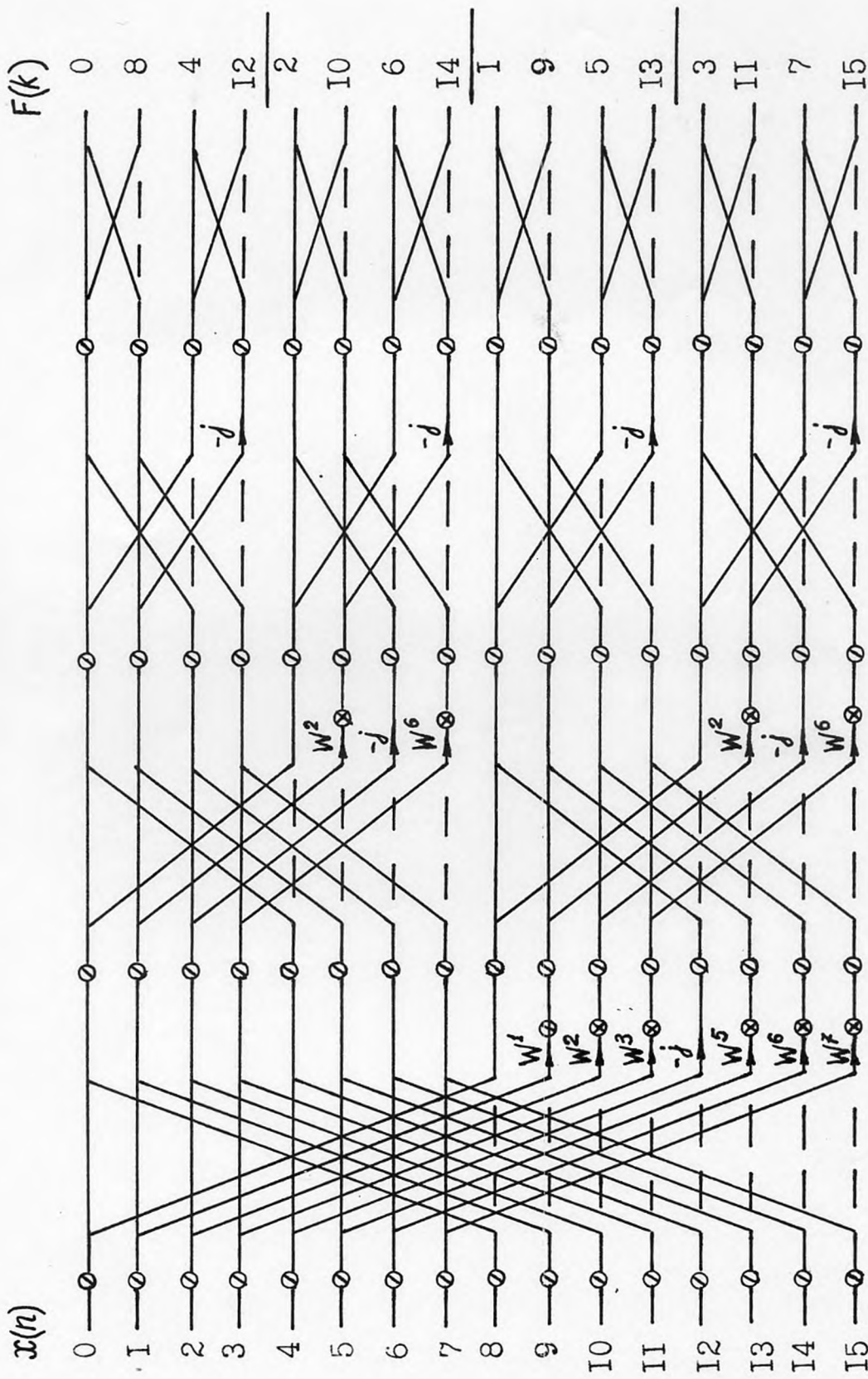


Рис.3.2. Граф-схема алгоритма БПФ по основанию 2 с прореживанием по частоте с источниками вычислительных ошибок

и т.д. и в общем случае определяется как

$$\frac{1}{2} \sum_{l=s(k)}^{m-1} 2^{m-l} = \frac{N}{2^{s(k)}} - 1,$$

где $s(k)$ с учётом того, что точки на выходе данного алгоритма БПФ расположены в двоично-инверсном порядке определяется как

$$s(k) = 2 + \min_i \{i : k_i = 1\}, \quad (3.7)$$

где $k = \sum_{i=0}^{m-1} k_i \cdot 2^i$.

Тогда с учётом (3.2) СКЗ индивидуальных ошибок на выходе алгоритма БПФ определяется как

$$M(|\varepsilon(k)|^2) = \begin{cases} 0 & , k = 0, N/4, N/2, 3N/4 \\ \left(\frac{N}{2^{s(k)}} - 1\right) \cdot M(|\varepsilon_y^k|^2) & \text{для других } k. \end{cases} \quad (3.8)$$

Среднее суммарное СКЗ выходной ошибки пользуясь утверждением I, и с учётом (3.2) определяется как

$$\begin{aligned} \frac{1}{N} \sum_{k=0}^{N-1} M(|\varepsilon(k)|^2) &= M(|\varepsilon_y^k|^2) \cdot \frac{1}{N} \sum_{l=2}^{m-1} (N/2 - N/2^l) \cdot 2^{m-l} = \\ &= M(|\varepsilon_y^k|^2) \cdot (N/6 - 1 + 4/(3N)) \approx \frac{N}{6} M(|\varepsilon_y^k|^2). \end{aligned} \quad (3.9)$$

При автоматическом масштабировании СКЗ индивидуальной ошибки и среднее суммарное СКЗ выходной ошибки, обусловленной ошибками умножения, можно получить, используя утверждение 2 и результаты, полученные выше, с учётом (3.6). Тогда

$$M_y(|\varepsilon(k)|^2) = \begin{cases} 0, & k = 0, N/4, N/2, 3N/4 \\ M(|\varepsilon_y^k|^2) \frac{1}{2} \sum_{l=S(k)}^{m-1} 2^{m-l} \cdot (1/2)^{2(m-l-1)} = \\ = (2 - \frac{2^{S(k)+1}}{N}) M(|\varepsilon_y^k|^2) & \text{для других } k \end{cases} \quad (3.10)$$

и

$$\begin{aligned} \frac{1}{N} \sum_{k=0}^{N-1} M_y(|\varepsilon(k)|^2) &= M(|\varepsilon_y^k|^2) \cdot \frac{1}{N} \sum_{l=2}^{m-1} (N/2 - N/2^l) \cdot 2^{m-l} \times \\ &\times (1/2)^{2(m-l-1)} = (2 - \frac{4m}{N}) M(|\varepsilon_y^k|^2) \approx 2 M(|\varepsilon_y^k|^2). \end{aligned} \quad (3.11)$$

СКЗ индивидуальной ошибки и среднее суммарное СКЗ выходной ошибки, обусловленной ошибками сдвига, равны между собой и, пользуясь утверждениями 1 и 2, с учётом (3.6) их можно определить как

$$\begin{aligned} M_c(|\varepsilon(k)|^2) &= \frac{1}{N} \sum_{k=0}^{N-1} M_c(|\varepsilon(k)|^2) = M(|\varepsilon_c^k|^2) \cdot \frac{1}{N} \sum_{l=0}^{m-1} N \cdot 2^{m-l} \times \\ &\times (1/2)^{2(m-l-1)} = (4 - \frac{4}{N}) M(|\varepsilon_c^k|^2) \approx 4 M(|\varepsilon_c^k|^2). \end{aligned} \quad (3.12)$$

Таким образом, СКЗ индивидуальной ошибки и среднее суммарное СКЗ выходной ошибки определяются соответственно как

$$M(|\varepsilon(k)|^2) = \begin{cases} (4 - \frac{4}{N}) M(|\varepsilon_c^k|^2), & k = 0, N/4, N/2, 3N/4 \\ (2 - \frac{2^{S(k)+1}}{N}) M(|\varepsilon_y^k|^2) + \\ + (4 - \frac{4}{N}) M(|\varepsilon_c^k|^2) & \text{для других } k \end{cases} \quad (3.13)$$

и

$$\frac{1}{N} \sum_{k=0}^{N-1} M(|\varepsilon(k)|^2) = \left(2 - \frac{4m}{N}\right) M(|\varepsilon_y^k|^2) + \left(4 - \frac{4}{N}\right) M(|\varepsilon_c^k|^2) \approx$$

$$\approx 2M(|\varepsilon_y^k|^2) + 4M(|\varepsilon_c^k|^2). \quad (3.14)$$

Как видно из рис.3.2, точки на выходе алгоритма БПФ с прореживанием по частоте, объединенные в группы, имеют одинаковые СКЗ ошибки. Количество точек в каждой группе равно четырем, а количество таких групп, очевидно, равно $N/4$.

Рассмотрим сначала случай входного масштабирования.

Учитывая, что на каждом из этапов алгоритма БПФ, за исключением двух последних, может возникнуть $N/2^l - 2$ ошибок умножения, распространяющихся к каждой точке группы, и, учитывая, что точки на выходе алгоритма БПФ расположены в двоично-инверсном порядке, количество ошибок умножения, распространяющихся к каждой точке группы, определяется как

$$\sum_{l=1}^{m-2} k_l \cdot (N/2^l - 2),$$

где $k = \sum_{l=1}^m k_l \cdot 2^{l-1}$.

Тогда СКЗ индивидуальной ошибки с учётом (3.2) определяется как

$$M(|\varepsilon(k)|^2) = \begin{cases} 0 & , k = 0, N/4, N/2, 3N/4 \\ M(|\varepsilon_y^k|^2) \sum_{l=1}^{m-2} k_l \cdot (N/2^l - 2) & \text{для других } k. \end{cases} \quad (3.15)$$

Среднее суммарное СКЗ выходной ошибки на основании утверждения I и с учётом (3.2) определяется как

$$\begin{aligned} \frac{1}{N} \sum_{k=0}^{N-1} M(|\varepsilon(k)|^2) &= M(|\varepsilon_y^k|^2) \cdot \frac{1}{N} \sum_{l=1}^{m-2} (N/2 - 2^l) \cdot 2^{m-l} = \\ &= (N/2 - m) \cdot M(|\varepsilon_y^k|^2) \approx \frac{N}{2} M(|\varepsilon_y^k|^2). \end{aligned} \quad (3.16)$$

При автоматическом масштабировании СКЗ индивидуальной ошибки и среднее суммарное СКЗ выходной ошибки, обусловленной ошибками умножения, можно получить, используя утверждение 3 и результаты, полученные выше, с учётом (3.6). Тогда

$$M_y(|\varepsilon(k)|^2) = \begin{cases} 0 & , k=0, N/4, N/2, 3N/4 \\ M(|\varepsilon_y^k|^2) \sum_{l=1}^{m-2} k_l \cdot (N/2^l - 2) \cdot (1/2)^{2(m-l)} & \text{для других } k \end{cases} \quad (3.17)$$

и

$$\begin{aligned} \frac{1}{N} \sum_{k=0}^{N-1} M_y(|\varepsilon(k)|^2) &= M(|\varepsilon_y^k|^2) \cdot \frac{1}{N} \sum_{l=1}^{m-2} (N/2 - 2^l) \cdot 2^{m-l} \cdot (1/2)^{2(m-l)} = \\ &= \left(\frac{1}{6} - \frac{1}{N} + \frac{4}{3N^2} \right) M(|\varepsilon_y^k|^2) \approx \frac{1}{6} M(|\varepsilon_y^k|^2). \end{aligned} \quad (3.18)$$

СКЗ индивидуальной ошибки и среднее суммарное СКЗ выходной ошибки, обусловленной ошибками сдвига, очевидно, такие же, как и в алгоритме БПФ с прореживанием по времени. Поэтому СКЗ индивидуальной ошибки и среднее суммарное СКЗ выходной ошибки определяется как

$$M(|\varepsilon(k)|^2) = \begin{cases} (4 - \frac{4}{N}) M(|\varepsilon_c^k|^2), & k=0, N/4, N/2, 3N/4 \\ M(|\varepsilon_y^k|^2) \sum_{l=1}^{m-2} k_l \cdot (N/2^l - 2) \cdot (1/2)^{2(m-l)} + \\ + M(|\varepsilon_c^k|^2) \cdot (4 - \frac{4}{N}) & \text{для других } k \end{cases} \quad (3.19)$$

и

$$\frac{1}{N} \sum_{k=0}^{N-1} M(|\varepsilon(k)|^2) = \left(\frac{1}{6} - \frac{1}{N} + \frac{4}{3N^2} \right) M(|\varepsilon_y^k|^2) + \left(4 - \frac{4}{N} \right) M(|\varepsilon_c^k|^2) \approx$$

$$\approx \frac{1}{6} M(|\varepsilon_y^k|^2) + 4 M(|\varepsilon_c^k|^2). \quad (3.20)$$

Аналогичным образом может быть проведен анализ вычислительных ошибок алгоритмов БПФ по основаниям 4, 8 и с расцепленным основанием. Представим окончательные результаты данного анализа.

Для алгоритма БПФ по основанию 4 с прореживанием по времени в случае входного масштабирования СКЗ индивидуальной ошибки и среднее суммарное СКЗ выходной ошибки соответственно определяются как

$$M(|\varepsilon(k)|^2) = \begin{cases} 0 & , k = 0, N/4, N/2, 3N/4 \\ \left(\mu \frac{N}{2 \cdot 4^{s(k)}} + \frac{N}{4^{s(k)} + \mu} - 1 \right) M(|\varepsilon_y^k|^2) & \text{для других } k, \end{cases} \quad (3.21)$$

где $s(k) = 1 + \min_i \{i : k_i \neq 0\}$, $k = \sum_{i=0}^{m-1} k_i \cdot 4^i$, $k_i = \overline{0, 3}$; $\mu = 1$,

если один из $k_i = 2$ для $i = \overline{0, m-2}$

и

$$\frac{1}{N} \sum_{k=0}^{N-1} M(|\varepsilon(k)|^2) = \left(\frac{11N}{60} - 1 + \frac{16}{15N} \right) M(|\varepsilon_y^k|^2) \approx \frac{11N}{60} M(|\varepsilon_y^k|^2). \quad (3.22)$$

В случае автоматического масштабирования СКЗ индивидуальной ошибки и среднее суммарное СКЗ выходной ошибки соответственно

определяются как

$$M(|\varepsilon(k)|^2) = \begin{cases} \left(\frac{16}{3} - \frac{16}{3N}\right) M(|\varepsilon_c^k|^2) & , k=0, N/4, N/2, 3N/4 \\ \left(\mu \frac{4^{S(k)+2}}{2N} + 4 - \frac{4^{S(k)+\mu+1}}{N}\right) M(|\varepsilon_y^k|^2) + \\ + \left(\frac{16}{3} - \frac{16}{3N}\right) M(|\varepsilon_c^k|^2) & \text{для других } k \end{cases} \quad (3.23)$$

и

$$\begin{aligned} \frac{1}{N} \sum_{k=0}^{N-1} M(|\varepsilon(k)|^2) &= \left(4 - \frac{16m}{N}\right) M(|\varepsilon_y^k|^2) + \left(\frac{16}{3} - \frac{16}{3N}\right) M(|\varepsilon_c^k|^2) \approx \\ &\approx 4M(|\varepsilon_y^k|^2) + \frac{16}{3} M(|\varepsilon_c^k|^2). \end{aligned} \quad (3.24)$$

Для алгоритма БПФ по основанию 4 с прореживанием по частоте в случае входного масштабирования СКЗ индивидуальной ошибки и среднее суммарное СКЗ выходной ошибки соответственно определяются как

$$M(|\varepsilon(k)|^2) = \begin{cases} 0 & , k=0, N/4, N/2, 3N/4 \\ M(|\varepsilon_y^k|^2) \sum_{l=0}^{m-2} \lambda \cdot \left(\frac{N}{4^{l+1}} - 1 - \mu\right) & \text{для других } k, \end{cases} \quad (3.25)$$

где $\lambda=1$, если $k_l \neq 0$; $k = \sum_{l=0}^{m-1} k_l \cdot 4^l$, $k_l = \overline{0,3}$; $\mu=1$, если $k_l=2$

и

$$\frac{1}{N} \sum_{k=0}^{N-1} M(|\varepsilon(k)|^2) = \left(\frac{N}{4} - m\right) M(|\varepsilon_y^k|^2) \approx \frac{N}{4} M(|\varepsilon_y^k|^2). \quad (3.26)$$

В случае автоматического масштабирования СКЗ индивидуальной ошибки и среднее суммарное СКЗ выходной ошибки соответственно определяются как

$$M(|\varepsilon(k)|^2) = \begin{cases} (\frac{16}{3} - \frac{16}{3N}) M(|\varepsilon_c^k|^2), & k = 0, N/4, N/2, 3N/4 \\ M(|\varepsilon_y^k|^2) \sum_{l=0}^{m-2} \lambda \cdot (\frac{N}{4^{l+1}} - 1 - \mu) \cdot (1/4)^{2(m-l-1)} + \\ + (\frac{16}{3} - \frac{16}{3N}) M(|\varepsilon_c^k|^2) & \text{для других } k \end{cases} \quad (3.27)$$

и

$$\begin{aligned} \frac{1}{N} \sum_{k=0}^{N-1} M(|\varepsilon(k)|^2) &= (\frac{11}{60} - \frac{1}{N} + \frac{16}{15N^2}) M(|\varepsilon_y^k|^2) + (\frac{16}{3} - \frac{16}{3N}) M(|\varepsilon_c^k|^2) \approx \\ &\approx \frac{11}{60} M(|\varepsilon_y^k|^2) + \frac{16}{3} M(|\varepsilon_c^k|^2). \end{aligned} \quad (3.28)$$

Для алгоритмов БПФ по основанию 8 при анализе вычислительных ошибок следует учитывать, что вычислительные ошибки возникают как снаружи так и внутри основных вычислительных операций "бабочек", представляющих собой восьми-точечное ДПФ. При этом последние вычисляются с использованием алгоритма БПФ по основанию 2. Для обеспечения минимальных ошибок в случае входного масштабирования используется алгоритм с прореживанием по времени, а в случае автоматического масштабирования - алгоритм с прореживанием по частоте. Тогда для алгоритма БПФ по основанию 8 с прореживанием по времени в случае входного масштабирования СКЗ индивидуальной ошибки и среднее суммарное СКЗ выходной ошибки определяются соответственно как

$$M(|\varepsilon(k)|^2) = \begin{cases} 0 & , \quad k=0, N/4, N/2, 3N/4 \\ (\mu \frac{3N}{4 \cdot 8^{s(k)}} + \frac{N}{8^{s(k)+\mu}} + \sum_{i=0}^{m-1} \sigma \frac{N}{8^{i+1}}) M(|\varepsilon_y^k|^2) \end{cases} \quad (3.29)$$

для других k ,

где $s(k) = 1 + \min \{i : k_i \neq 0\}$,

$$k = \sum_{i=0}^{m-1} k_i \cdot 8^i, \quad k_i = \overline{0, 7}; \quad \mu = 1,$$

если один из $k_i = 4, i = \overline{0, m-2}$; $\sigma = 1$, если k_i - нечётное

и

$$\begin{aligned} \frac{1}{N} \sum_{k=0}^{N-1} M(|\varepsilon(k)|^2) &= \left[\left(\frac{55N}{504} - 1 + \frac{64}{63N} \right) + \left(\frac{N}{14} - \frac{1}{14} \right) \right] M(|\varepsilon_y^k|^2) \approx \\ &\approx \frac{13N}{72} M(|\varepsilon_y^k|^2). \end{aligned} \quad (3.30)$$

В случае автоматического масштабирования СКЗ индивидуальной ошибки и среднее суммарное СКЗ выходной ошибки соответственно определяются как

$$M(|\varepsilon(k)|^2) = \begin{cases} \left(\frac{64}{7} - \frac{64}{7N} \right) M(|\varepsilon_c^k|^2), & k=0, N/4, N/2, 3N/4 \\ \left(\mu \frac{3 \cdot 8^{s(k)+2}}{4N} + 8 - \frac{8^{s(k)+\mu+1}}{N} + \frac{1}{N} \sum_{i=0}^{m-1} \sigma \cdot 8^{i+1} \right) \times \\ \times M(|\varepsilon_y^k|^2) + \left(\frac{64}{7} - \frac{64}{7N} \right) M(|\varepsilon_c^k|^2) \end{cases} \quad (3.31)$$

для других k

и

$$\frac{1}{N} \sum_{k=0}^{N-1} M(|\varepsilon(k)|^2) = \left[\left(8 - \frac{64m}{N} \right) + \left(\frac{4}{7} - \frac{4}{7N} \right) \right] M(|\varepsilon_y^k|^2) +$$

$$+ \left(\frac{64}{7} - \frac{64}{7N} \right) M(|\varepsilon_c^k|^2) \approx \frac{60}{7} M(|\varepsilon_y^k|^2) + \frac{64}{7} M(|\varepsilon_c^k|^2). \quad (3.32)$$

Для алгоритма БПФ по основанию 8 с прореживанием по частоте в случае входного масштабирования СКЗ индивидуальной ошибки и среднее суммарное СКЗ выходной ошибки соответственно определяются как

$$M(|\varepsilon(k)|^2) = \begin{cases} 0 & , k = 0, N/4, N/2, 3N/4. \\ M(|\varepsilon_y^k|^2) \cdot \left[\sum_{l=0}^{m-2} \lambda \cdot \left(\frac{N}{8^{l+1}} - 1 - \mu \right) + \right. \\ \left. + \sum_{l=0}^{m-1} \delta \frac{N}{8^{l+1}} \right] & \text{для других } k, \end{cases} \quad (3.33)$$

где $\lambda = 1$, если $k_l \neq 0$; $k = \sum_{l=0}^{m-1} k_l \cdot 8^l$, $k_l = \overline{0, 7}$; $\mu = 1$, если

$k_l = 4$ и

$$\frac{1}{N} \sum_{k=0}^{N-1} M(|\varepsilon(k)|^2) = \left[\left(\frac{N}{8} - m \right) + \left(\frac{N}{14} - \frac{1}{14} \right) \right] M(|\varepsilon_y^k|^2) \approx$$

$$\approx \frac{11N}{56} M(|\varepsilon_y^k|^2). \quad (3.34)$$

В случае автоматического масштабирования СКЗ индивидуальной ошибки и среднее суммарное СКЗ выходной ошибки соответственно определяются как

$$M(|\varepsilon(k)|^2) = \begin{cases} \left(\frac{64}{7} - \frac{64}{7N}\right) M(|\varepsilon_c^k|^2) & , k = 0, N/4, N/2, 3N/4 \\ \left[\sum_{l=0}^{m-2} 2 \cdot \left(\frac{N}{8^{l+1}} - 1 - \mu\right) \cdot (1/8)^{2(m-l)} + \frac{1}{N} \sum_{l=0}^{m-1} 5 \cdot 8^{l+1} \right] \times \\ \times M(|\varepsilon_y^k|^2) + \left(\frac{64}{7} - \frac{64}{7N}\right) M(|\varepsilon_c^k|^2) & \text{для других } k \end{cases} \quad (3.35)$$

и

$$\begin{aligned} \frac{1}{N} \sum_{k=0}^{N-1} M(|\varepsilon(k)|^2) &= \left[\left(\frac{55}{504} - \frac{1}{N} + \frac{64}{63N}\right) + \left(\frac{4}{7} - \frac{4}{7N}\right) \right] M(|\varepsilon_y^k|^2) + \\ &+ \left(\frac{64}{7} - \frac{64}{7N}\right) M(|\varepsilon_c^k|^2) \approx \frac{49}{72} M(|\varepsilon_y^k|^2) + \frac{64}{7} M(|\varepsilon_c^k|^2). \end{aligned} \quad (3.36)$$

Для алгоритма БПФ с расщепленным основанием с прореживанием по времени в случае входного масштабирования СКЗ индивидуальной ошибки и среднее суммарное СКЗ выходной ошибки соответственно определяются как

$$M(|\varepsilon(k)|^2) = \begin{cases} 0 & , k = 0, N/4, N/2, 3N/4 \\ \left(\frac{2N}{3 \cdot 2^{s(k)}} - 1 + \frac{(-1)^{m-s(k)}}{3}\right) M(|\varepsilon_y^k|^2) & \text{для других } k, \end{cases} \quad (3.37)$$

где $s(k) = 1 + \min_i \{i : k_i = 1\}$, $k = \sum_{i=0}^{m-1} k_i \cdot 2^i$

и

$$\frac{1}{N} \sum_{k=0}^{N-1} M(|\varepsilon(k)|^2) = \frac{1}{3} \cdot \left(\frac{2N}{3} - 3 + \frac{1}{3} (-1)^{m-1} + \frac{8}{3N}\right) M(|\varepsilon_y^k|^2) \approx \frac{2N}{9} M(|\varepsilon_y^k|^2) \quad (3.38)$$

В случае автоматического масштабирования СКЗ индивидуальной ошибки и среднее суммарное СКЗ выходной ошибки соответственно определяются как

$$M(|\varepsilon(k)|^2) = \begin{cases} (4 - \frac{4}{N}) M(|\varepsilon_c^k|^2), & k = 0, N/4, N/2, 3N/4 \\ (\frac{4}{5} - \frac{4 \cdot 2^{s(k)}}{3N} + \frac{8 \cdot (-1)^{m-s(k)} \cdot 4^{s(k)}}{15N^2}) M(|\varepsilon_y^k|^2) + \\ + (4 - \frac{4}{N}) M(|\varepsilon_c^k|^2) & \text{для других } k \end{cases} \quad (3.39)$$

и

$$\begin{aligned} \frac{1}{N} \sum_{k=0}^{N-1} M(|\varepsilon(k)|^2) &= (\frac{4}{5} - \frac{4m}{3N} - \frac{4}{9N} + \frac{16 \cdot (-1)^{m-1}}{45N^2}) \cdot M(|\varepsilon_y^k|^2) + \\ &+ (4 - \frac{4}{N}) M(|\varepsilon_c^k|^2) \approx \frac{4}{5} M(|\varepsilon_y^k|^2) + 4M(|\varepsilon_c^k|^2). \end{aligned} \quad (3.40)$$

Для алгоритма БПФ с расщеплённым основанием с прореживанием по частоте в случае входного масштабирования СКЗ индивидуальной ошибки и среднее суммарное СКЗ выходной ошибки соответственно определяются как

$$M(|\varepsilon(k)|^2) = \begin{cases} 0 & , k = 0, N/4, N/2, 3N/4 \\ M(|\varepsilon_y^k|^2) \sum_{l=s(k)}^{m-1} k_{l-2} \cdot (\frac{N}{2^l} - 1) & \text{для других } k, \end{cases} \quad (3.41)$$

где $s(k) = 2 + \min_i \{i : k_i = 1\}$, $k = \sum_{i=0}^{m-1} k_i \cdot 2^i$; l принимает только

четные или нечетные значения в зависимости от того, является ли $s(k)$ четным или нечетным соответственно,

$$\text{и} \quad \frac{1}{N} \sum_{k=0}^{N-1} M(|\varepsilon(k)|^2) = \left(\frac{N}{5} - \frac{m}{3} - \frac{1}{9} - \frac{4 \cdot (-1)^m}{45N} \right) M(|\varepsilon_y^k|^2) \approx \frac{N}{5} M(|\varepsilon_y^k|^2). \quad (3.42)$$

В случае автоматического масштабирования СКЗ индивидуальной ошибки и среднее суммарное СКЗ выходной ошибки соответственно определяются как

$$M(|\varepsilon(k)|^2) = \begin{cases} \left(4 - \frac{4}{N}\right) M(|\varepsilon_c^k|^2) & , \quad k=0, N/4, N/2, 3N/4 \\ M(|\varepsilon_y^k|^2) \sum_{l=s(k)}^{m-1} k_{l-1} \cdot \frac{(-1)^l}{2^l} \cdot \left(\frac{N}{2} - 1\right) \cdot (1/2)^{2(m-l)} + \\ + \left(4 - \frac{4}{N}\right) M(|\varepsilon_c^k|^2) & \text{для других } k \end{cases} \quad (3.43)$$

$$\text{и} \quad \frac{1}{N} \sum_{k=0}^{N-1} M(|\varepsilon(k)|^2) = \left(\frac{2}{9} - \frac{1}{N} - \frac{(-1)^m}{N} + \frac{8}{9N^2} \right) \cdot M(|\varepsilon_y^k|^2) + \\ + \left(4 - \frac{4}{N}\right) \cdot M(|\varepsilon_c^k|^2) \approx \frac{2}{9} M(|\varepsilon_y^k|^2) + 4 M(|\varepsilon_c^k|^2). \quad (3.44)$$

3.3. Исследование точности алгоритмов быстрого преобразования Хартли

Исследование точности алгоритмов БПХ по основанию 2 проведем на основании граф-схем, представленных на рис.3.3 и 3.4 соответственно для алгоритма с прореживанием по времени и частоте.

Сравнительный анализ данных граф-схем с граф-схемами, представленным на рис.3.1 и 3.2 для алгоритмов БПФ, показывает, что они отличаются лишь этапами, на которых выполняются умножения на поворачивающие множители. Так как $|W_N^k| = 1$, то СКЗ ошибки умножения или сдвига в алгоритмах БПФ после умножения на

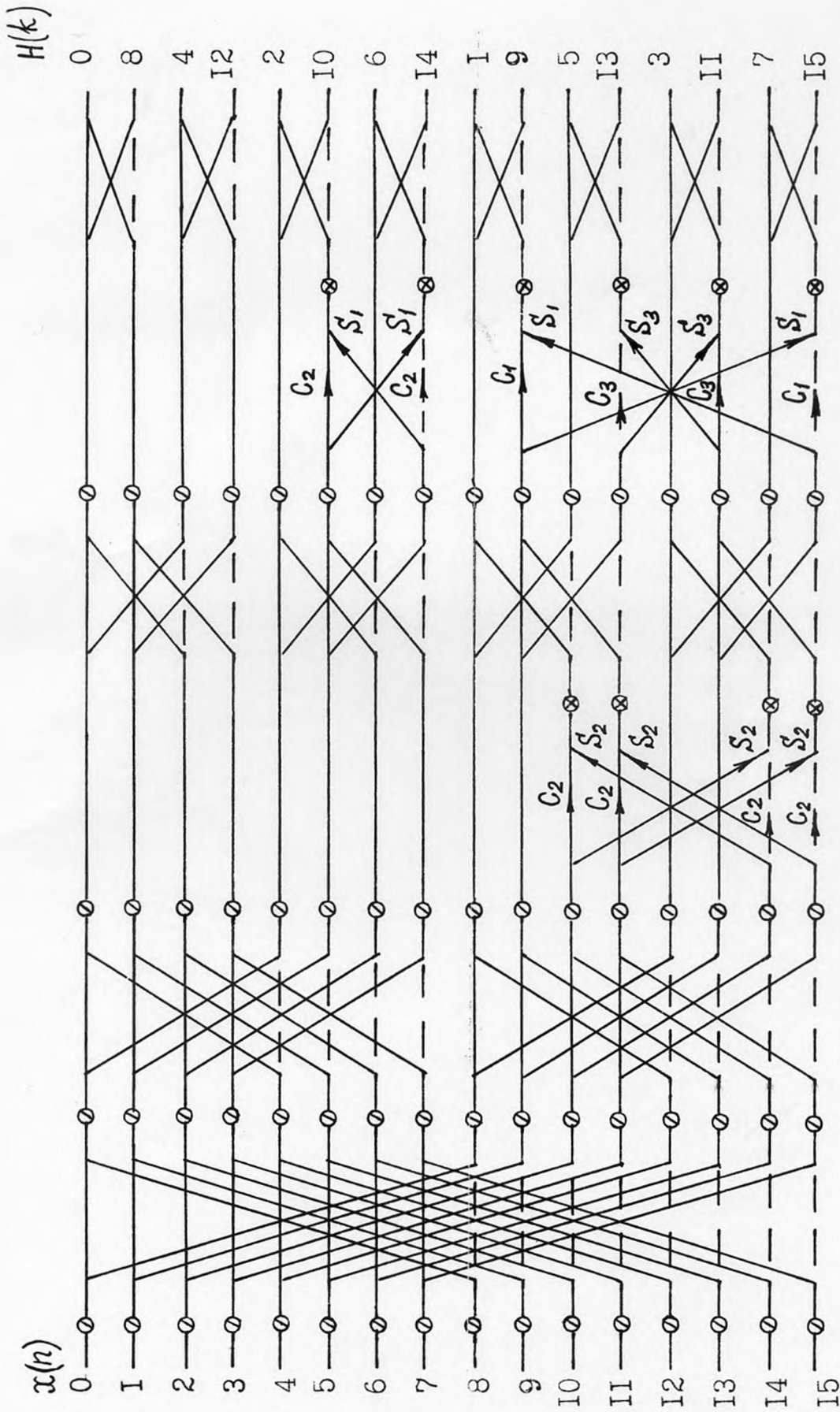


Рис.3.3.3. Граф-схема алгоритма БПХ по основанию 2 с прореживанием по времени с источниками вычислительных ошибок

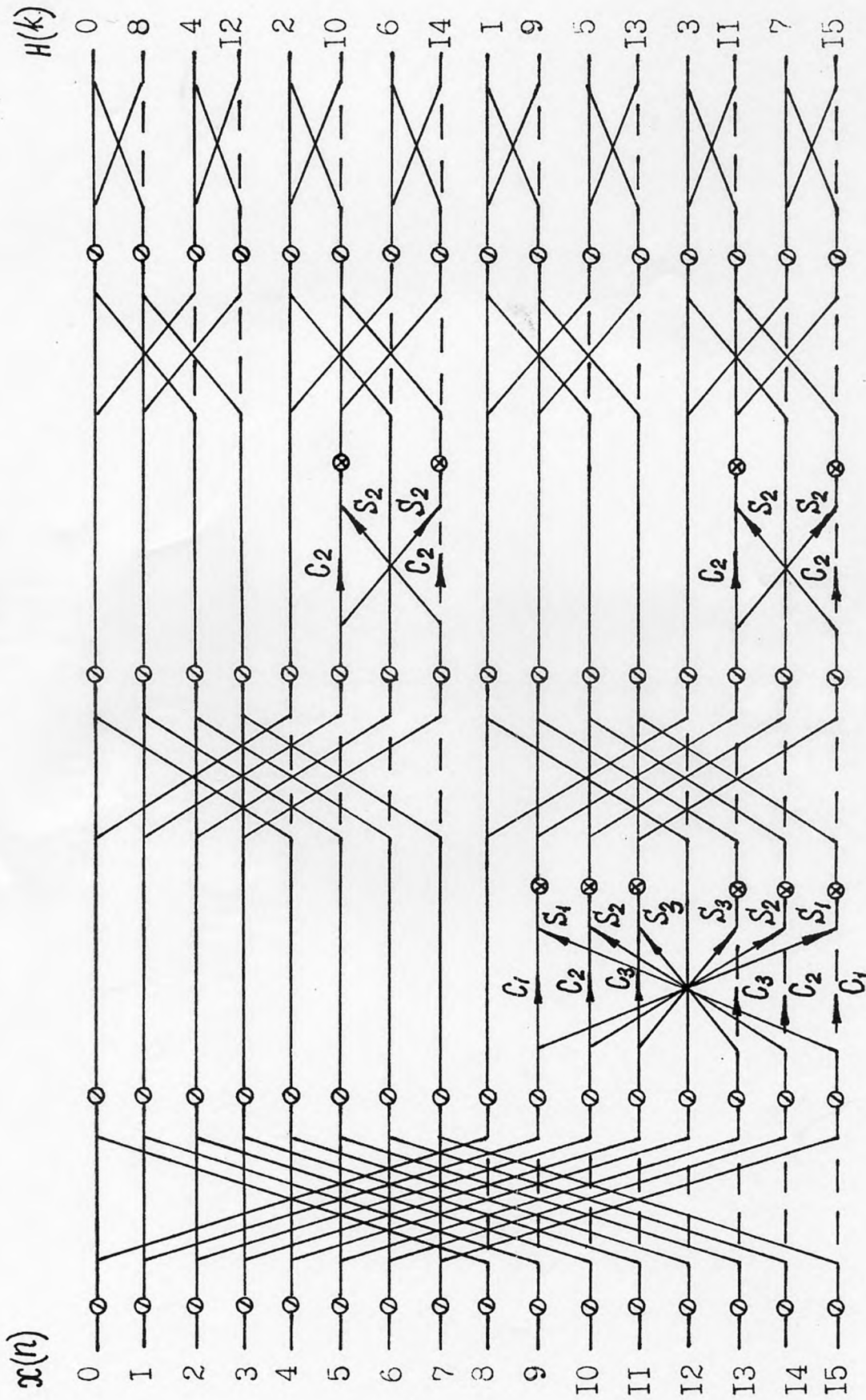


Рис.3.4. Граф-схема алгоритма БПХ по основанию 2 с прореживанием по частоте с источниками вычислительных ошибок

W_N^R не изменяется. Учитывая, что $C_R^2 + S_R^2 = 1$ можно показать, что в алгоритмах БПХ СКЗ ошибок на выходе этапов, на которых выполняются умножения на поворачивающие множители, равно СКЗ ошибок, имеющих на входе данных этапов. Таким образом, для исследования точности алгоритмов БПХ по основанию 2 можно воспользоваться аналогичными выражениями, полученными для алгоритмов БПФ по основанию 2. К аналогичным выводам можно прийти, анализируя вычислительные ошибки алгоритмов БПХ по другим основаниям. Используя аналитические выражения, полученные для алгоритмов БПФ, следует учесть, что ошибки сдвига и умножения являются вещественными, при этом СКЗ ошибки умножения равно

$$2M(\epsilon_y) + 2m_{\epsilon_y}^2,$$

где $M(\epsilon_y)$ и m_{ϵ_y} — соответственно СКЗ и математическое ожидание ошибки умножения двух вещественных чисел. Тогда очевидно, что СКЗ ошибок на выходе алгоритмов БПХ равно вдвое меньше СКЗ ошибок на выходе аналогичных алгоритмов БПФ.

Определим СКЗ сигнала на выходе алгоритма БПХ. Нетрудно показать, что в случае входного масштабирования для исключения возникновения переполнений входной сигнал должен быть такой, чтобы $|x(n)| < 1/N$ для всех n . Тогда при равномерном распределении СКЗ сигнала на выходе алгоритма будет равно $1/(3N)$. В случае автоматического масштабирования, анализируя основные вычислительные операции алгоритмов БПХ, можно определить, что входной сигнал должен быть такой, чтобы $|x(n)| < 1/\sqrt{2}$ для всех n . Тогда при равномерном законе распределения СКЗ сигнала на выходе алгоритма будет равно $1/(6N)$.

С учетом выше приведенных результатов можно сделать вывод [114], что точность алгоритмов БПХ при входном масштабировании равно вдвое выше точности аналогичных алгоритмов БПФ, а при авто-

матическом масштабировании алгоритмы БПХ и БПФ обеспечивают одинаковую точность.

3.4. Сравнительный анализ точности алгоритмов быстрого преобразования Фурье и Хартли

Используя результаты, полученные в предыдущих разделах, и СКЗ ошибок умножения и сдвига, приведенные в табл.1.7 и 1.8, можно получить аналитические выражения для определения вычислительных ошибок и точности в зависимости от длины обрабатываемой последовательности N и длины разрядной сетки, за исключением знакового бита, b . Аналитические выражения для определения точности алгоритмов БПФ и БПХ приведены в табл.3.1 и 3.2 соответственно для случаев входного и автоматического масштабирования.

В результате анализа точности алгоритмов БПФ и БПХ можно сделать следующие выводы:

1. При входном масштабировании точность алгоритмов с прореживанием по времени выше точности алгоритмов с прореживанием по частоте, за исключением алгоритмов с расщепленным основанием. Наилучшую точность среди алгоритмов с прореживанием по времени обеспечивают алгоритмы по основанию 2, а среди алгоритмов с прореживанием по частоте – алгоритмы по основанию 8, при этом первые обеспечивают точность в $6/5$ раза выше, чем последние.

2. При автоматическом масштабировании точность алгоритмов с прореживанием по частоте выше точности алгоритмов с прореживанием по времени. Наилучшую точность среди алгоритмов с прореживанием по частоте обеспечивают алгоритмы по основанию 2, а среди алгоритмов с прореживанием по времени – алгоритмы с расщепленным основанием, при этом первые обеспечивают точность в $6/5$ раза выше, чем последние.

3. Точность алгоритмов при автоматическом масштабировании выше точности алгоритмов при входном масштабировании для длин

Таблица 3.1

Точность алгоритмов БПФ и БПХ в случае входного масштабирования

Вид алгоритма	округление прямого, обратного и дополнительного кода		усечение прямого и обратного кода		усечение дополнительно-го кода	
	Точность	БПХ	Точность	БПФ	Точность	БПФ
По времени	Основание 2	$\frac{N^2}{6} 2^{-2b}$	$\frac{N^2}{12} 2^{-2b}$	$\frac{2N^2}{3} 2^{-2b}$	$\frac{2N^2}{6} 2^{-2b}$	$\frac{7N^2}{12} 2^{-2b}$
	Основание 4	$\frac{11N^2}{60} 2^{-2b}$	$\frac{11N^2}{120} 2^{-2b}$	$\frac{11N^2}{15} 2^{-2b}$	$\frac{11N^2}{30} 2^{-2b}$	$\frac{77N^2}{180} 2^{-2b}$
	Основание 8	$\frac{13N^2}{72} 2^{-2b}$	$\frac{13N^2}{144} 2^{-2b}$	$\frac{13N^2}{18} 2^{-2b}$	$\frac{13N^2}{36} 2^{-2b}$	$\frac{91N^2}{144} 2^{-2b}$
	Расцеплённое основание	$\frac{2N^2}{9} 2^{-2b}$	$\frac{N^2}{9} 2^{-2b}$	$\frac{8N^2}{9} 2^{-2b}$	$\frac{4N^2}{9} 2^{-2b}$	$\frac{7N^2}{9} 2^{-2b}$
По частоте	Основание 2	$\frac{N^2}{2} 2^{-2b}$	$\frac{N^2}{4} 2^{-2b}$	$2N^2 2^{-2b}$	$N^2 2^{-2b}$	$\frac{7N^2}{4} 2^{-2b}$
	Основание 4	$\frac{N^2}{4} 2^{-2b}$	$\frac{N^2}{8} 2^{-2b}$	$N^2 2^{-2b}$	$\frac{N^2}{2} 2^{-2b}$	$\frac{7N^2}{8} 2^{-2b}$
	Основание 8	$\frac{11N^2}{56} 2^{-2b}$	$\frac{11N^2}{112} 2^{-2b}$	$\frac{11N^2}{14} 2^{-2b}$	$\frac{11N^2}{28} 2^{-2b}$	$\frac{11N^2}{16} 2^{-2b}$
	Расцеплённое основание	$\frac{N^2}{5} 2^{-2b}$	$\frac{N^2}{10} 2^{-2b}$	$\frac{4N^2}{5} 2^{-2b}$	$\frac{2N^2}{5} 2^{-2b}$	$\frac{7N^2}{10} 2^{-2b}$

Таблица 3.2

Точность алгоритмов БПФ и БПХ в случае автоматического масштабирования

Вид алгоритма	! округление прямого, обратного и дополнительного кода !		! усечение прямого и обратного кода !		! усечение дополнительного кода !	
	Точность БПФ и БПХ	Точность БПФ и БПХ	Точность БПФ и БПХ	Точность БПФ и БПХ	Точность БПФ и БПХ	Точность БПФ и БПХ
Прореживание по времени	Основание 2	$5N \cdot 2^{-2b}$	$11N \cdot 2^{-2b}$	$17N \cdot 2^{-2b}$		
	Основание 4	$11N \cdot 2^{-2b}$	$23N \cdot 2^{-2b}$	$35N \cdot 2^{-2b}$		
	Основание 8	$\frac{165N}{7} \cdot 2^{-2b}$	$\frac{345N}{7} \cdot 2^{-2b}$	$75N \cdot 2^{-2b}$		
	Расщеплённое основание	$\frac{19N}{5} \cdot 2^{-2b}$	$\frac{31N}{5} \cdot 2^{-2b}$	$\frac{43N}{5} \cdot 2^{-2b}$		
Прореживание по частоте	Основание 2	$\frac{19N}{6} \cdot 2^{-2b}$	$\frac{11N}{3} \cdot 2^{-2b}$	$\frac{25N}{6} \cdot 2^{-2b}$		
	Основание 4	$\frac{43N}{60} \cdot 2^{-2b}$	$\frac{116N}{15} \cdot 2^{-2b}$	$\frac{497N}{60} \cdot 2^{-2b}$		
	Основание 8	$\frac{1129N}{72} \cdot 2^{-2b}$	$\frac{319N}{18} \cdot 2^{-2b}$	$\frac{1423N}{72} \cdot 2^{-2b}$		
	Расщеплённое основание	$\frac{29N}{9} \cdot 2^{-2b}$	$\frac{35N}{9} \cdot 2^{-2b}$	$\frac{41N}{9} \cdot 2^{-2b}$		

последовательностей $N > 32$, поскольку для первых она пропорциональна N , а для вторых — N^2 .

4. Алгоритмы с расщепленным основанием обеспечивают точность, близкую к точности лучших алгоритмов при прореживании по времени и частоте в случае входного и автоматического масштабирования, в силу чего их можно считать оптимальными.

В заключение рассмотрим некоторые вопросы повышения точности и связи динамического диапазона входного сигнала с точностью алгоритмов.

Анализ вычислительных ошибок алгоритмов БПФ и БПХ был проведен в предположении, что каждое из вещественных умножений вносит ошибку умножения. Однако, аппроксимацию результата комплексного произведения можно осуществить после определения точных значений действительной и мнимой частей результата. Тогда ошибки умножения в алгоритмах БПФ и БПХ будут иметь СКЗ $2M(\epsilon_y^2)$ и $M(\epsilon_y^2)$ соответственно. При этом в случае входного масштабирования точность алгоритмов с округлением результатов произведений повысится в 2 раза. В случае же автоматического масштабирования точность алгоритмов с округлением результатов произведений повысится в $(1,12 \div 1,25)$ раза при прореживании по времени и в $(1,02 \div 1,07)$ раза при прореживании по частоте.

Таким образом, за счет дополнительных затрат (аппаратурных или временных) можно повысить точность алгоритмов с прореживанием по времени. Точность алгоритмов с прореживанием по частоте повышается незначительно, поэтому для них использовать данный метод повышения точности нецелесообразно.

При исследовании точности в случае автоматического масштабирования предполагалось, что $|x(n)| < 1$ или $|x(n)| < 1/\sqrt{2}$ для всех n соответственно в алгоритмах БПФ и БПХ. При расширении динамического диапазона входного сигнала, при котором

$|x(n)| < \sqrt{2}$ или $|x(n)| < 1$ соответственно в алгоритмах БПФ и БПХ, для успешного выполнения алгоритмов необходимо осуществить сдвиг входного сигнала на $\log_2 L + 1$ разрядов вправо на первом этапе и на $\log_2 L$ разрядов вправо на последующих этапах алгоритмов. Поскольку СКЗ входного сигнала увеличится вдвое, и с учетом того, что на выходе алгоритмов будут $2N$ -ые части выходных гармоник, СКЗ выходного сигнала уменьшится вдвое. Тогда в предположении, что СКЗ выходной ошибки такое же как и в случае сдвига на $\log_2 L$ разрядов вправо на каждом этапе алгоритмов, точность алгоритмов БПФ и БПХ понизится в 2 раза.

ВЫВОДЫ

1. Точность алгоритмов БПХ при входном масштабировании вдвое выше точности аналогичных алгоритмов БПФ, а при автоматическом масштабировании алгоритмы БПХ и БПФ обеспечивают одинаковую точность.
2. При входном масштабировании точность алгоритмов с прореживанием по времени выше точности алгоритмов с прореживанием по частоте, за исключением алгоритмов с расщепленным основанием. Наилучшую точность среди алгоритмов с прореживанием по времени обеспечивают алгоритмы по основанию 2, а среди алгоритмов с прореживанием по частоте – алгоритмы по основанию 3, при этом первые обеспечивают точность в $6/5$ раза выше, чем последние.
3. При автоматическом масштабировании точность алгоритмов с прореживанием по частоте выше точности алгоритмов с прореживанием по времени. Наилучшую точность среди алгоритмов с прореживанием по частоте обеспечивают алгоритмы по основанию 2, а среди алгоритмов с прореживанием по времени – алгоритмы с расщепленным основанием, при этом первые обеспечивают точность в $6/5$ раза выше, чем последние.

4. Точность алгоритмов при автоматическом масштабировании выше точности алгоритмов при входном масштабировании для длин последовательностей $N \gg 2$, поскольку для первых она пропорциональна N , а для вторых - N^2 .

5. Алгоритмы с расщепленным основанием обеспечивают точность, близкую к точности лучших алгоритмов при прореживании по времени и частоте в случае входного и автоматического масштабирования, в силу чего их можно считать оптимальными.

6. Точность алгоритмов можно повысить, уменьшая количество вещественных ошибок умножения. При этом в случае входного масштабирования точность алгоритмов с округлением результатов произведений повысится вдвое, а в случае автоматического масштабирования точность алгоритмов с округлением результатов произведений повысится в $(1,12 \div 1,25)$ раза при прореживании по времени и в $(1,02 \div 1,07)$ раза при прореживании по частоте.

7. Расширение динамического диапазона входного сигнала при котором $|x(n)| < \sqrt{2}$ или $|x(n)| < 1$ соответственно в алгоритмах БПФ или БПХ приводит в случае автоматического масштабирования к снижению точности алгоритмов вдвое относительно точности алгоритмов с входным сигналом, при котором $|x(n)| < 1$ или $|x(n)| < 1/\sqrt{2}$ соответственно в алгоритмах БПФ или БПХ.

ГЛАВА 4. АППАРАТУРНЫЕ И ПРОГРАММНЫЕ СРЕДСТВА МИКРО-ПРОЦЕССОРНЫХ АНАЛИЗАТОРОВ СПЕКТРА

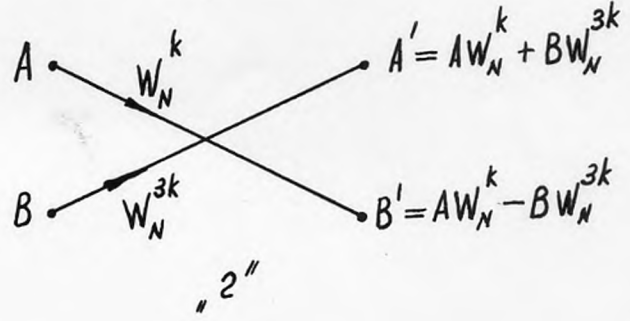
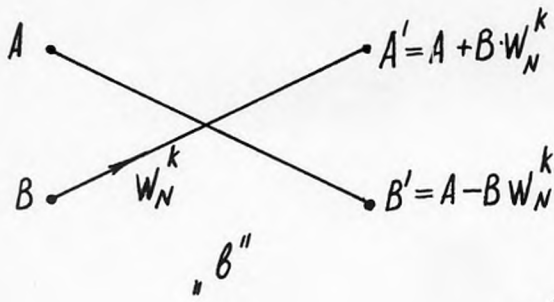
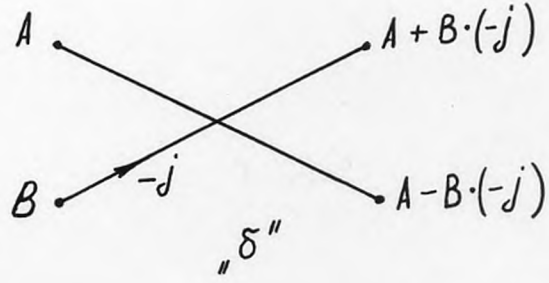
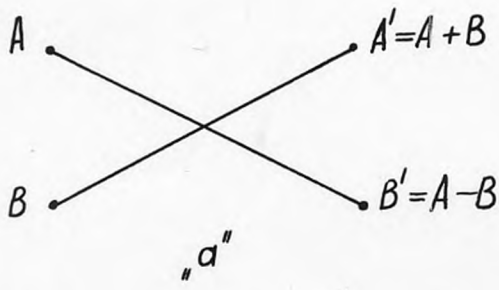
4.1. Программная реализация алгоритмов быстрого преобразования Фурье и Хартли

Рассмотрим организацию выполнения алгоритмов БПФ и БПХ. В качестве примера выберем алгоритмы по основанию 2 и с расщепленным основанием.

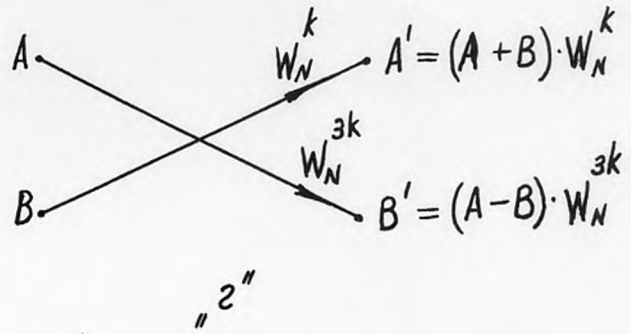
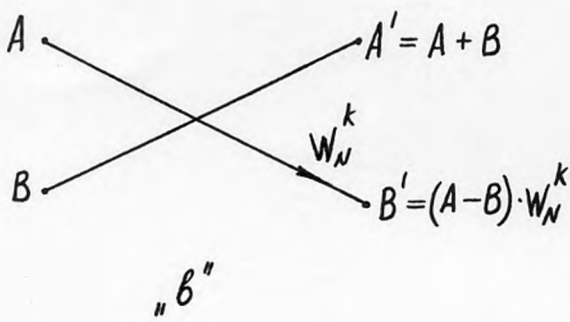
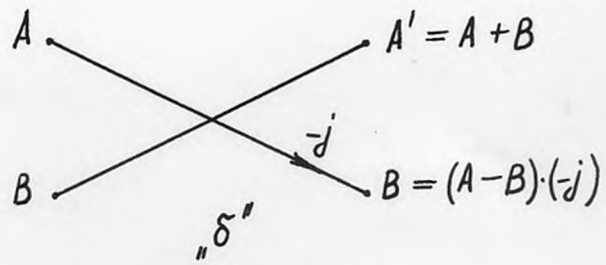
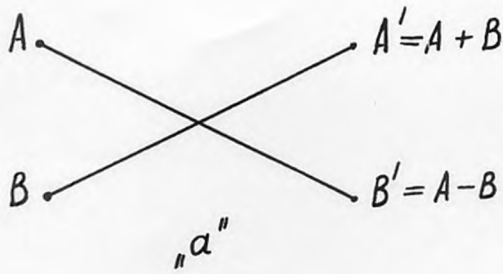
Анализ алгоритмов БПФ по основанию 2 и с расщепленным основанием показывает, что данные алгоритмы выполняются поэтапно, при этом количество этапов $m = \log_2 N$, где N — длина обрабатываемой последовательности. На каждом этапе базовые операции группируются в блоки, количество которых равно 2^l где $l = \overline{0, m-1}$. Смещение между отсчетами, являющимися операндами базовых операций, и количество базовых операций в блоке равно $N/2^{l+1}$.

Вид базовых операций, используемых в алгоритмах БПФ, представлен на рис. 4.1. При этом в алгоритмах БПФ по основанию 2 используются базовые операции "а", "б", и "в", а в алгоритмах БПФ с расщепленным основанием — базовые операции "а", "б" и "г".

В алгоритмах БПФ по основанию 2 с прямым порядком отсчетов на входе при прореживании по времени и двоично-инверсным порядком отсчетов на входе при прореживании по частоте в каждом блоке используются базовые операции одного вида, поэтому требуемый поворачивающий множитель $W_N^k = \exp(2\pi k/N)$, где $k = \overline{0, N/2-1}$, может быть определен один раз. При этом следует учесть, что на каждом этапе вычислений данных алгоритмов поворачивающие множители, используемые в базовых операциях очередного блока, следуют друг за другом в двоично-инверсном порядке.



a)



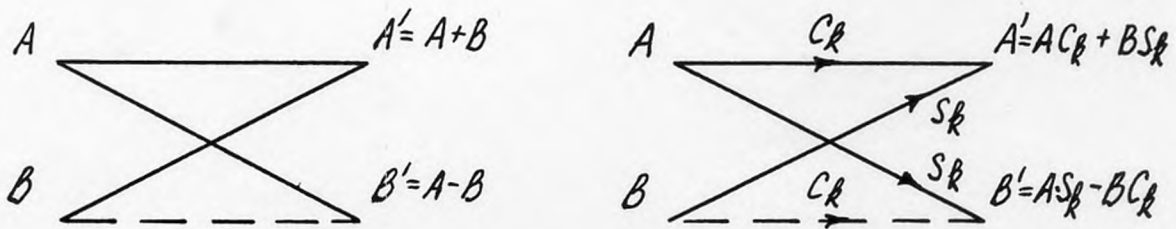
б)

Рис.4.Г. Базовые операции алгоритмов БПФ
 а) с прореживанием по времени
 б) с прореживанием по частоте

В алгоритмах БПФ с расщепленным основанием в каждом блоке вычислений, за исключением первого или последнего этапа вычислений соответственно в случае прореживания по времени и прореживания по частоте, используются различные базовые операции. При этом в блоках, маркируемых как блоки "0", первую половину базовых операций составляют базовые операции "а", а вторую половину базовых операций составляют базовые операции "б". В блоках, маркируемых как блоки "1", первой базовой операцией является базовая операция "а", а остальные базовые операции являются базовыми операциями "г". Для определения вида блока используется информационный вектор, содержащий признаки блоков "0" и "1" - 0 и 1 соответственно. Данный вектор строится за $\log_2 N - 1$ шага, начиная с 0. На каждом шаге информационный вектор увеличивается вдвое согласно правилу: $0 \begin{matrix} \leftarrow 0 \\ \leftarrow 1 \end{matrix}$, $1 \begin{matrix} \leftarrow 0 \\ \leftarrow 0 \end{matrix}$.

Анализ алгоритмов БПХ по основанию 2 и с расщепленным основанием показывает, что данные алгоритмы также как и алгоритмы БПФ выполняются поэтапно за $m = \log_2 N$ этапов, при этом каждый из этапов состоит из двух подэтапов, за исключением двух первых или двух последних соответственно в алгоритмах БПХ по основанию 2 с прореживанием по времени или частоте, а также первого и последнего в алгоритмах БПХ с расщепленным основанием. На одном из подэтапов выполняются базовые операции "а", на другом - базовые операции "б", которые представлены на рис.4.2. На каждом этапе базовые операции группируются в блоки, количество которых равно 2^l , где $l = 0, m-1$.

В алгоритмах БПХ по основанию 2 смещение между отсчетами, являющимися операндами базовой операции "а", и количество базовых операций "а" в блоке равно $N/2^{l+1}$. Количество базовых операций "б" в блоке равно $N/2^{l+2} - 1$, а смещение между отсчетами, являющимися операндами базовой операции



$$C_k = \cos(2\pi k/N), \quad S_k = \sin(2\pi k/N)$$

Рис.4.2. Базовые операции алгоритмов БПХ

"б", равно $N/2^{\ell+1} - 2\rho$, где $\rho = \overline{1, N/2^{\ell+2} - 1}$.

В алгоритмах БПХ с расщепленным основанием также как и в алгоритмах БПФ с расщепленным основанием используется информационный вектор для определения вида блока. При этом в блоке, которому ставится в соответствие признак 0, выполняются лишь базовые операции "а", количество и смещение между операндами которых равно $N/2^{\ell+1}$. В блоке, которому ставится в соответствие признак 1, выполняется $N/2^{\ell+1} - 1$ базовых операций "а", смещение между операндами которых равно $N/2^{\ell+1}$, и $N/2^{\ell+1} - 2$ базовых операций "б", смещение между операндами которых равно $N/2^{\ell+1} - 2\rho$, где $\rho = \overline{1, N/2^{\ell+2} - 1}$.

Изложенные выше особенности алгоритмов БПФ и БПХ положены в основу их программной реализации на языке ассемблера микропроцессора К580ИК80А. В приложениях 1 и 2 приведены соответственно листинги программ БПФ и БПХ по основанию 2 с прореживанием по времени и частоте. При этом для реализации алгоритмов БПФ по основа-

нию 2 использованы граф-схемы, представленные на рис. 3.1 и 3.2. В приложениях 3 и 4 приведены соответственно листинги программ БПФ и БПХ с расцепленным основанием с прореживанием по времени и частоте. В программах БПФ используются отдельные подпрограммы для выполнения базовых операций "а", "б", "в" и "г", а в программах БПХ – подпрограмма выполнения базовой операции "б". Исходными данными в программах БПФ и БПХ являются N – точечная соответственно комплексная или вещественная входная последовательность и массив поворачивающих множителей. Отсчеты входной и выходной последовательностей, а также поворачивающих множителей представлены в восьмиразрядном дополнительном коде. Размер преобразования N , начальные адреса входной последовательности и соответственно результатов преобразования, поворачивающих множителей и информационного вектора, а также некоторые другие параметры задаются в блоке параметров.

В приложении 5 приведен листинг программы-дополнения к программам БПФ с двоично-инверсным порядком отсчетов на выходе, используемый при обработке вещественной последовательности по алгоритму, определяемому выражениями (1.16) и (1.18).

Программы БПФ и БПХ используют подпрограммы умножения, в основе которых лежит эффективная программа быстрого умножения для микропроцессора КР580ИК80А [115]. Результаты умножения округляются.

Программа умножения может быть оптимизирована для алгоритмов с прореживанием по времени, поскольку в [116] показано, что в алгоритмах с прореживанием по времени операция умножения может быть выполнена быстрее, чем в алгоритмах с прореживанием по частоте, так как в первом случае максимально возможное значение входного отсчета базовой операции по абсолютной величине будет занимать не более v – 1 младших разрядов двоичного кода числа, где v – количество разрядов числа, исключая знаковый,

а во втором случае один из сомножителей может занимать младших разрядов числа.

Для исключения возможных переполнений при выполнении программ БПФ и БПХ на каждом этапе вычислений используется подпрограмма масштабирования, осуществляющая условное масштабирование, заключающееся в сдвиге на один разряд вправо (на первом этапе возможен сдвиг на два разряда вправо) всего обрабатываемого массива в случае необходимости.

Для упорядочивания отсчетов в прямом или двоично-инверсном порядке на входе или выходе алгоритмов БПФ и БПХ может быть использована подпрограмма двоично-инверсной перестановки, в основу которой положен эффективный алгоритм, представленный в [117].

Листинги подпрограмм умножения, масштабирования и двоично-инверсной перестановки приведены в приложении 6.

Результаты анализа программных реализаций алгоритмов БПФ и БПХ [114] приведены в табл.4.1.

Таблица 4.1

Характеристики программ БПФ и БПХ

Показатель	БПФ-2	Действит. с исполь- зованием БПФ-2	БПХ-2	БПФРО	Действит. с исполь- зованием БПФРО	БПХРО
Время выполнения, сек ($N = 1024$)	3,0	1,6	1,5	2,6	1,4	1,25
Объем кодов программы, байт	600	965	470	880	1245	715

Используя данные табл.4.1, а также результаты, полученные в предыдущих главах, для оценки объемов требуемой памяти и точности алгоритмов можно оценить вычислительную эффективность алгоритмов

БПХ относительно алгоритмов БПФ по комплексному критерию [92], учитывающему относительные оценки по быстродействию, точности вычислений, объему требуемой памяти и обратимость алгоритмов:

$$\mathcal{Z} = \frac{T_{БПФ}}{T_{БПХ}} \cdot \frac{\delta_{БПФ}}{\delta_{БПХ}} \cdot \frac{S_{БПФ}}{S_{БПХ}} \cdot R,$$

где $T_{БПХ}$, $\delta_{БПХ}$ и $S_{БПХ}$ - соответственно время выполнения, отношение шум/сигнал и требуемый объем памяти в алгоритмах БПХ; $T_{БПФ}$, $\delta_{БПФ}$ и $S_{БПФ}$ - аналогичные параметры алгоритмов БПФ; R - параметр, характеризующий обратимость алгоритмов ($R = 2$ для алгоритмов БПФД и $R = 1$ - для остальных).

Тогда в случае автоматического масштабирования алгоритмы БПХ в $2,6 \div 3,0$ раза превосходят алгоритмы БПФ для обработки вещественной последовательности и в $3,7 \div 4,1$ раза - алгоритмы БПФ для обработки комплексной последовательности.

4.2. Микропроцессорный анализатор спектра на базе микропроцессора КР580ИК80А

Микропроцессорный анализатор спектра [118, 119], дальнейшим развитием которого является мультимикропроцессорная система ЦОС [120], предназначен для определения энергетического спектра вещественных сигналов и его визуализации на экране монитора. Он имеет три фиксированных режима работы для обработки сигналов в разных полосах частот, которые могут быть заранее запрограммированы, и позволяет отобразить на экране монитора две последовательные гистограммы энергетического спектра, каждая из которых содержит 128 равномерно распределенных в обрабатываемой полосе частот гармоник.

Основные технические характеристики микропроцессорного анализатора спектра следующие: количество частотных полос обрабаты-

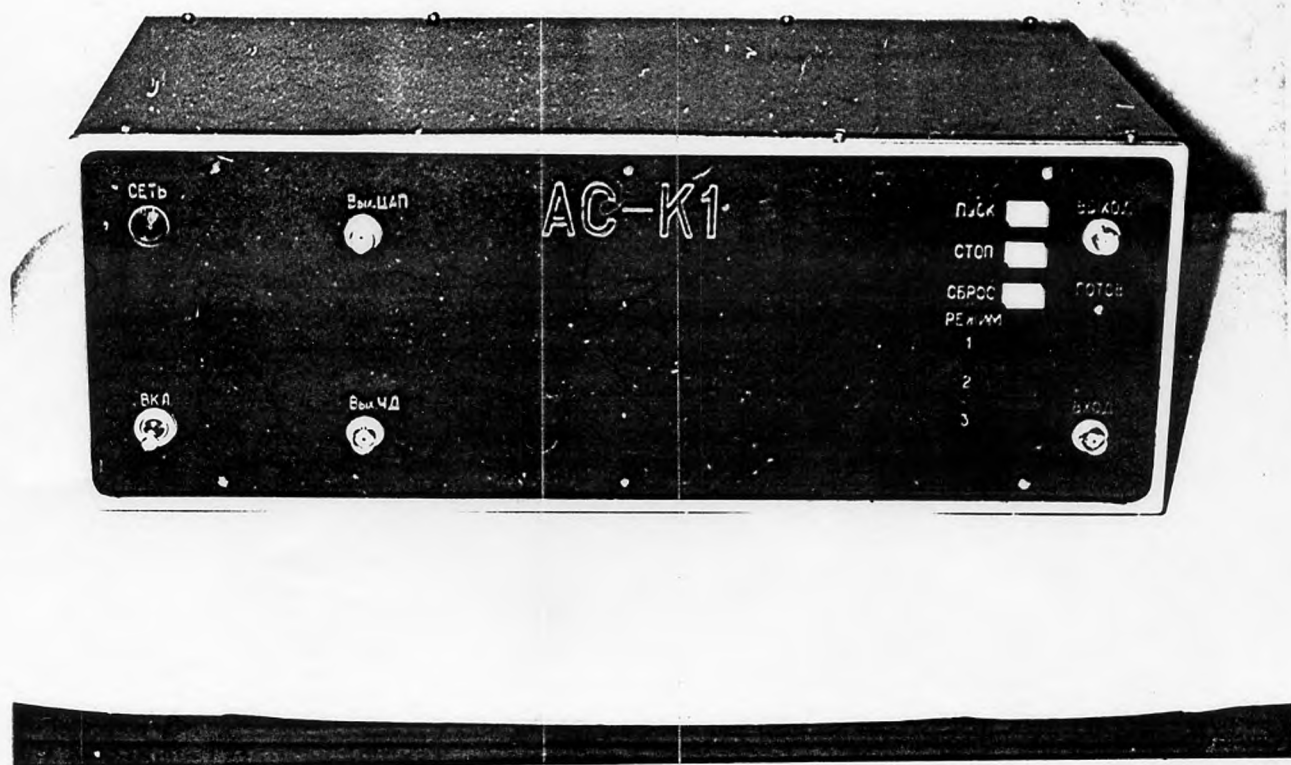


Рис.4.3'. Микропроцессорный анализатор спектра

ваемого сигнала - 3; число гистограмм энергетического спектра на экране монитора - 2; число гармоник энергетического спектра - 128; частота смены информации на экране монитора - 2 Гц; диапазон анализируемых частот - $0 \div 12,8$ кГц; амплитуды входного сигнала - ± 1 В; потребляемая мощность - 35 в; габариты - $410 \times 250 \times 150$ мм³, масса - 8,25 кг.

Внешний вид микропроцессорного анализатора спектра представлен на рис.4.3.

Структурная схема анализатора спектра представлена на рис.4.4. Анализатор состоит из трех основных модулей: ввода исходной информации 1, вычислителя энергетического спектра 2 и вывода результатов обработки на экран монитора 3.

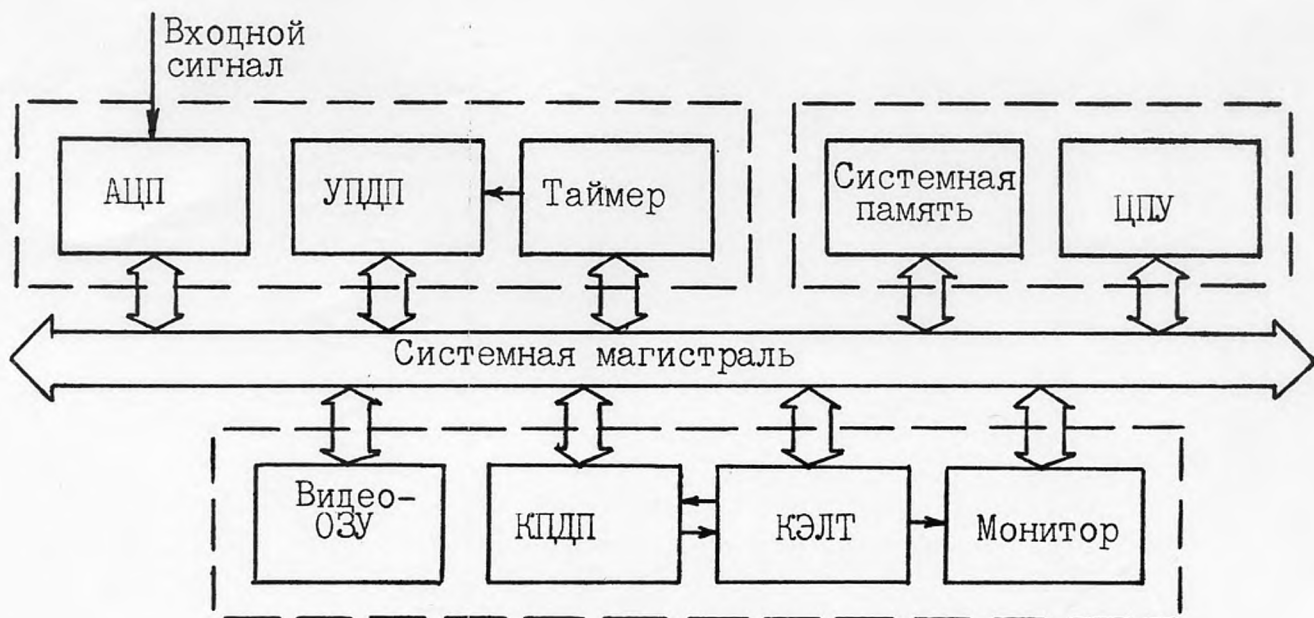


Рис.4.4. Структурная схема микропроцессорного анализатора спектра

4.2.1. Модуль ввода исходной информации

Модуль ввода исходной информации I преобразует входной аналоговый сигнал в восьми-разрядный дополнительный код и загружает его в системную память в режиме прозрачного прямого доступа к памяти (ПДП). Модуль содержит блок аналого-цифрового преобразования (АЦП), устройство ПДП (УПДП) и таймер.

В свою очередь блок АЦП включает в свой состав три аналоговых фильтра нижних частот (ФНЧ), предназначенных для подавления частот, не входящих в обрабатываемую полосу частот, с целью исключения эффекта наложения частот; аналоговый мультиплексор, коммутирующий ФНЧ, усилитель, устройство выборки-хранения, собственно АЦП на базе микросхемы К1113ПВ1, позволяющей кодировать аналоговый сигнал с частотой дискретизации ~ 30 кГц, и цифро-аналоговый преобразователь на базе микросхемы К1108ПА1, предназначенный для контроля аналого-цифрового преобразования. Частота дискретизации аналогового сигнала задается таймером, выполненным на базе микросхемы КР580ВИ53.

Устройство ПДП представлено в [121, 122] и предназначено для загрузки в режиме прозрачного ПДП кода, формируемого на выходе АЦП, в системную память. Данное устройство позволяет осуществить параллельную работу МП и режим ПДП за счет использования временных интервалов в первом машинном цикле работы МП, когда последний после извлечения кода команды из памяти занимается его декодированием и выполняет внутренние операции, в результате чего магистральные шины МП системы могут быть использованы для обмена информации в режиме ПДП.

На рис.4.5 представлена функциональная схема устройства; на рис.4.6 – временная диаграмма работы устройства.

Устройство (рис.4.5) содержит шинный формирователь I адреса, двунаправленный шинный формирователь 2 данных, шинный фор-

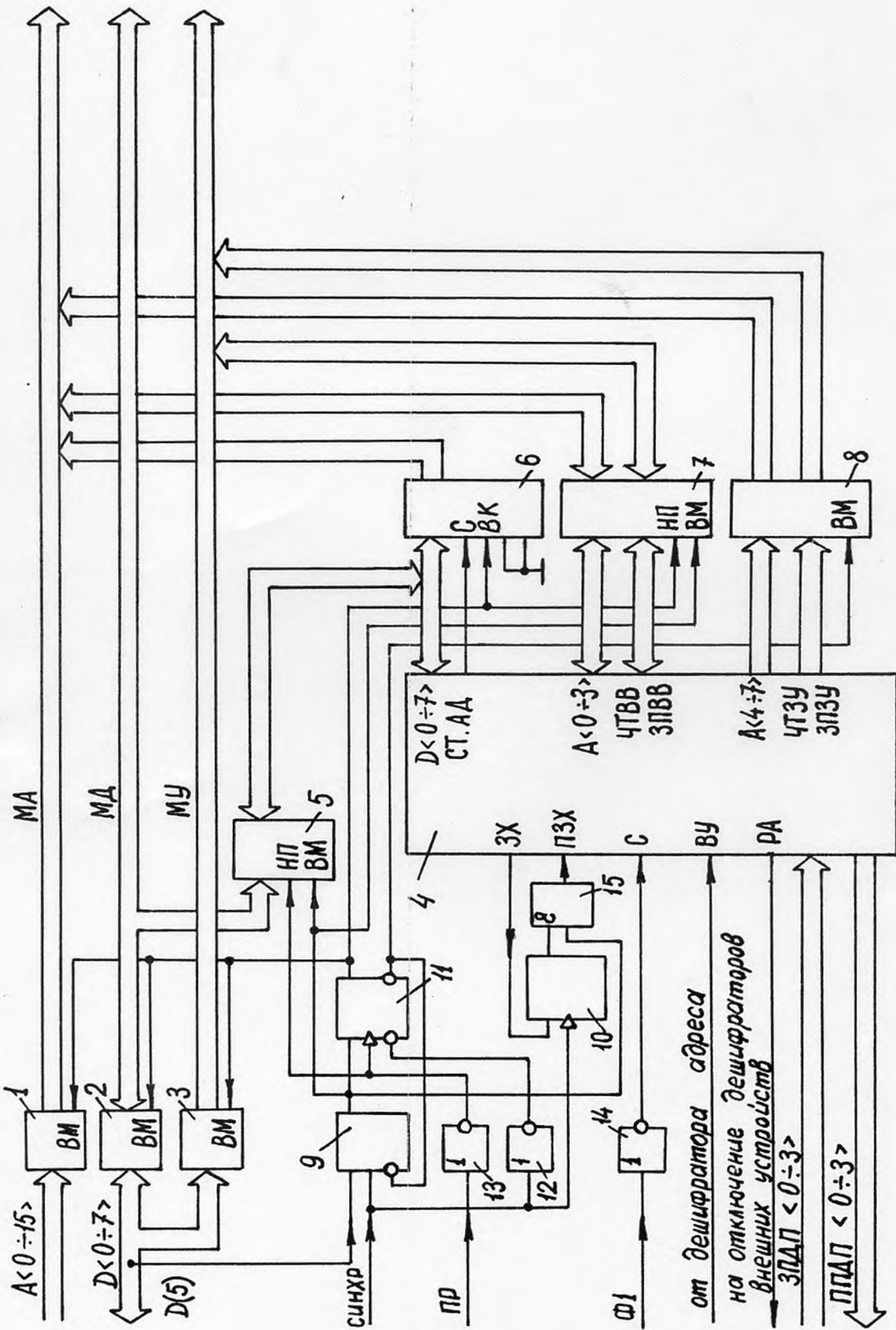


Рис.4.5. Функциональная схема устройства прозрачного ЦДП

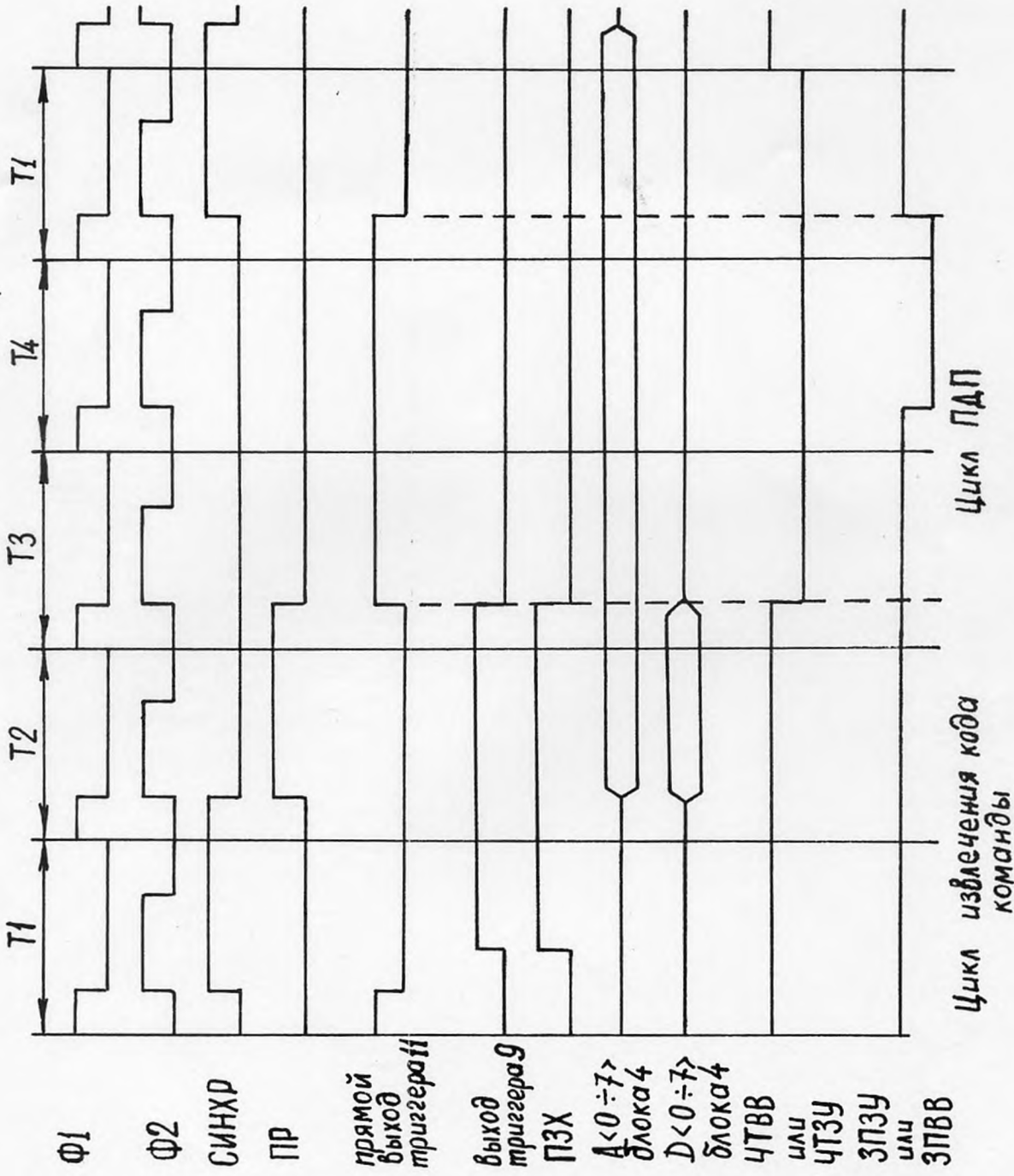


Рис.4.6.6. Временная диаграмма работы устройства прозрачного ЦДП

мирователь 3 сигналов управления, контроллер ЦДП 4 (БИС КР580ВТ57), первый двунаправленный коммутатор 5, буферный регистр 6, второй двунаправленный коммутатор 7, однонаправленный коммутатор 8, триггеры 9 - 11, элементы НЕ 12 - 14 и элемент И 15.

Устройство работает следующим образом.

При инициировании работы МП системы контроллер ЦДП 4, буферный регистр 6 и триггер 10 сбрасываются сигналом начальной установки. При этом сигнал "0" с прямого выхода триггера 10 поступает на вход элемента И 15, в результате чего на входе подтверждения захвата (ПЗХ) контроллера ЦДП 4 устанавливается сигнал "0", запрещая осуществление режима ЦДП.

В начале каждого машинного цикла триггер 11 сбрасывается по сигналу синхронизации (СИНХР), поступающему через элемент НЕ 12 на вход сброса триггера 11. При этом сигнал "0" с прямого выхода триггера 11 поступает на входы выбора микросхем (ВМ) шинного формирователя 1 адреса, двунаправленного шинного формирователя 2 данных и шинного формирователя 3 сигналов управления, подключая адресные выходы $A < 0 \div 15 >$ и информационные входы-выходы $D < 0 \div 7 >$ микропроцессора, а также сигналы управления, формируемые шинным формирователем 3, соответственно к магистрали адреса (МА), магистрали данных (МД) и магистрали управления (МУ) микропроцессорной системы. Этот же сигнал поступает на вход направления передачи (НП) двунаправленного коммутатора 7.

Одновременно с сигналом СИНХР на информационные входы-выходы $D < 0 \div 7 >$ микропроцессора выдается код слова состояния, пятый разряд Д (5) которого записывается в триггер 9. Если на прямом выходе триггера 9 устанавливается сигнал "0", то текущий машинный цикл не является первым машинным циклом команды. Следовательно, в этом машинном цикле может произойти обмен информацией между МП и контроллером ЦДП 4. Сигнал "0" с прямого выхода триггера 9 посту-

пает на входы ВМ двунаправленных коммутаторов 5 и 7, коммутируя информационные входы-выходы $D < 0 \div 7 >$, адресные входы-выходы $A < 0 \div 15 >$ и управляющие входы-выходы чтения из устройств ввода-вывода (ЧГВВ) и записи в устройства ввода-вывода (ЗПВВ) контроллера ПДП 4 соответственно с МД, МА и МУ МП системы. Таким образом, если на вход выбора устройства (ВУ) контроллера ПДП 4 с выхода дешифратора адреса поступает сигнал "0", то осуществляется обмен информацией между МП и контроллером ПДП 4. Направлением передачи управляет сигнал приема (ПР), подключенный через элемент НЕ I3 к входу НП двунаправленного коммутатора 5.

Если на прямом выходе триггера 9 во время действия сигнала СИНХР устанавливается уровень "1", то текущий машинный цикл является первым и, следовательно, в этом цикле может осуществиться режим ПДП (рис.4.6).

Сигнал "1" с прямого выхода триггера 9 поступает на вход ВМ двунаправленного коммутатора 5, отключая МД МП системы от информационных входов-выходов контроллера ПДП 4. Если внешнее устройство подало сигнал запроса ПДП (ЗПДП) на один из входов $ZПДП < 0 \div 3 >$ контроллера ПДП 4, которое в ответ на этот сигнал формирует сигнал захвата на выходе захвата (ЗХ), то по сигналу СИНХР происходит запись сигнала "1" в триггер I0. Таким образом, на обоих входах элемента И I5 устанавливаются сигналы "1", в результате чего на вход ПЗХ контроллера ПДП 4 поступает сигнал "1" и начинается выполнение режима ПДП. Вначале на выходе разрешения адреса (РА) контроллера ПДП 4 появляется сигнал на отключение дешифраторов адреса внешних устройств. По спаду тактового сигнала ФI в машинном такте T2 контроллер ПДП 4 выдает код младших разрядов адреса на выходы $A < 0 \div 3 >$ и старших разрядов адреса - на выходы $D < 0 \div 7 >$. В этот же момент времени на выходе стробирования адреса (СТ. АД) формируется активный сигнал, по которому старшие раз-

ряды адреса записываются в буферный регистр 6.

По окончании ИР, т.е. после извлечения кода команды и записи его в микропроцессор сигнал "1" с выхода триггера 9 записывается в триггер II, в результате чего сигнал "1" с прямого выхода триггера II поступает на входы ВМ шинного формирователя 1 адреса, двунаправленного шинного формирователя 2 данных и шинного формирователя 3 сигналов управления, отключая адресные выходы $A < 0 \div 15 >$ и информационные входы-выходы $D < 0 \div 7 >$ МП, а также управляющие сигналы, формируемые формирователем 3 сигналов управления, соответственно от МА, МД и МУ МП системы. Сигнал "1" поступает также на вход выбора кристалла (ВК) буферного регистра 6, в результате чего его выходы подключаются к МА МП системы, и на вход НИ двунаправленного коммутатора 7. Сигнал "0" с обратного выхода триггера II поступает на вход ВМ однонаправленного коммутатора 8, в результате чего сигналы с выходов $A < 4 \div 7 >$, чтения из запоминающегося устройства (ЧЗУ) и записи в запоминающее устройство (ЗЗУ) контроллера ПДП 4 подключаются соответственно к МА и МУ МП системы. Этот же сигнал поступает также на вход сброса триггера 9, в результате чего сигнал "0" с прямого выхода триггера поступает на вход ВМ двунаправленного коммутатора 7, подключая сигналы с выходов $A < 0 \div 3 >$, ЧТВВ и ЗПВВ контроллера ПДП 4 соответственно к МА, МУ МП системы.

По спаду тактового сигнала Ф1 в такте Т3 контроллер ПДП 4 вырабатывает сигнал ЧТВВ, по которому на МД МП системы выдается информация, извлеченная из устройства ввода-вывода. По спаду тактового сигнала Ф1 в такте Т4 контроллер ПДП 4 вырабатывает сигнал ЗПЗУ, по которому информация, находящаяся на МД устройства, записывается в память. Адрес ячейки памяти, в которую записывается информация, поддерживается на МА МП системы.

По сигналу СИНХР в следующем машинном цикле на прямом выходе триггера II устанавливается сигнал "0", который поступает на входы

ВМ шинного формирователя 1 адреса, двунаправленного шинного формирователя 2 данных и шинного формирователя 3 сигналов управления, подключая адресные выходы $A < 0 \div 15 >$ и информационные входы-выходы МП, а также сигналы управления, формируемые формирователем 3 сигналов управления соответственно к МА, МД и МУ МП системы. Этот же сигнал поступает на вход ВК буферного регистра 6, отключая его выходы от МА МП системы, и на вход НП двунаправленного коммутатора 7, переключая направление передачи. Сигнал "1" с обратного выхода триггера П1 поступает на вход ВМ однонаправленного коммутатора 8, отключая адресные выходы $A < 4 \div 7 >$ и выходы ЧТЗУ и ЗПЗУ соответственно от МА и МУ МП системы. На этом цикл передачи информации в режиме ПДП заканчивается, контроллер ПДП 4 сбрасывает свои адресные и управляющие сигналы, и устройство переходит в исходное состояние.

Таким образом, данное устройство прямого доступа к памяти осуществляет ввод исходной информации в режиме ПДП параллельно с работой МП, в результате чего в случае, когда время обработки не превышает время ввода информации, повышает производительность МП систем в 2 раза.

4.2.2. Модуль вычислителя энергетического спектра

Модуль вычислителя энергетического спектра 2 (рис.4.4) содержит системную память, предназначенную для хранения исходной информации, результатов обработки и программы вычислений, а также центральное процессорное устройство (ЦПУ) на базе МП КР580ИК80А.

Алгоритм вычислений энергетического спектра реализован программно на языке ассемблера МП КР580ИК80А. В основе вычислений лежит использование эффективной программы БПХ, вычисляющей 256-точечное ДПХ, что позволяет вычислить мгновенный энергетический спектр за $\sim 0,5$ сек.

4.2.3. Модуль вывода результатов обработки

Модуль вывода результатов обработки на экран монитора 3 (рис.4.4) предназначен для отображения на экране монитора гистограмм двух последовательных энергетических спектров входного сигнала, определенных в процессе обработки. Модуль реализован на основе устройства для отображения на экране электронно-лучевой трубки (ЭЛТ), представленного в [123], что позволяет осуществлять отображение информации на экране монитора без снижения производительности ЦПУ.

На рис. 4.7 представлена блок-схема устройства для отображения на экране ЭЛТ; на рис.4.8 - блок-схема контроллера ПДП; на рис.4.9 - временная диаграмма работы устройства.

Устройство (рис.4.7) содержит центральное процессорное устройство (ЦПУ) 1, системную память 2, контроллер ПДП 3, контроллер ЭЛТ 4 (БИС КР580ВГ75), ОЗУ 5 экрана, шинный формирователь 6, дешифратор 7, триггер 8, инвертор 9, элементы ИЛИ 10 и 11, элемент И-НЕ 12.

Контроллер ПДП (рис.4.8) содержит счетчик 16, параллельный сдвиговый регистр 17, элемент И-НЕ 18 и шинный формирователь 19.

Устройство работает следующим образом. В начале каждого машинного цикла работы ЦПУ 1 формируется сигнал синхронизации (СИНХР), который поступает на инвертор 9, в результате чего на второй вход элемента ИЛИ 10 поступает сигнал "0". По спаду тактового сигнала $\Phi 2$ с выхода ЦПУ 1, который поступает на первый вход элемента ИЛИ 10, с выхода элемента ИЛИ 10 сигнал "0" через элемент ИЛИ 11 поступает на вход сброса триггера 8, в результате чего сигнал "0" с прямого выхода триггера 8 поступает на вход элемента И-НЕ 12, запрещая формирование режима ПДП, и вход выбора микросхемы (ВМ) шинного формирователя 6, который соединяет первую и вторую части системной шины 13 и 14. Одновременно с этим

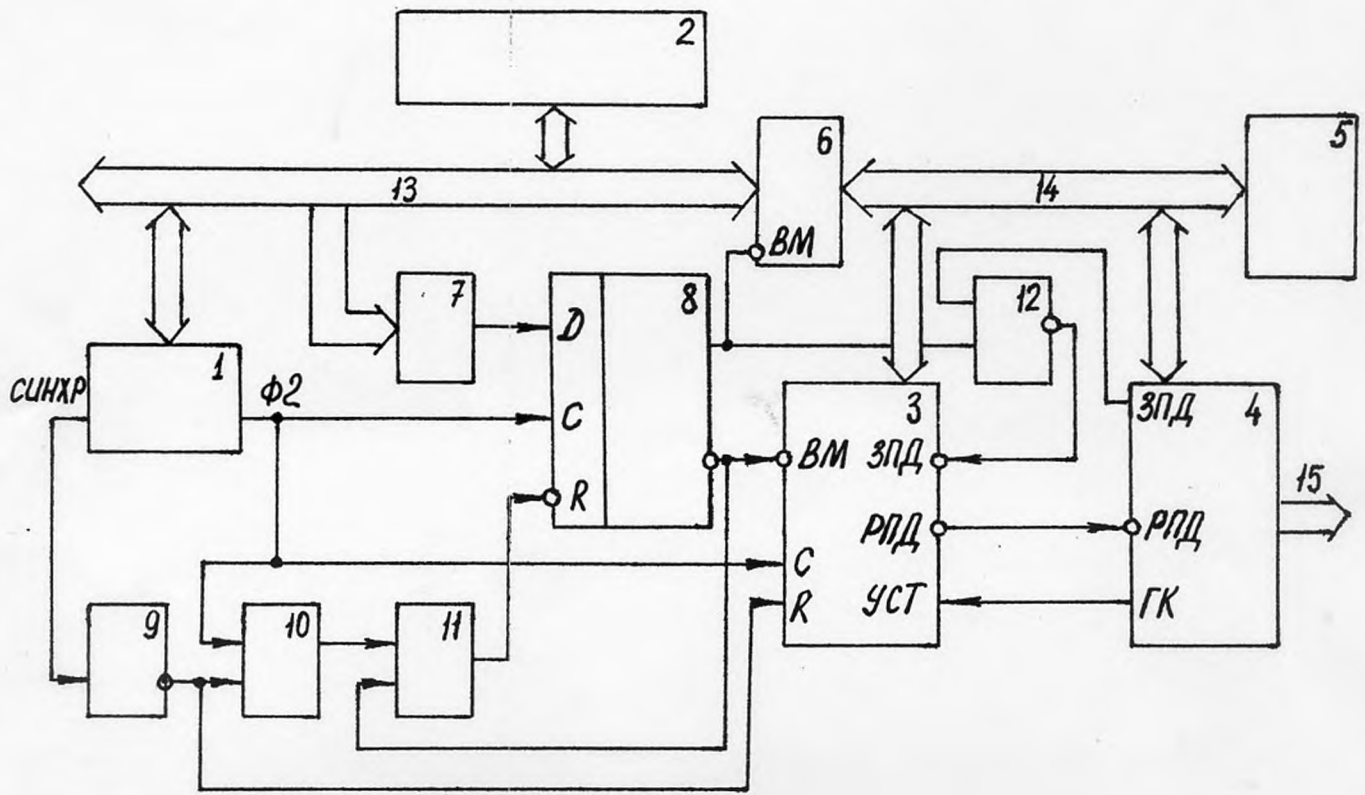


Рис.4.7. Блок-схема устройства для отображения на экране ЭЛТ

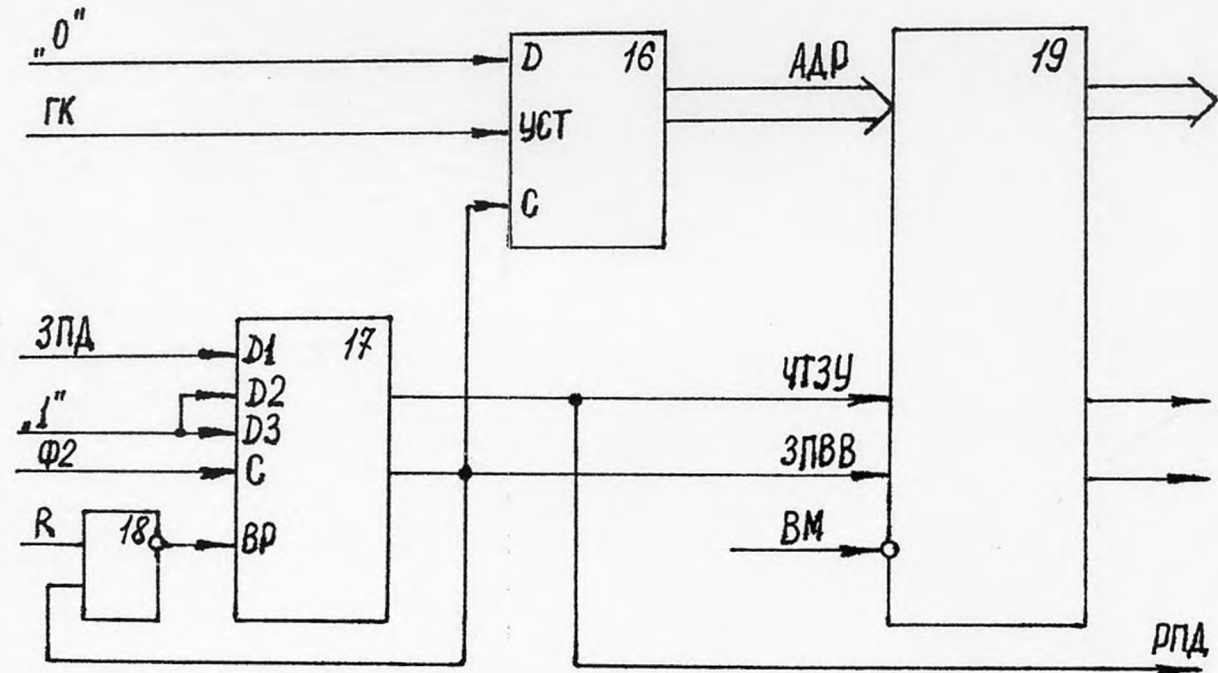


Рис.4.8. Блок - схема контроллера ПДП

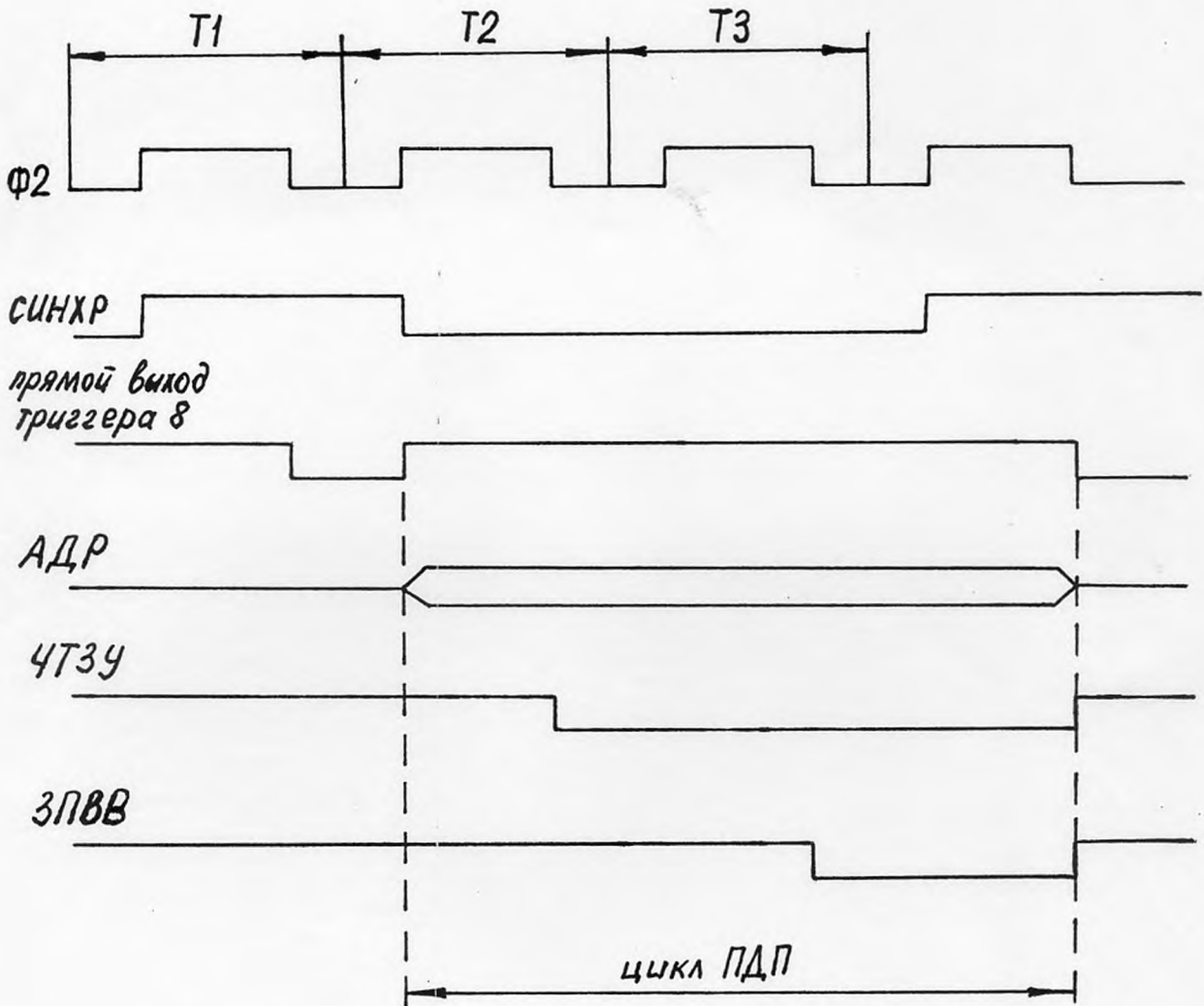


Рис.4.9. Временная диаграмма работы устройства для отображения на экране ЭЛТ

сигнал "1" с инверсного выхода триггера 8 поступает на вход ВМ шинного формирователя 19 контроллера ЦДП 3, в результате чего выходы контроллера ЦДП 3 переводятся в высокоимпенданское состояние, и на вход элемента ИЛИ 11, снимая сигнал сброса на входе сброса триггера 8. Если в текущем машинном цикле ЦПУ 1 обращается к ОЗУ 5 экрана или контроллеру ЭЛТ 4, то на выходе дешифратора 7 формируется сигнал "0", который по фронту $\Phi 2$ во втором машинном такте Т2 устанавливает на прямом выходе триггера 8 сигнал "0", т.е. состояние устройства остается прежним, и ЦПУ 1 может произвести обращение к ОЗУ 5 экрана или контроллеру ЭЛТ 4. В противном случае на прямом выходе триггера 8 устанавливается сигнал "1", который поступает на вход элемента И-НЕ 12, разрешая прохождение сигнала запроса прямого доступа (ЗПД) контроллера ЭЛТ 4 на односторонний вход контроллера ЦДП 3 и вход ВМ шинного формирователя 6, который разъединяет первую и вторую части системной шины 13 и 14. Сигнал "0" с инверсного выхода триггера 8 поступает на вход ВМ контроллера ЦДП 3, подключая его выходы ко второй части системной шины 14. Таким образом, в этом случае возможна параллельная работа ЦПУ 1 по первой части системной шины 13 и режим ЦДП для обслуживания контроллера ЭЛТ 4 по второй части системной шины 14.

Режим ЦДП осуществляется следующим образом (рис.4.9).

По фронту $\Phi 2$ в такте Т2 на вторую часть системной шины 14 подаются сигналы адреса (АДР) контроллера ЦДП 3, которые адресуют ячейку ОЗУ 5 экрана. Если к моменту спада $\Phi 2$ в такте Т2 контроллер ЭЛТ 4 подает сигнал ЗПД, то сигнал "0", который поступит при этом на первый информационный вход параллельного сдвигового регистра 17 по спаду $\Phi 2$ в такте Т2 появится на первом выходе регистра 17, так как на его входе выбора режима (ВР) присутствует сигнал "0", т.е. регистр 17 работает в режиме сдвигового регистра. Таким образом, будет сформирован сигнал чтения из запоминаю-

щего устройства (ЧГЗУ) и разрешения прямого доступа (РЦД). По спаду $\Phi 2$ в такте ТЗ на втором выходе регистра I7 появится сигнал "0", т.е. сформируется сигнал записи в устройство ввода-вывода (ЗПВВ). По спаду $\Phi 2$ в такте ТЗ сигналы ЧГЗУ и ЗПВВ сбрасываются, так как регистр I7 по сигналу ЗПВВ был предварительно установлен в режим параллельной записи. Кроме того, счетчик I6 по сбросу сигнала ЗПВВ сформирует следующий адрес обращения к ОЗУ 5 экрана.

Если текущий машинный цикл содержит три машинных такта, то описанный выше процесс повторится сначала. Если же текущий машинный цикл содержит четыре или пять машинных тактов и на входе ЗПД контроллера ПДП 3 присутствует активный сигнал, то может сформироваться очередной сигнал ЧГЗУ, однако сигнал ЗПВВ и следующий адрес ОЗУ 5 экрана не будут сформированы, поскольку регистр I7 по сигналу СИНХР установится в режим параллельной записи, так как на вход ВР регистр I7 через элемент И-НЕ I8 поступит сигнал "0".

Процесс регенерации изображения на экране ЭЛТ достигается тем, что в конце отображения кадра контроллер ЭЛТ формирует сигнал гашения кадра (ГК), который по входу установки (УСТ) счетчика I6 посылает на его выходы сигналы "0" с информационных входов, в результате чего устанавливается адрес начальной ячейки ОЗУ 5 экрана.

Таким образом, данное устройство для отображения на экране ЭЛТ позволяет полностью исключить простои ЦПУ при использовании контроллера ЭЛТ, который требует режим ПДП. Это достигается за счет разделения системной шины на две части для обеспечения возможности параллельной работы ЦПУ и режима ПДП, а также использования контроллера ПДП, требующего для выполнения цикла ПДП три машинных такта. Возможно также построение устройства для отображения на экране ЭЛТ, использующего контроллер ПДП, требующий для

выполнения цикла ЦДП четыре машинных такта [124]. Однако, в этом случае возможны незначительные простои ЦПУ.

Таким образом, с учетом результатов, приведенных в разделе 1.5, данные устройства позволяют повысить производительность МП систем в $1,125 \div 1,5$ раза.

В заключение отметим, что при реализации описанного выше микропроцессорного анализатора спектра использован синхронный режим работы устройств ввода, обработки и выдачи информации. В случае необходимости можно реализовать асинхронный режим, используя устройство для сопряжения МП с периферийными устройствами [125], которое позволяет в 2 раза повысить производительность МП систем за счет того, что интерфейсная магистраль предоставляется МП лишь на время цикла обмена информации, а в остальное время может быть использована другими активными устройствами.

В Ы В О Д Ы

1. Разработаны эффективные программные реализации алгоритмов БПФ и БПХ на языке ассемблера микропроцессора КР580ИК80А, анализ которых показал, что алгоритмы БПХ обеспечивают наибольшее быстроедействие при наименьших объемах требуемой памяти. При этом алгоритмы БПХ по комплексному критерию в 3 раза превосходят алгоритмы БПФ для обработки вещественной последовательности и в 4,1 раза - алгоритмы БПФ для обработки комплексной последовательности.

2. Разработаны устройства прозрачного ЦДП, отображения информации на экране ЭЛТ, сопряжения с периферийными устройствами, которые входят в состав ИИС ЦОС и повышают их производительность в 2 раза.

3. Разработан микропроцессорный анализатор спектра на базе микропроцессора КР580ИК80А, использующий устройства ввода и вы-

вода информации, обеспечивающие максимальную производительность анализатора, а также программу вычислений энергетического спектра на языке ассемблера, в основе которой лежит использование эффективного алгоритма БПХ, что позволило проводить оперативный спектральный анализ сигналов.

ЗАКЛЮЧЕНИЕ

1. Разработаны эффективные алгоритмы БПХ по основаниям 4, 8 и с расщепленным основанием с прореживанием по времени и частоте, которые по комплексному критерию превосходят аналогичные алгоритмы БПФ для обработки вещественных последовательностей в $2,6 \div 3$ раза, а алгоритмы БПФ для обработки комплексных последовательностей - в $3,7 \div 4,1$ раза. Наилучшими среди алгоритмов БПХ являются алгоритмы БПХ с расщепленным основанием.

2. Разработаны рекуррентные алгоритмы вычисления ДПХ и энергетического спектра на скачущих (скользящих) интервалах. Данные алгоритмы превосходят алгоритмы БПХ, обеспечивая максимальный выигрыш при вычислении на скользящих интервалах, который для $N = 1024$ составляет 2,34 и 5,15 раза соответственно по количеству операций вещественного умножения и вещественного сложения относительно алгоритмов БПХ с расщепленным основанием.

3. Исследована точность алгоритмов БПФ и БПХ с фиксированной запятой. Точность алгоритмов БПХ при входном масштабировании вдвое выше точности аналогичных алгоритмов БПФ, а при автоматическом масштабировании алгоритмы БПХ и БПФ обеспечивают одинаковую точность. При входном масштабировании точность алгоритмов с прореживанием по времени выше точности алгоритмов с прореживанием по частоте, а при автоматическом масштабировании - наоборот. Алгоритмы с расщепленным основанием обеспечивают точность, близкую к точности лучших алгоритмов при прореживании по времени и частоте в случае входного и автоматического масштабирования.

4. Разработаны эффективные программные реализации алгоритмов ЦОС, в частности, программы БПФ и БПХ на языке ассемблера микропроцессора КР580ИК80А с временем выполнения для 1024-точечной последовательности соответственно $2,6 \div 3$ и $1,25 \div 1,5$ сек.

5. Разработаны структурные и функциональные схемы устройств прозрачного ЦДП, отображения информации на экране ЭЛТ, сопряжения с периферийными устройствами, которые позволяют повысить производительность ИИС ЦОС в 2 раза.

6. На основе структурных и функциональных схем устройств обмена информации, а также эффективных программных реализаций алгоритмов ЦОС разработан микропроцессорный анализатор спектра на базе микропроцессора КР580ИК80А, предназначенный для оперативного спектрального анализа сигналов, который внедрен в ИРЭ АН УССР (г. Харьков). Годовой экономический эффект от внедрения составил 29,700 тыс.руб.

7. По результатам проведенных исследований опубликовано 17 печатных работ, в том числе 4 авторских свидетельства. Результаты работы докладывались на 6 научно-технических конференциях и семинарах.

ЛИТЕРАТУРА

1. Цапенко М.П. Измерительно-информационные системы. - М.: Энергия, 1974. - 320с.
2. Орнатский П.П. Теоретические основы информационно-измерительной техники. - 2-е изд. - К.: Выща школа, 1983. - 455с.
3. Кузьмин И.В. Оценка эффективности и оптимизации автоматических систем контроля и управления. - М.: Сов.радио, 1971. - 296с.
4. Кузьмин И.В., Кедрус В.А. Основы теории информации и кодирования. - К.: Выща школа, 1977. - 280с.
5. Стахов А.П., Лихтциндер Б.Я., Орлович Ю.П., Сторожук Ю.А. Кодирование данных в информационно-регистрирующих системах. К.: Техника, 1975. - 127с.
6. Страхов А.Ф. Автоматизированные измерительные комплексы. М.: Энергоиздат, 1982.
7. Капиев Р.Э. Измерительно-вычислительные комплексы. - Л.: Энергоатомиздат, Ленингр. отд-ние, 1988. - 176с.
8. Чеголин П.М. Автоматизация спектрального и корреляционного анализа. - М.: Энергия, 1969. - 384с.
9. Проектирование специализированных информационно-вычислительных систем: Учеб. пособие по спец. ЭВМ и АСУ / Смирнов Ю.М. и др.; под ред. Ю.М.Смирнова. - М.: Высш.школа, 1984. - 359с.
10. Арутюнов П.А. Теория и применение алгоритмических измерений. - М.: Энергоатомиздат, 1990. - 256с.
11. Грешилов А.А. Некоторые задачи цифровой обработки информации и сигналов. - М.: Радио и связь, 1984. - 160с.
12. Отнес Р., Энексон Л. Прикладной анализ временных рядов. - М.: Мир, 1982. - 428с.

13. Харкевич А.А. Спектры и анализ. - М.: Физматиз, 1962. - 236с.
14. Оппенгейм А.В., Шаффер Р.В. Цифровая обработка сигналов. М.: Связь, 1979. - 416с.
15. Рабинер Л.Р., Гоулд В. Теория и применение цифровой обработки сигналов. - М.: Мир, 1978. - 848с.
16. Бендат Дж., Пирсол А. Применение корреляционного и спектрального анализа. - М.: Мир, 1983. - 312с.
17. Цифровая обработка сигналов и ее применение, - М.: Наука, 1981. - 222с.
18. Применение цифровой обработки сигналов / Под ред. Оппенгейма.- М.: Мир, 1980. - 522с.
19. Макс Ж. Методы и техника обработки сигналов при физических измерениях. - М.: Мир, 1983 . - I т. - 312с.; - 2 т. - 256с.
20. Дженкинс Г., Ваттс Д. Спектральный анализ и его приложения.- М.: Мир, 1971. - Вып. I. - 314с.
21. Прэтт У. Цифровая обработка изображений. - М.: Мир, 1982. - I т. - 312с.; - 2 т. - 480с.
22. Залманзон Л.А. Преобразования Фурье, Уолша, Хаара и их применение в управлении, связи и других областях. - М.: Наука. Гл. ред. физ. - мат. лит., 1989. - 496с.
23. Логинов В.П. Функции Уолша и области их применения // Зарубежная радиоэлектроника. - 1973. - №4. - С.73 - 101.
24. Полонников Р.И., Костюк В.И., Краскевич В.Е. Матричные методы обработки сигналов. - К.: Техніка, 1977. - 136с.
25. Голубов Б.И., Ефимов А.В., Скворцов В.А. Ряды и преобразования Уолша: Теория и применения. - М.: Наука. Гл.ред. физ.-мат. лит., 1987. - 344с.
26. Ахмед Н., Рао К.Р. Ортогональные преобразования при обработке цифровых сигналов. - М.: Связь, 1980. - 248с.
27. Трахтман А.М. Введение в обобщенную спектральную теорию сигнала-

- лов. - М.: Советское радио, 1972. - 352с.
28. Трахтман А.М., Трахтман В.А. Основы теории дискретных сигналов на конечных интервалах. - М.: Советское радио, 1975. - 208с.
29. Пойда В.Н. Спектральный анализ в дискретных ортогональных базах. - Минск: Наука и техника, 1978. - 136с.
30. Дагман Э.Е., Кухарев Г.А. Быстрые дискретные ортогональные преобразования. - Новосибирск: Наука, 1983. - 232с.
31. Белый А.А., Бовбель Е.И., Микулович В.И. Алгоритмы быстрого преобразования Фурье и их свойства // Зарубежная радиоэлектроника. - 1979, - №2. - С.3 - 29.
32. Агарвал Р.С., Баррас Ч.С. Теоретико-числовые преобразования для быстрого вычисления цифровой свертки // ТИИЭР. - 1975. - т.63. - №4 - С.6 - 20.
33. Стручев В.Ф., Щёлкин Д.В. Использование числовых преобразований для цифровой обработки сигналов // Зарубежная радиоэлектроника. - 1975. - №11. - С.39 - 53.
34. Лабунец В.Г. Алгебраическая теория сигналов и систем (цифровая обработка сигналов). - Красноярск: Изд-во Краснояр. ун-та, 1984. - 244с.
35. Брейсуэлл Р. Преобразование Хартли: Пер. с англ. - М.: Мир, 1990. - 175с.
36. Шиков Н.Б. Дискретное преобразование Хартли для систем автоматизации эксперимента. - Минск, 1987. - 62с. (Препринт / АН БССР. - Ин-т техн. кибернетики, №28).
37. Влейхут Р. Быстрые алгоритмы цифровой обработки сигналов. - М.: Мир, 1989. - 448с.
38. Голд Б., Рэйдер Ч. Цифровая обработка сигналов. - М.: Сов.радио. - 1973. - 367с.
39. Макклелан Дж.Х., Рэйдер Ч.М. Применение теории чисел в цифро-

- вой обработке сигналов. - М.: Радио и связь. - 1983. - 264с.
40. Elliot D.F., Rao R. Fast transforms: Algorithms, analyses and applications. Academic Press, New York, 1983.
41. Нуссбаумер Г. Быстрое преобразование Фурье и алгоритмы вычисления свертки. - М.: Радио и связь. - 1985. - 248с.
42. Burrus C.S., Parks T.W. DFT/FFT and convolution algorithms: Theory and implementation - N.Y. etc: Wiley, 1985. - 245p.
43. Чеголин П.М., Шихов Н.Б. Эффективный алгоритм преобразования Хартли // Автоматизация технической подготовки производства. - Минск: Ин-т техн. кибернетика АН БССР, 1986. - С.138 - 151.
44. Meckelburg H.-j., Lipka D. Fast Hartley transform algorithm // Electronics Letters - 1985. - v.21. - N°8. - P.341 - 343.
45. Оппенгейм А.В., Вайнштейн К. Влияние конечной длины регистра при цифровой фильтрации и быстром преобразовании Фурье // ТИИЭР. - 1972. - т.60. - №8. - С.41 - 65.
46. Welch P.D. A fixed-point fast Fourier transform error analyses // IEEE Trans. Audio and Electroacoustics. - 1969. - v.17. - N°2. - P.151 - 157.
47. Trân-Thông, Liu B. Fixed-point fast Fourier transform error analysis // IEEE Trans. on ASSP. - 1976. - v.24. - N°6. - P.563 - 573.
48. Бовбель Е.И., Зайцева Е.М., Микулович В.И. Ошибки цифровых систем, основанных на вычислении дискретного преобразования Фурье // Зарубежная радиоэлектроника. - 1981. - №5. - С.3-25.
49. Байкова А.Г. Анализ вычислительных ошибок процессоров быстрого преобразования Фурье. - Л.: "Спец. астрофиз. обсер. АН СССР, Препринт". - 1987. - №44л. - 29с.
50. Knight W.R., Kaiser R. A simple fixed-point error bound for the fast Fourier transform // IEEE Trans. on ASSP. - 1979. -

v.27. - №6. - P.615 - 620.

51. Zakhor A., Oppengejm A.V. Quantization errors in the computation of the discrete Hartley transform // IEEE Trans. on ASSP. - 1987. - v.35. - №11. - P.1592 - 1602.
52. Белецкий А.Я. Рекуррентный алгоритм дискретного преобразования Фурье // Электронное моделирование. - 1987. - №2. - С.94 - 95.
53. Иваненко В.Г. Рекуррентное вычисление дискретного преобразования Фурье. М.: Препринт / МИФИ №014 - 87. - 1987. - 16с.
54. Бернанди А., Брейсуэлл Р.Н. Обновление спектральной функции действительного сигнала методом Хартли // ТИИЭР. - 1987. - т.75. - №7. - С.III - II2.
55. Петровский А.А. Методы и микропроцессорные средства обработки широкополосных и быстропротекающих процессов в реальном времени / Под ред. Г.В.Римского. - Мн. : Наука и техника, 1988. - 272с.
56. Цифровые фильтры и устройства обработки сигналов на интегральных микросхемах / Под ред. Б.Ф. Высоцкого. - М.: Радио и связь, 1984. - 216с.
57. Гамкрелидзе С.А. и др. Цифровая обработка информации на основе быстродействующих БИС / Под ред. В.Г.Домрачева. - М.: Энергоатомиздат, 1988. - 135с.
58. Вайрадян А.С., Пчелинцев И.П., Чельшев М.М. Алгоритмы вычисления цифровых сверток // Зарубежная радиоэлектроника, 1982, №3. - С.3 - 34.
59. Вольнец В.И., Мельниченко В.Ю., Чаюков И.Р., Гладко Е.К. Микропроцессорная система для цифровой обработки сигналов // Применение микропроцессоров, микро- и персональных ЭВМ. - Тезисы докладов научно-технического семинара. - Челябинск - 1989. - С.29 - 30.
60. Виноград С. О билинейных формах, мультипликативная сложность

- которых зависит от поля констант. - В кн. Маккелан Дж. Х., Рэйдер Ч.М. Применение теории чисел в цифровой обработке сигналов. - М.: Радио и связь, 1983. - С.225 - 233.
61. Buneman O. Inversion of the Helmholtz (or Laplace-Poisson) operator for slab geometry // J.Comput.Phys. - 1973. - v.12. - №1. - P.124 - 130.
62. Блейхут Р.Э. Алгебраические поля, обработка сигналов, контроль ошибок // ТИИЭР. - 1985. - т.73. - №5. - С.30 - 53.
63. Пакуловский В.Ф., Сведе-Швец В.Н., Соломин В.П. Алгоритм умножения комплексных чисел, использующий два действительных умножения // Вопросы кибернетики. Устройства и системы. - М. - 1988. - С.22 - 26.
64. Байков В.Д., Смолон В.Б. Специализированные процессоры: итерационные алгоритмы и структуры. - М.: Радио и связь, 1985. - 288с.
65. Бахтиаров Г.Д., Зиняков Н.Л. Алгоритм Волдера и его применение в вычислительной технике и цифровой обработке сигналов // Зарубежная радиоэлектроника. - 1983. - №3. - С.3 - 23.
66. Байков В.Д., Смолон В.Б. Аппаратурная реализация элементарных функций в ЦВМ. - Л.: Изд-во Ленинград. ун-та, 1975.
67. Карцев М.А. Арифметика цифровых машин. - М.: Наука, 1969. - 575с.
68. Акушский И.Я., Юдицкий Д.И. Машинная арифметика в остаточных классах. - М.: Сов.радио, 1968. - 439с.
69. Стахов А.П. Коды золотой пропорции. - М.: Радио и связь, 1984. - 152с.
70. Cooley J.W., Tukey J.W. An algorithm for the machine calculation of complex Fourier series // J. Math. Comput. - 1965. - v.19. - №90. - p.297 - 301.

71. Zolesio Jean-Luc. Transformations de Fourier descretetes algorithmes rapides // Ann. telecommun. - 1985. - v.40. - N°9 - 10. - P.495 - 507.
72. Гольденберг Л.М. и др. Цифровая обработка сигналов: Справочник. - М.: Радио и связь, 1985. - 312с.
73. Singleton R.C. On computing the fast Fourier transform // Commun. Assoc. Comput. Mach. - 1967. - N°10. - P.647 - 654.
74. Волюнец В.И. Сравнительный анализ алгоритмов быстрого преобразования Фурье и Хартли / Винница. политехн. ин-т. - Винница, 1990. - 16с. Деп. в УкрНИИИТИ 10.04.90, № 642 - Ук90.
75. Gentleman W.M., Sandy G. Fast Fourier transform for fun and profit // A FIPS Proceeding. Fall Joint Computer Conference. - 1966. - v.29. - P.563 - 578.
76. Bergland G.D. A fast transform algorithm using base 8 iterations // IEEE Trans. on ASSP. - 1968. - v.17. - N°2. - P.4-108.
77. Sorensen H.V., Heideman M.T., Burrus C.S. On computing the split-radix FFT // IEEE Trans. on ASSP. - 1986. - v.34. - N°1. - P.152 - 156.
78. Duhamel P., Hollman H. "Split-radix" FFT algorithm // Electron. Lett. - 1984. - v.20. - N°1. - P.14 - 16.
79. Bergland G.D. A fast Fourier transform algorithm for real-valued series // Commun. ACM - 1968. - v.11. - P.703 - 710.
80. Bergland G.D. A radix-eight fast Fourier subroutine for real-valued series // IEEE Trans. - 1969. - v.AU-17. - N°2.- P.138 - 144.
81. Duhamel P. Implementation of "split-radix" FFT algorithms for complex, real and real-symmetric data // IEEE Trans. on ASSP. - 1986. - v.34. - N°2. - P.285 - 295.

82. Sorensen H.V., Jones D.L., Burrus C.S. Real-valued algorithms for the FFT // ICASSP '87: Proc. Int. Conf. on ASSP, Dallas, Tex., Apr.6-9. - 1987. - v.3. - P.1831-1834.
83. Крот А.М., Минервина Е.Б. Быстрые алгоритмы гармонического анализа одномерных и двумерных действительных последовательностей. - Минск. - 1988. - 28с. (Препринт / Ин-т техн. кибернетики АН БССР, №25).
84. Ярославский Л.П. Цифровая обработка сигналов в оптике и голографии: Введение в цифровую оптику. - М.: Радио и связь, 1987. - 296с.
85. Вольнец В.И., Гладко Е.К., Осадчук В.С. Модифицированный алгоритм быстрого преобразования Фурье действительной последовательности. // Информатика и автоматизация в регионе. - Тезисы докладов республиканской научно-технической конференции. - Винница, 1988
86. Кончак В.С., Негода В.П. Алгоритм дискретного преобразования Фурье над полем действительных чисел // Автоматизация научных исследований. - Минск. - 1982. - С.13 - 25.
87. Ersoy O.K., Hu N.C. Fast algorithms for the real discrete Fourier transform // ICASSP '88: Int. Conf. on ASSP, New York City, Apr.11-14, 1988. - v.3. - P.1902 - 1905.
88. Брейсуэлл Р.Н. Быстрое преобразование Хартли // ТИИЭР. - 1984. - т.72. - №8. - С.19 - 27.
89. Прадо Ж. Замечания к статье "Быстрое преобразование Хартли" // ТИИЭР. - 1985. - т.73. - №12. - С.182 - 183.
90. Усатов В.М. Алгоритмы быстрого преобразования Хартли // техн. и прогр. средства автоматизации научн. исследований. - Алма-Ата, 1987. - С.117 - 125.
91. Яцимирский М.Н. Алгоритмы быстрого преобразования Хартли по основаниям 2 и 4 // Радиотехника (Москва). - 1989. - №2. - С.31 - 36.

92. Власенко В.А., Данильчук С.В. Сравнительное исследование вычислительной эффективности алгоритмов быстрых преобразований Фурье и Хартли // Рук. деп. в УкрНИНТИ 11.08.88, № 1932 - Ук88.
93. Гагарин Ю.И. Быстрые алгоритмы дискретного преобразования Хартли на основе матрично-рекурсивных форм // Рук. деп. в ВИНТИ 11.03.88, № 3515 - В88.
94. Власенко В.А., Лаппа Ю.М., Ярославский Л.П. Дискретное преобразование Хартли как альтернатива ДПФ в цифровой обработке сигналов // Изв. вузов. Радиоэлектроника. - 1989. - т.32.- № 12. - С.5 - 12.
95. Самофалов К.Г., Викторов О.В., Кузник А.К. Микропроцессоры.- К.: Техника, 1986. - 278с.
96. Микропроцессорные комплекты интегральных схем: Состав и структура: Справочник Под ред. А.А.Васенкова, В.А.Шахнова. - М.: Радио и связь, 1982. - 191с.
97. Микропроцессорные БИС и микро-ЭВМ. Построение и применение / Под ред. А.А.Васенкова. - М.: Сов.радио, 1980. - 279с.
98. Микропроцессорные комплекты БИС на основе интегральной инжекционной логики / Под ред. Э.П.Каложкина. М.: Радио и связь, 1984. - 246с.
99. Березенко А.И., Корягин Л.Н., Назарьян А.Р. Микропроцессорные комплекты повышенного быстродействия. - М.: Радио и связь, 1981. - 167с.
100. Абрайтис Б.Д., Седаускас С.Ю., Пятраускас А.В. Микропроцессорный комплект БИС высокого быстродействия К1800. - М.: Радио и связь, 1986. - 184с.
101. Басманов А.С., Широков Ю.Ф. Микропроцессоры и однокристалльные микроЭВМ: Номенклатура и функциональные возможности /

- Под ред. В.Г.Домрачева. - М.: Энергоатомиздат, 1988. - 128с.
102. Микропроцессоры и микропроцессорные комплекты интегральных микросхем: Справочник: В 2 т. / Под ред. В.А.Шахнова. - М.: Радио и связь, 1988.
103. Van der Auweraer H., Snoeys R. FFT implementation alternatives in advanced measurement systems // IEEE Micro - 1987. - v.7. - №1. - p.39 - 49.
104. Ушкар М.Н. Микропроцессорные устройства в радиоэлектронной аппаратуре / Под ред. Б.Ф. Высоцкого. - М.: Радио и связь, 1988. - 128с.
105. Скорик В.Н. и др. Мультипроцессорные системы. - К.: Техника, 1989. - 192с.
106. Кухарев Г.А., Тропченко А.Ю., Шмерко В.П. Системные процессоры для обработки сигналов. - Мн.: Беларусь, 1988. - 127с.
107. Кухарев Г.А., Шмерко В.П., Зайцева Е.Н. Алгоритмы и системные процессоры для обработки многозначных данных. - Мн.: Наука и техника, 1990. - 296с.
108. Панфилов Д.И., Романенко О.А., Сафонюк В.С., Шаронин С.Р. Принципы организации и работы дисплеев на основе БИС КР580ВГ75 // Микропроцессорные средства и системы. - 1985. - №3. - С.51 - 60.
109. Глазов А.Б., Костарев С.А., Суханова Е.В. Быстрое преобразование Фурье на микропроцессоре КР580ИК80А // Микропроцессорные средства и системы. - 1988. - №4. - С.48 - 49.
110. Волинец В.И. Алгоритмы быстрого преобразования Хартли по основаниям 4, 8 и с расщепленным основанием с прореживанием по времени и частоте / Винниц.политехн.ин-т. - Винница,

1990. - 35 с. - Деп. в УкрНИИНТИ 10.04.90, №641 - Ук90.
- III. Волюнец В.И. Рекуррентные алгоритмы вычисления дискретных преобразований и энергетического спектра / Винниц. политехн. ин-т. - Винница, 1988. - 14 с. - Деп. в УкрНИИНТИ 18.11.88, №2898 - Ук88.
- II2. А.с. №1488837. G 06 F 15/336. Устройство скользящего спектрально-корреляционного анализа / Осадчук В.С., Гикавый В.А., Прокспов И.Д. и Волюнец В.И. Винниц. политехн. ин-т.; Заяв. №4364243/24-24 от 26.10.87. Оpubл.23.06.89; Б.И. №23, С.233-234.
- II3. Волюнец В.И., Гикавый В.А. Анализ вычислительных ошибок алгоритмов быстрого преобразования Фурье в системе счисления с фиксированной запятой / Винниц. политехн. ин-т. - Винница, 1988. - 19 с. - Деп. в УкрНИИНТИ 18.11.88, №2897 - Ук88.
- II4. Волюнец В.И., Гикавый В.А. Микропроцессорная реализация алгоритмов спектрального анализа сигналов // Применение микропроцессоров, микро- и персональных ЭВМ. - Тезисы докладов научно-технического семинара. - Челябинск, 1989. - С.28 - 29.
- II5. Глазов А.Б., Костарев С.А., Суханова Е.В. Эффективные программы быстрого умножения для микропроцессора КР580ИК80А // Микропроцессорные средства и системы. - 1986. - №5. - С.43 - 44.
- II6. Волюнец В.И., Гикавый В.А., Осадчук В.С. К вопросу о выборе алгоритма быстрого преобразования Фурье // Программа и аннотации докладов 3-й Всесоюзной школы "Проектирование автоматизированных систем контроля и управления сложными объектами". - Харьков, 1988. - С.12 - 13.

117. Francik A., Koscielniak J. Fast algorithm of data permutation in discrete fast Fourier transform // J. Roy. Statist. Soc. Ser.C. Appl. Statist. - 1982. - v.31. - N 3. - P.327 - 330.
118. Волюнец В.И., Чаюков И.Р., Мельниченко В.Ю. Микропроцессорный анализатор спектра // Актуальные проблемы информатики, управления, радиоэлектроники и лазерной техники: Сборник докладов и сообщений Международной молодежной научно-технической конференции. - М., МГТУ им.Н.Э.Баумана, 1989. - С.38.
119. Волюнец В.И., Гладко Е.К., Мельниченко В.Ю., Чаюков И.Р. Анализатор спектра / Приборы и техника эксперимента. - 1990. - №1 - С.232.
120. Волюнец В.И., Гладко Е.К., Мельниченко В.Ю., Чаюков И.Р. Мультипроцессорная система цифровой обработки сигналов // Контроль, управление и автоматизация в современном производстве: Сборник докладов и сообщений Второй Всесоюзной конференции молодых ученых и специалистов с международным участием. - М.: НПО "Магистр", 1990. - С.67.
121. Гикавый В.А., Волюнец В.И., Щербацкий А.Д. Устройство прозрачного прямого доступа к памяти в микропроцессорной системе // XII Всесоюзная научная конференция по микроэлектронике. Тезисы докладов, часть IV. - Тбилиси, 1987.- С.143 - 144.
122. А.с. №1434443. G 06 F 13/00. Устройство прямого доступа к памяти / В.С.Осадчук, В.А.Гикавый, В.И.Волюнец и В.Ю.Мельниченко. Винниц.политехн. ин-т; Заяв. №4229975/24-24 от 13.04.87; Оpubл.30.10.88; Б.И. №40, С.232 - 233.
123. Положительное решение №4668140. G 06 F 3/153. Устройство

для отображения информации на экране электронно-лучевой трубки (ЭЛТ) / В.И.Волынец, В.Ю.Мельниченко, Е.К.Гладко и И.Р.Чаюков. Винниц. политехн. ин-т; Заяв. №4668I40/24-24 (043664) от 30.03.89.

124. Гикавый В.А., Волынец В.И., Щербацкий А.Д. Дисплейный модуль с микропроцессорной системой управления // XII Всесоюзная научная конференция по микроэлектронике. Тезисы докладов, часть IУ. - Тбилиси, 1987. - С.141-142.
125. А.с. №1434440. G 06 F 13/00. Устройство для сопряжения микропроцессора с периферийными устройствами / В.С.Осадчук, В.А.Гикавый, В.И.Волынец, В.Ю.Мельниченко. Винниц. политехн. ин-т; Заяв. №4229I67/24-24 от 13.04.87; Оpubл.30.10.88; Б.И. №40, С.230 - 231.

ПРИЛОЖЕНИЯ

```

;*****
;* ПРОГРАММЫ БЫСТРОГО ПРЕОБРАЗОВАНИЯ ФУРЬЕ ПО ОСНОВАНИЮ 2 *
;*****
;
CONST:      EQU      1      ;ПРЯМОЕ БПФ
SIZE:       EQU      256    ;РАЗМЕР ОБР. ПОСЛ-ТИ N
SIZE2:      EQU      512    ;N*2
DIS1:       EQU      128    ;N/2
DIS2:       EQU      64     ;N/4
BEGINSIGNAL: EQU      4000H  ;НАЧ. АДР. ОБР. ПОСЛ-ТИ
BEGINCOEFF: EQU      0780H  ;НАЧ. АДР. ТАБЛ. КОЭФФИЦ. W
ADR1:       EQU      BEGINSIGNAL+SIZE2;АДР.КОЛ-ВА ОБР.БЛОК.
ADR2:       EQU      ADR1+2 ;АДР. КОЛ-ВА БЛОКОВ НА ЭТАПЕ
ADR3:       EQU      ADR2+2 ;АДР. КОЛ-ВА ПАР В БЛОКЕ
ADR4:       EQU      ADR3+2 ;АДР. КОЛ-ВА ОБР. ПАР В БЛОКЕ
ADR9:       EQU      ADR4+2 ;АДР. ОПЕРАНДА А
ADR10:      EQU      ADR9+2 ;АДР. ОПЕРАНДА В
ADR11:      EQU      ADR10+2;АДР. ОПЕРАНДА W
ADR12:      EQU      ADR11+2;АДР. ВСПОМ. ЯЧЕЙКИ ПАМЯТИ
ADR13:      EQU      ADR12+1;АДР. ВСПОМ. ЯЧЕЙКИ ПАМЯТИ
ADR14:      EQU      ADR13+1;АДР. СЧЕТЧИКА МАСШТАБИР.
;
;*****
;* FFT1- ПРОГРАММА БПФ-2 (ПРОРЕЖИВАНИЕ ПО ВРЕМЕНИ) *
;*****
;
FFT1:       MVI      A,1     ;ЗАГР.НАЧ.ЗНАЧЕНИЯ СЧЕТЧИКА
            STA      ADR14   ;МАСШТАБИРОВАНИЯ
            LXI     H,DIS1   ;НАЧ.СМЕЩЕНИЕ И КОЛ-ВО ОБР.
            SHLD   ADR3     ;ПАР В БЛОКЕ
            SHLD   ADR4
            LXI     H,1     ;НАЧ.КОЛ-ВО ОБР. БЛОКОВ
ITERT:      SHLD   ADR1
            SHLD   ADR2
            CALL   SCALING ;МАСШТАБИРОВАНИЕ
            LXI     H,BEGINSIGNAL
            DCX    H
            DCX    H
            SHLD   ADR9
CYCL1T:     LHL    ADR3     ;ВЫПОЛНЕНИЕ ВЫЧИСЛЕНИЯ В БЛОКЕ
            DAD    H       ;С КОЭФФИЦ. W=1
            XCHG
            LHL    ADR9
            INX    H
            INX    H
            SHLD   ADR9
            DAD    D
            SHLD   ADR10
            CALL   BUT1
            LHL    ADR4   ;ПРОВЕРКА КОЛ-ВА ОБР. ПАР
            DCX    H
            SHLD   ADR4
            MOV    A,H
            ORA    L
            JNZ    CYCL1T

```

```

LHLD   ADR1   ;ПРОВЕРКА КОЛ-ВА ОБР. БЛОКОВ
DCX    H
SHLD   ADR1
MOV    A,H
ORA    L
JZ     EXAMT
LHLD   ADR10
SHLD   ADR9
LHLD   ADR3
SHLD   ADR4
CYCL2T: LHLD   ADR3   ;ВЫПОЛНЕНИЕ ВЫЧИСЛЕНИИ В БЛОКЕ
DAD    H       ;С КОЭФФИЦ. W=J
XCHG
LHLD   ADR9
INX    H
INX    H
SHLD   ADR9
DAD    D
SHLD   ADR10
CALL   BUT2T
LHLD   ADR4   ;ПРОВЕРКА КОЛ-ВА ОБР. ПАР
DCX    H
SHLD   ADR4
MOV    A,H
ORA    L
JNZ    CYCL2T
LHLD   ADR1   ;ПРОВЕРКА КОЛ-ВА ОБР. БЛОКОВ
DCX    H
SHLD   ADR1
MOV    A,H
ORA    L
JZ     EXAMT
LXI    H,BEGINCOEFF
BLOCKT: SHLD   ADR11
LHLD   ADR10
SHLD   ADR9
LHLD   ADR3
SHLD   ADR4
CYCL3T: LHLD   ADR3   ;ВЫПОЛНЕНИЕ ВЫЧИСЛЕНИИ В БЛОКЕ
DAD    H       ;С ПРОИЗВОЛЬНЫМИ КОЭФФИЦ. W
XCHG
LHLD   ADR9
INX    H
INX    H
SHLD   ADR9
DAD    D
SHLD   ADR10
CALL   BUT3T
LHLD   ADR4   ;ПРОВЕРКА КОЛ-ВА ОБР.ПАР
DCX    H
SHLD   ADR4
MOV    A,H
ORA    L
JNZ    CYCL3T
LHLD   ADR1   ;ПРОВЕРКА КОЛ-ВА ОБР. БЛОКОВ
DCX    H
SHLD   ADR1
MOV    A,H

```

```

ORA      L
JZ       EXAMT
LHLD    ADR11
INX     H
INX     H
SHLD    ADR11
JMP     BLOCKT
EXAMT:  LHLD    ADR3      ;ОПР-Е СМЕШЕНИЯ И КОЛ-ВА ОБР.
        MOV     A,H      ;ПАР В БЛОКЕ НА СЛЕД. ЭТАПЕ
        RAR
        MOV     H,A
        MOV     A,L
        RAR
        MOV     L,A
        RC
        SHLD    ADR3
        SHLD    ADR4
        LHLD    ADR2      ;ОПР-Е КОЛ-ВА БЛОКОВ НА СЛЕД.
        DAD     H        ;ЭТАПЕ
        JMP     ITERT

```

```

;*****
;*      BUT1- ПОДПРОГРАММА "БАБОЧКА", W=1      *
;*****
;      RE[A'] = RE[A] + RE[B], IM[A'] = IM[A] + IM[B]
;      RE[B'] = RE[A] - RE[B], IM[B'] = IM[A] - IM[B]
;-----

```

```

BUT1:   MOV     B,M      ;RE[B]
        INX     H
        MOV     C,M      ;IM[B]
        LHLD   ADR9
        MOV     A,M      ;RE[A]
        MOV     D,A
        ADD    B        ;RE[A]+RE[B]
        MOV     M,A
        INX     H
        MOV     A,M      ;IM[A]
        MOV     E,A
        ADD    C        ;IM[A]+IM[B]
        MOV     M,A
        LHLD   ADR10
        MOV     A,D
        SUB    B        ;RE[A]-RE[B]
        MOV     M,A
        INX     H
        MOV     A,E
        SUB    C        ;IM[A]-IM[B]
        MOV     M,A
        RET

```

```

;*****
;*      BUT2- ПОДПРОГРАММА "БАБОЧКА", W=J      *
;*****
;      ПБПФ: RE[A'] = RE[A] + IM[B], IM[A'] = IM[A] - RE[B]
;      RE[B'] = RE[A] - IM[B], IM[B'] = IM[A] + RE[B]
;      ОБПФ: RE[A'] = RE[A] - IM[B], IM[A'] = IM[A] + RE[B]
;      RE[B'] = RE[A] + IM[B], IM[B'] = IM[A] - RE[B]
;-----

```

```

BUT2:   MOV     B,M      ;RE[B]
        INX     H

```

```

MOV      C,M      ;IM[B]
LHLD    ADR9
MOV      D,M      ;RE[A]
MVI     A,CONST
CPI     01H
JNZ     IFFT1
MOV     A,D
ADD     C          ;RE[A]+IM[B]
MOV     M,A
INX    H
MOV     A,M      ;IM[A]
MOV     E,A
SUB     B          ;IM[A]-RE[B]
MOV     M,A
LHLD    ADR10
MOV     A,D
SUB     C          ;RE[A]-IM[B]
MOV     M,A
INX    H
MOV     A,E
ADD     B          ;IM[A]+RE[B]
MOV     M,A
RET
IFFT1:  MOV     A,D
SUB     C          ;RE[A]-IM[B]
MOV     M,A
INX    H
MOV     A,M      ;IM[A]
MOV     E,A
ADD     B          ;IM[A]+RE[B]
MOV     M,A
LHLD    ADR10
MOV     A,D
ADD     C          ;RE[A]+IM[B]
MOV     M,A
INX    H
MOV     A,E
SUB     B          ;IM[A]-RE[B]
MOV     M,A
RET
;*****
;*      BUT3T- ПОДПРОГРАММА "БАБОЧКА", W - ПРОИЗВОЛЬНОЕ      *
;*****
;      RE[A']=RE[A]+(RE[B]*RE[W]+IM[B]*IM[W])
;      IM[A']=IM[A]+(IM[B]*RE[W]-RE[B]*IM[W])
;      RE[B']=RE[A]-(RE[B]*RE[W]+IM[B]*IM[W])
;      IM[B']=IM[A]-(IM[B]*RE[W]-RE[B]*IM[W])
;      ОБПФ: IM[W]=-IM[W]
;-----
BUT3T:  MOV     C,M      ;RE[B]
LHLD    ADR11
MOV     E,M      ;RE[W]
CALL    MULTIPLICATION ;RE[B]*RE[W]
LXI     H,ADR12
MOV     M,B
LHLD    ADR10
INX    H
MOV     C,M      ;IM[B]

```

```

LHL D   ADR11
INX     H
MOV     E,M       ; IM[W]
MVI     A,CONST
CPI     01H
JZ      MULT1T
MOV     A,E
CMA
INR     A
MOV     E,A
MULT1T: CALL MULTIPLICATION ; IM[B]*IM[W]
LDA     ADR12
ADD     B         ; RE[B]*RE[W]+IM[B]*IM[W]
STA     ADR12
LHL D   ADR10
INX     H
MOV     C,M       ; IM[B]
LHL D   ADR11
MOV     E,M       ; RE[W]
CALL MULTIPLICATION ; IM[B]*RE[W]
LXI     H,ADR13
MOV     M,B
LHL D   ADR10
MOV     C,M       ; RE[B]
LHL D   ADR11
INX     H
MOV     E,M       ; IM[W]
MVI     A,CONST
CPI     01H
JZ      MULT2T
MOV     A,E
CMA
INR     A
MOV     E,A
MULT2T: CALL MULTIPLICATION ; RE[B]*IM[W]
LDA     ADR13
SUB     B         ; IM[B]*RE[W]-RE[B]*IM[W]
MOV     B,A
LXI     H,ADR12
MOV     D,M
LHL D   ADR9
MOV     A,M       ; RE[A]
MOV     E,A
ADD     D         ; RE[A]+(RE[B]*RE[W]+IM[B]*IM[W])
MOV     M,A
INX     H
MOV     A,M       ; IM[A]
MOV     C,A
ADD     B         ; IM[A]+(IM[B]*RE[W]-RE[B]*IM[W])
MOV     M,A
LHL D   ADR10
MOV     A,E
SUB     D         ; RE[A]-(RE[B]*RE[W]+IM[B]*IM[W])
MOV     M,A
INX     H
MOV     A,C
SUB     B         ; IM[A]-(IM[B]*RE[W]-RE[B]*IM[W])
MOV     M,A

```

RET

```

;*****
;*      FFT2- ПРОГРАММА БФФ-2 (ПРОРЕЖИВАНИЕ ПО ЧАСТОТЕ)      *
;*****

```

```

FFT2:      MVI      A,1      ;ЗАГР.НАЧ.ЗНАЧЕНИЯ СЧЕТЧИКА
           STA      ADR14    ;МАСШТАБИРОВАНИЯ
           LXI      H,DIS1   ;ЗАГР.НАЧ.КОЛ-ВА БЛОКОВ
           SHLD     ADR1
           SHLD     ADR2
ITERF:     LXI      H,1      ;ЗАГР.НАЧ.СМЕЩЕНИЯ И НАЧ.
           SHLD     ADR3    ;КОЛ-ВА ПАР В БЛОКЕ
           SHLD     ADR4
           CALL     SCALING ;МАСШТАБИРОВАНИЕ
           LXI      H,BEGINSIGNAL
           DCX      H
           DCX      H
           SHLD     ADR9
CYCL1F:    LHLD     ADR3    ;ВЫПОЛНЕНИЕ ВЫЧИСЛЕНИИ В
           DAD      H      ;БЛОКЕ С КОЭФФИЦ. W=1
           XCHG
           LHLD     ADR9
           INX      H
           INX      H
           SHLD     ADR9
           DAD      D
           SHLD     ADR10
           CALL     BUT1
           LHLD     ADR4    ;ПРОВЕРКА КОЛ-ВА ОБР. ПАР
           DCX      H
           SHLD     ADR4
           MOV      A,H
           ORA      L
           JNZ      CYCL1F
           LHLD     ADR1    ;ПРОВЕРКА КОЛ-ВА ОБР. БЛОКОВ
           DCX      H
           SHLD     ADR1
           MOV      A,H
           ORA      L
           JZ       EXAMF
           LHLD     ADR10
           SHLD     ADR9
           LHLD     ADR3
           SHLD     ADR4
CYCL2F:    LHLD     ADR3    ;ВЫПОЛНЕНИЕ ВЫЧИСЛЕНИИ В
           DAD      H      ;БЛОКЕ С КОЭФФИЦ. W=J
           XCHG
           LHLD     ADR9
           INX      H
           INX      H
           SHLD     ADR9
           DAD      D
           SHLD     ADR10
           CALL     BUT2F
           LHLD     ADR4    ;ПРОВЕРКА КОЛ-ВА ОБР. ПАР
           DCX      H
           SHLD     ADR4

```



```

MOV      A,H
ORA      L
JNZ      CYCL2F
LHLD     ADR1      ;ПРОВЕРКА КОЛ-ВА ОБР. БЛОКОВ
DCX      H
SHLD     ADR1
MOV      A,H
ORA      L
JZ       EXAMF
LXI      H,BEGINCOEFF
SHLD     ADR11
BLOCKF:  LHLD     ADR10
SHLD     ADR9
LHLD     ADR3
SHLD     ADR4
CYCL3F:  LHLD     ADR3      ;ВЫПОЛНЕНИЕ ВЫЧИСЛЕНИЯ В
DAD      H          ;БЛОКЕ С ПРОИЗВОЛЬНЫМИ
XCHG    ;КОЭФФИЦ. W
LHLD     ADR9
INX      H
INX      H
SHLD     ADR9
DAD      D
SHLD     ADR10
CALL     BUT3F
LHLD     ADR4      ;ПРОВЕРКА КОЛ-ВА ОБР. ПАР
DCX      H
SHLD     ADR4
MOV      A,H
ORA      L
JNZ      CYCL3F
LHLD     ADR1      ;ПРОВЕРКА КОЛ-ВА ОБР. БЛОКОВ
DCX      H
SHLD     ADR1
MOV      A,H
ORA      L
JZ       EXAMF
LHLD     ADR11
INX      H
INX      H
SHLD     ADR11
JMP     BLOCKF
EXAMF:   LHLD     ADR2      ;ОПР-Е КОЛ-ВА БЛОКОВ НА
MOV      A,H
RAR
MOV      H,A
MOV      A,L
RAR
MOV      L,A
RC
SHLD     ADR2
SHLD     ADR1
LHLD     ADR3      ;ОПР-Е СМЕЩЕНИЯ И КОЛ-ВА ПАР
DAD      H          ;В БЛОКЕ НА СЛЕД. ЭТАПЕ
JMP     ITERF

```

```

;*****
;*      BUT2F- ПОДПРОГРАММА "БАБОЧКА", W=J      *
;*****

```

```

;      ПЕПФ: RE[A'] = RE[A] + RE[B], IM[A'] = IM[A] + IM[B]
;      RE[B'] = IM[A] - IM[B], IM[B'] = RE[B] - RE[A]
;      ОБПФ: RE[A'] = RE[A] + RE[B], IM[A'] = IM[A] + IM[B]
;      RE[B'] = IM[B] - IM[A], IM[B'] = RE[A] - RE[B]
;

```

```

-----
BUT2F:      MOV      B,M      ;RE[B]
            INX      H
            MOV      C,M      ;IM[B]
            LHLD     ADR9
            MOV      A,M      ;RE[A]
            MOV      D,A
            ADD      B        ;RE[A]+RE[B]
            MOV      M,A
            INX      H
            MOV      A,M
            MOV      E,A
            ADD      C        ;IM[A]+IM[B]
            MOV      M,A
            LHLD     ADR10
            MVI      A,CONST
            CPI      01H
            JNZ      IFFT2
            MOV      A,E
            SUB      C        ;IM[A]-IM[B]
            MOV      M,A
            INX      H
            MOV      A,B
            SUB      D        ;RE[B]-RE[A]
            MOV      M,A
            RET
IFFT2:      MOV      A,C
            SUB      E        ;IM[B]-IM[A]
            MOV      M,A
            INX      H
            MOV      A,D
            SUB      B        ;RE[A]-RE[B]
            MOV      M,A
            RET

```

```

;*****
;*      BUT3F- ПОДПРОГРАММА "БАБОЧКА", W - ПРОИЗВОЛЬНОЕ      *
;*****
;      RE[A'] = RE[A] + RE[B]
;      IM[A'] = IM[A] + IM[B]
;      RE[B'] = (RE[A] - RE[B]) * RE[W] + (IM[A] - IM[B]) * IM[W]
;      IM[B'] = (IM[A] - IM[B]) * RE[W] - (IM[A] - IM[B]) * IM[W]
;      ОБПФ: IM[W] = -IM[W]
;

```

```

-----
BUT3F:      MOV      B,M      ;RE[B]
            INX      H
            MOV      C,M      ;IM[B]
            LHLD     ADR9
            MOV      A,M      ;RE[A]
            MOV      D,A
            ADD      B        ;RE[A]+RE[B]
            MOV      M,A
            INX      H
            MOV      A,M      ;IM[A]
            MOV      E,A

```

```

ADD      C          ; IM[A]+IM[B]
MOV      M,A
LHLD    ADR10
MOV      A,D
SUB      B          ; RE[A]-RE[B]
MOV      M,A
INX     H
MOV      A,E
SUB      C          ; IM[A]-IM[B]
MOV      M,A
MOV      C,A
LHLD    ADR11
INX     H
MOV      E,M       ; IM[W]
MVI     A,CONST
CPI     01H
JZ      MULT1F
MOV      A,E
CMA
INR     A
MOV      E,A
MULT1F: CALL    MULTIPLICATION ; (IM[A]-IM[B])*IM[W]
LXI     H,ADR12
MOV      M,B
LHLD    ADR10
MOV      C,M       ; (RE[A]-RE[B])
LHLD    ADR11
MOV      E,M       ; RE[W]
CALL    MULTIPLICATION ; (RE[A]-RE[B])*RE[W]
LDA     ADR12
ADD     B          ; (RE[A]-RE[B])*RE[W]+
STA     ADR12     ; (IM[A]-IM[B])*IM[W]
LHLD    ADR10
INX     H
MOV      C,M       ; (IM[A]-IM[B])
LHLD    ADR11
MOV      E,M       ; RE[W]
CALL    MULTIPLICATION ; (IM[A]-IM[B])*RE[W]
LXI     H,ADR13
MOV      M,B
LHLD    ADR10
MOV      C,M       ; (RE[A]-RE[B])
LHLD    ADR11
INX     H
MOV      E,M       ; IM[W]
MVI     A,CONST
CPI     01H
JZ      MULT2F
MOV      A,E
CMA
INR     A
MOV      E,A
MULT2F: CALL    MULTIPLICATION ; (RE[A]-RE[B])*IM[W]
LDA     ADR12
LHLD    ADR10
LDA     ADR13
SUB     B          ; (IM[A]-IM[B])*RE[W]-

```

I80

INX
MOV
RET

H
M,A

; (RESAJ-RE[B])*IM[W]

```

;*****
;*   ПРОГРАММЫ БЫСТРОГО ПРЕОБРАЗОВАНИЯ ХАРТЛИ ПО ОСНОВАНИЮ 2 *
;*****

```

```

SIZE:          EQU      256      ;РАЗМЕР ОБРАБАТ. ПОСЛЕД-ТИ N
BEGIN SIGNAL:  EQU      4000H     ;НАЧ. АДР. ПОСЛЕД-ТИ
BEGIN COEFF:   EQU      0780H     ;НАЧ. АДР. ТАБЛ. КОЭФФИЦ. W -2
DIS1:         EQU      128      ;N/2
DIS2:         EQU      64       ;N/4
DIS3:         EQU      2        ;N/128
ADR1:         EQU      BEGIN SIGNAL+SIZE ;АДР. КОЛ-ВА БЛОКОВ
ADR2:         EQU      ADR1+2    ;АДР. КОЛ-ВА ОБР. БЛОКОВ
ADR3:         EQU      ADR2+2    ;АДР. КОЛ-ВА ПАР В БЛОКЕ
ADR4:         EQU      ADR3+2    ;АДР. КОЛ-ВА ОБР. ПАР В БЛОКЕ
ADR5:         EQU      ADR4+2    ;АДР. СМЕЩЕНИЯ ДЛЯ КОЭФФИЦ.
ADR6:         EQU      ADR5+2    ;АДР. КОЛ-ВА ПАР УМНОЖЕНИЯ
ADR7:         EQU      ADR6+2    ;АДР. КОЛ-ВА ОБР. ПАР УМНОЖЕНИЯ
ADR8:         EQU      ADR7+2    ;АДР. НАЧ. АДРЕСА ТЕК. БЛОКА
ADR9:         EQU      ADR8+2    ;АДР. ОПЕРАНДА А
ADR10:        EQU      ADR9+2    ;АДР. ОПЕРАНДА В
ADR11:        EQU      ADR10+2   ;АДР. ОПЕРАНДА W
ADR12:        EQU      ADR11+2   ;АДР. ВСПОМ. ЯЧЕЙКИ ПАМЯТИ
ADR13:        EQU      ADR12+1   ;АДР. ВСПОМ. ЯЧЕЙКИ ПАМЯТИ
ADR14:        EQU      ADR13+1   ;АДР. СЧЕТЧИКА МАСШТАБИРОВАНИЯ

```

```

;*****
;*   FHT1- ПРОГРАММА БПХ-2 (ПРОРЕЖИВАНИЕ ПО ВРЕМЕНИ) *
;*****

```

```

FHT1:         MVI      A,1       ;НАЧ. ЗАГР. СЧЕТЧИКА МАСШТАБИР.
              STA      ADR14
              LXI     H,DIS1    ;НАЧ. КОЛ-ВО БЛОКОВ НА ЭТАПЕ
              SHLD   ADR1
              SHLD   ADR2
              LXI     H,1       ;НАЧ. КОЛ-ВО ОБР. ПАР В БЛОКЕ
              SHLD   ADR3
              LXI     H,DIS2    ;НАЧ. СМЕЩЕНИЕ ДЛЯ КОЭФФИЦ.
              SHLD   ADR5
              LXI     H,DIS3    ;НАЧ. КОЛ-ВО УМНОЖ. В БЛОКЕ
ITERH1:       SHLD   ADR6
              CALL   SCALING    ;МАСШТАБИРОВАНИЕ
              LXI     H,BEGIN SIGNAL ;ЗАГР. НАЧ. АДР. ОБР. ПОС-ТИ
              SHLD   ADR8
BLOCH1:       LHL    ADR3      ;ПРОВЕРКА НА НЕОБХ. ВЫПОЛНЕНИЯ
              MOV    A,H       ;БЛОКА УМНОЖЕНИЯ
              ORA    A
              JNZ   TOMUL1
              MVI   A,2
              CMP   L
              JNC   TOADD1
TOMUL1:       LXI     H,BEGIN COEFF
              SHLD   ADR11
              LHL    ADR6      ;ОПР-Е КОЛ-ВА ПАР В БЛОКЕ УМ-
              DCX   H          ;НОЖЕНИЯ
              SHLD   ADR7

```

```

LHLD   ADR3
XCHG
LHLD   ADR8
DAD    D
SHLD   ADR9
DAD    D
SHLD   ADR10
MULTH1: LHLD   ADR9   ;ВЫПОЛНЕНИЕ ПОДЭТАПА УМНОЖ.
        INX    H   ;В БЛОКЕ
SHLD   ADR9
LHLD   ADR10
DCX    H
SHLD   ADR10
LHLD   ADR11
XCHG
LHLD   ADR5
DAD    D
SHLD   ADR11
CALL   BUTH
LHLD   ADR7   ;ПРОВЕРКА КОЛ-ВА ОБР. ПАР В
DCX    H   ;БЛОКЕ УМНОЖЕНИЯ
SHLD   ADR7
MOV    A,H
ORA    L
JNZ    MULTH1
TOADD1: LHLD   ADR3   ;УСТ-КА СЧЕТЧ. КОЛ-ВА ОБР. ПАР
SHLD   ADR4   ;В БЛОКЕ
LHLD   ADR8   ;ОПР-НИЕ НАЧ. АДР. ОПЕРАНДОВ
XCHG   ;В БАБОЧКЕ
LHLD   ADR3
DAD    D
ADDH1: LDAX   D   ;ВЫЧИСЛЕНИЕ ПОДЭТАПА СЛОЖЕНИЯ-
MOV    B,A   ;ВЫЧИТАНИЯ В БАБОЧКЕ
ADD    M
STAX   D
MOV    A,B
SUB    M
MOV    M,A
INX    D
INX    H
PUSH   H
LHLD   ADR4   ;ПРОВЕРКА КОНЦА БЛОКА СЛОЖ.
DCX    H
SHLD   ADR4
MOV    A,H
ORA    L
POP    H
JNZ    ADDH1
SHLD   ADR8   ;ЗАПОМ. НАЧ. АДР. СЛЕД. БЛОКА
LHLD   ADR2   ;ПРОВЕРКА КОЛ-ВА ОБР. БЛОКОВ
DCX    H
SHLD   ADR2
MOV    A,H
ORA    L
JNZ    BLOCH1
LHLD   ADR1   ;ОПР-Е КОЛ-ВА БЛОКОВ НА СЛЕД.
MOV    A,H   ;ЭТАПЕ
RAR

```

```

MOV      H,A
MOV      A,L
RAR
MOV      L,A
RC
SHLD     ADR1
SHLD     ADR2
LHLD     ADR3      ;ОПР-Е СМЕЩЕНИЯ И КОЛ-ВА ОБР.
DAD      H          ;ПАР В БЛОКЕ СЛОЖЕНИЯ НА
SHLD     ADR3      ;СЛЕД. ЭТАПЕ
LHLD     ADR5      ;ОПР-Е СМЕЩЕНИЯ ДЛЯ КОЭФ. НА
MOV      A,H        ;СЛЕД. ЭТАПЕ
RAR
MOV      H,A
MOV      A,L
RAR
MOV      L,A
SHLD     ADR5
LHLD     ADR6      ;ОПР-Е КОЛ-ВА ПАР В БЛОКЕ УМ-
DAD      H          ;НОЖЕНИЯ НА СЛЕД. ЭТАПЕ
JMP      ITERH1
;*****
;*          BUTH- ПОДПРОГРАММА "БАБОЧКА"          *
;*****
;          A' = A * COSW + B * SINW
;          B' = A * SINW - B * COSW
;-----
BUTH:    MOV      E,M      ;ЗАГР. COS(K)
LHLD     ADR10     ;ЗАГР. ОПЕРАНДА А
MOV      C,M
CALL     MULTIPLICATION ;ОПР-Е A * COS(K) И ЕГО
LXI      H,ADR12   ;ЗАПОМИНАНИЕ
MOV      M,B
LHLD     ADR9      ;ЗАГР. ОПРЕРАНДА В
MOV      C,M
LHLD     ADR11     ;ЗАГР. SIN(K)
INX      H
MOV      E,M
CALL     MULTIPLICATION ;ОПР-Е B * SIN(K)
LDA      ADR12     ;ОПР-Е A * COS(K) + B * SIN(K)
ADD      B          ;И ЕГО ЗАПОМИНАНИЕ
STA      ADR12
LHLD     ADR10     ;ЗАГР. ОПЕРАНДА А
MOV      C,M
LHLD     ADR11     ;ЗАГР. SIN(K)
INX      H
MOV      E,M
CALL     MULTIPLICATION ;ОПР-Е A * SIN(K) И ЕГО
LXI      H,ADR13   ;ЗАПОМИНАНИЕ
MOV      M,B
LHLD     ADR9      ;ЗАГР. ОПЕРАНДА В
MOV      C,M
LHLD     ADR11     ;ЗАГР. COS(K)
MOV      E,M
CALL     MULTIPLICATION ;ОПР-Е B * COS(K)
LDA      ADR13     ;ОПР-Е A * SIN(K) - B * COS(K)
SUB      B
LHLD     ADR9      ;ЗАПОМИНАНИЕ РЕЗУЛЬТ. БАБОЧКИ

```

```

MOV      M,A
LDA      ADR12
LHLD     ADR10
MOV      M,A
RET

```

```

;*****
;*      FHT2- ПРОГРАММА БПХ-2 (ПРОРЕЖИВАНИЕ ПО ЧАСТОТЕ)      *
;*****

```

```

FHT2:      MVI      A,1      ;НАЧ.ЗАГР.СЧЕТЧИКА МАСШТАБИР.
           STA      ADR14
           LXI      H,1      ;НАЧ.КОЛ-ВО БЛОКОВ НА ЭТАПЕ
           SHLD     ADR1
           SHLD     ADR2
           LXI      H,DIS1   ;НАЧ. КОЛ-ВО ОБР. ПАР В БЛОКЕ
           SHLD     ADR3
           LXI      H,DIS3   ;НАЧ. СМЕЩЕНИЕ ДЛЯ КОЭФФИЦ.
           SHLD     ADR5
           LXI      H,DIS2   ;НАЧ. КОЛ-ВО УМНОЖ. В БЛОКЕ
ITERH2:    SHLD     ADR6
           CALL     SCALING ;МАСШТАБИРОВАНИЕ
           LXI      H,BEGIN SIGNAL ;ЗАГР.НАЧ.АДР.ОБР.ПОС-ТИ
           SHLD     ADR8
BLOCH2:    LHLD     ADR3      ;УСТ-КА СЧЕТЧ. КОЛ-ВА ОБР. ПАР
           SHLD     ADR4      ;В БЛОКЕ
           LHLD     ADR8      ;ОПР-НИЕ НАЧ. АДР. ОПЕРАНДОВ
           XCHG      ;В БАБОЧКЕ
           LHLD     ADR3
ADDH2:    DAD      D
           LDAX     D      ;ВЫЧИСЛЕНИЕ ПОДЭТАПА СЛОЖЕНИЯ-
           MOV      B,A      ;ВЫЧИТАНИЯ В БАБОЧКЕ
           ADD      M
           STAX     D
           MOV      A,B
           SUB      M
           MOV      M,A
           INX      D
           INX      H
           PUSH     H
           LHLD     ADR4      ;ПРОВЕРКА КОНЦА БЛОКА СЛОЖ.
           DCX      H
           SHLD     ADR4
           MOV      A,H
           ORA      L
           POP      H
           JNZ      ADDH2
           SHLD     ADR8      ;ЗАПОМ. НАЧ. АДР. СЛЕД. БЛОКА
           SHLD     ADR9      ;ЗАПОМ. НАЧ. АДР. БЛОКА УМНОЖ.
           XCHG
           SHLD     ADR10
           LHLD     ADR3      ;ПРОВЕРКА НА НЕОБХ. ВЫПОЛНЕНИЯ
           MOV      A,H      ;БЛОКА УМНОЖЕНИЯ
           ORA      A
           JNZ      TOMUL2
           MVI      A,2
           CMP      L
           JNC      COUNT

```



```

TOMUL2:    LXI      H,BEGINCOEFF
            SHLD    ADR11
            LHLD    ADR6      ;ОПР-Е КОЛ-ВА ПАР В БЛОКЕ УМ-
            DCX     H          ;НОЖЕНИЯ
            SHLD    ADR7
MULTH2:    LHLD    ADR10     ;ВЫПОЛНЕНИЕ ПОДЭТАПА УМНОЖЕНИЯ
            INX     H          ;В БЛОКЕ
            SHLD    ADR10
            LHLD    ADR9
            DCX     H
            SHLD    ADR9
            LHLD    ADR11
            XCHG
            LHLD    ADR5
            DAD     D
            SHLD    ADR11
            CALL   BUTH
            LHLD    ADR7     ;ПРОВЕРКА КОЛ-ВА ОБР. ПАР В
            DCX     H          ;БЛОКЕ УМНОЖЕНИЯ
            SHLD    ADR7
            MOV     A,H
            ORA     L
            JNZ    MULTH2
COUNT:    LHLD    ADR2     ;ПРОВЕРКА КОЛ-ВА ОБР. БЛОКОВ
            DCX     H
            SHLD    ADR2
            MOV     A,H
            ORA     L
            JNZ    BLOCH2
            LHLD    ADR3     ;ОПР-Е СМЕЩЕНИЯ И КОЛ-ВА ОБР.
            MOV     A,H       ;ПАР В БЛОКЕ УМНОЖЕНИЯ НА
            RAR     ;СЛЕД. ЭТАПЕ
            MOV     H,A
            MOV     A,L
            RAR
            MOV     L,A
            RC
            SHLD    ADR3
            LHLD    ADR1     ;ОПР-Е КОЛ-ВА БЛОКОВ НА СЛЕД.
            DAD     H          ;ЭТАПЕ
            SHLD    ADR1
            SHLD    ADR2
            LHLD    ADR5     ;ОПР-Е СМЕЩЕНИЯ ДЛЯ КОЭФ. НА
            DAD     H          ;СЛЕД. ЭТАПЕ
            SHLD    ADR5
            LHLD    ADR6     ;ОПР-Е КОЛ-ВА ПАР В БЛОКЕ УМ-
            MOV     A,H       ;НОЖЕНИЯ НА СЛЕД. ЭТАПЕ
            RAR
            MOV     H,A
            MOV     A,L
            RAR
            MOV     L,A
            JMP     ITERH2

```

```

;*****
;*   ПРОГРАММЫ БЫСТРОГО ПРЕОБРАЗОВАНИЯ ФУРЬЕ С РАСЩЕПЛЕННЫМ *
;*   ОСНОВАНИЕМ *
;*****
CONST:      EQU      1          ;ПРЯМОЕ БПФ
SIZE:       EQU      256       ;РАЗМЕР ОБР. ПОСЛ-ТИ N
SIZE2:      EQU      512       ;N*2
DIS1:       EQU      128       ;N/2
DIS2:       EQU      64        ;N/4
BEGINSIGNAL: EQU      4000H     ;НАЧ. АДР. ОБР. ПОСЛ-ТИ
BEGINCOEFF: EQU      0780H     ;НАЧ. АДР. ТАБЛ. КОЭФ. W(K),W(ZK)
INFVEC:     EQU      0760H     ;НАЧ. АДР. ИНФОРМ. ВЕКТОРА
ADR1:       EQU      BEGINSIGNAL+SIZE2;АДР. КОЛ-ВА ОБР. БЛОК.
ADR2:       EQU      ADR1+2    ;АДР. КОЛ-ВА БЛОКОВ НА ЭТАПЕ
ADR3:       EQU      ADR2+2    ;АДР. КОЛ-ВА ПАР В БЛОКЕ
ADR4:       EQU      ADR3+2    ;АДР. КОЛ-ВА ОБР. ПАР В БЛОКЕ
ADR5:       EQU      ADR4+2    ;АДР. СМЕЩЕНИЯ ДЛЯ КОЭФ.
ADR9:       EQU      ADR5+2    ;АДР. ОПЕРАНДА А
ADR10:      EQU      ADR9+2    ;АДР. ОПЕРАНДА В
ADR11:      EQU      ADR10+2   ;АДР. ОПЕРАНДА W
ADR12:      EQU      ADR11+2   ;АДР. ВСПОМ. ЯЧЕЙКИ ПАМЯТИ
ADR13:      EQU      ADR12+1   ;АДР. ВСПОМ. ЯЧЕЙКИ ПАМЯТИ
ADR14:      EQU      ADR13+1   ;АДР. СЧЕТЧИКА МАСШТАБИР.
ADR15:      EQU      ADR14+1   ;АДР. КОЛ-ВА ОДИНАК. ПАР В БЛОКЕ
ADR16:      EQU      ADR15+2   ;АДР. КОЛ-ВА ОБР. ОДИНАК. ПАР
ADR17:      EQU      ADR16+2   ;АДР. СЧЕТЧИКА БИТОВ В БАЙТЕ
ADR18:      EQU      ADR17+1   ;АДР. ТЕКУЩЕГО БАЙТА INFVEC
ADR19:      EQU      ADR18+2   ;АДР. ОБР. БАЙТА INFVEC

;*****
;*   FFTROT- ПРОГРАММА БПФО (ПРОРЕЖИВАНИЕ ПО ВРЕМЕНИ) *
;*****

FFTROT:     MVI      A,1       ;ЗАГР.НАЧ.ЗНАЧЕНИЯ СЧЕТЧИКА
            STA      ADR14     ;МАСШТАБИРОВАНИЯ
            LXI     H,DIS1     ;НАЧ. КОЛ-ВО БЛОКОВ НА ЭТАПЕ
            SHLD   ADR1
            SHLD   ADR5       ;НАЧ.СМЕЩЕНИЕ ДЛЯ КОЭФ. W
            LXI     H,1        ;НАЧ.КОЛ-ВО ОДИН. ПАР В
            SHLD   ADR15     ;БЛОКЕ "0"
            DAD    H          ;НАЧ.КОЛ-ВО И СМЕЩЕНИЕ ПАР В
            SHLD   ADR3      ;БЛОКЕ
            CALL   SCALING   ;МАСШТАБИРОВАНИЕ
            LXI     H,BEGINSIGNAL
            DCX    H
            DCX    H
            SHLD   ADR9
CYCLT:     LXI     D,2        ;ВЫПОЛНЕНИЕ ВЫЧИСЛЕНИЯ НА ПЕРВОМ
            LHL D,ADR9       ;ЭТАПЕ С КОЭФ. W=1
            DAD    D
            SHLD   ADR9
            DAD    D
            SHLD   ADR10
            CALL   BUT1
            LHL D,ADR10
            SHLD   ADR9

```

```

LHLD   ADR1   ;ПРОВЕРКА КОЛ-ВА ОБР. БЛОКОВ
DCX    H
SHLD   ADR1
MOV    A,H
ORA    L
JNZ    CYCLT
LXI    H,DIS2
SHLD   ADR1
SHLD   ADR2
ITERROT: MVI    A,9   ;НАЧ.ЗНАЧЕНИЕ СЧЕТЧИКА ОБР.БИТОВ
          STA   ADR17 ;В БАЙТЕ
          LXI   H,INFVEC;НАЧ.ЗНАЧЕНИЕ АДРЕСА БАЙТА INFVEC
          SHLD  ADR18
          MOV   A,M
          STA   ADR19
          CALL  SCALING ;МАСШТАБИРОВАНИЕ
          LXI   H,BEGINSIGNAL
          DCX   H
          DCX   H
          SHLD  ADR9
CONTRT: LDA   ADR17 ;ОПРЕДЕЛЕНИЕ ВИДА ОБР. БЛОКА
          DCR   A
          JNZ   CONT
          LHLD  ADR18
          INX   H
          SHLD  ADR18
          MOV   A,M
          STA   ADR19
          MVI   A,B
          STA   ADR17
          LDA   ADR19
          RLC
          STA   ADR19
          JC    BLOCK1T
          LHLD  ADR15
          SHLD  ADR16
BLOCKOOT: LHLD  ADR3   ;ВЫПОЛНЕНИЕ ВЫЧИСЛЕНИИ В БЛОКЕ "0"
          DAD   H      ;С КОЭФ. W=1
          KCHG
          LHLD  ADR9
          INX   H
          INX   H
          SHLD  ADR9
          DAD   D
          SHLD  ADR10
          CALL  BUT1
          LHLD  ADR16 ;ПРОВЕРКА КОЛ-ВА ОБР. ПАР
          DCX   H
          SHLD  ADR16
          MOV   A,H
          ORA   L
          JNZ   BLOCKOOT
          LHLD  ADR15
          SHLD  ADR16
BLOCKO1T: LHLD  ADR3   ;ВЫПОЛНЕНИЕ ВЫЧИСЛЕНИИ В БЛОКЕ "0"
          DAD   H      ;С КОЭФ. W=J
          KCHG
          LHLD  ADR9
          INX   H

```

```

      INX      H
      SHLD    ADR9
      DAD     D
      SHLD    ADR10
      CALL    BUT2T
      LHL    ADR16      ;ПРОВЕРКА КОЛ-ВА ОБР. ПАР
      DCX     H
      SHLD    ADR16
      MOV     A,H
      ORA     L
      JNZ     BLOCK01T
NEXTT:  LHL    ADR1      ;ПРОВЕРКА КОЛ-ВА ОБР. БЛОКОВ
      DCX     H
      SHLD    ADR1
      MOV     A,H
      ORA     L
      JZ      EXT
      LHL    ADR10
      SHLD    ADR9
      JMP     CONTRT
BLOCK1T: LHL    ADR3      ;ВЫПОЛНЕНИЕ ВЫЧИСЛЕНИЯ В БЛОКЕ "1"
      DAD     H          ;С КОЭФ. W=1
      XCHG
      LHL    ADR9
      INX     H
      INX     H
      SHLD    ADR9
      DAD     D
      SHLD    ADR10
      CALL    BUT1
      LHL    ADR3
      DCX     H
      SHLD    ADR4
      LXI     H,BEGINCOEFF-4
      SHLD    ADR11
BLOCK11T: LHL    ADR5      ;ВЫПОЛНЕНИЕ ВЫЧИСЛЕНИЯ В БЛОКЕ "1"
      XCHG    ;С ПРОИЗВОЛЬНЫМИ КОЭФ. W(K) И W(ZK)
      LHL    ADR11
      DAD     D
      SHLD    ADR11
      LHL    ADR3
      DAD     H
      XCHG
      LHL    ADR9
      INX     H
      INX     H
      SHLD    ADR9
      DAD     D
      SHLD    ADR10
      CALL    BUTROT
      LHL    ADR4      ;ПРОВЕРКА КОЛ-ВА ОБР. ПАР
      DCX     H
      SHLD    ADR4
      MOV     A,H
      ORA     L
      JNZ     BLOCK11T
      JMP     NEXTT
EXT:   LHL    ADR2      ;ОПР-Е СМЕЩЕНИЯ И КОЛ-ВА ОБР.
      SHLD    ADR5      ;ПАР НА СЛЕД. ЭТАПЕ

```

```

MOV     A,H
RAR
MOV     H,A
MOV     A,L
RAR
MOV     L,A
RC
SHLD   ADR2
SHLD   ADR1
LHLD   ADR3
SHLD   ADR15
DAD    H
SHLD   ADR3
JMP    ITERROT

```

```

;*****
;* BUTROT- ПОДПРОГРАММА "БАБОЧКА", W(K) И W(ZK)- ПРОИЗВОЛНЫЕ *
;*****
;
; REIA' ]=( REIA] * REIW(K) ] + IMIA] * IMIW(K) ] ) +
;           + ( REIB] * REIW(ZK) ] + IMIB] * IMIW(ZK) ] )
; IMIA' ]=( IMIA] * REIW(K) ] - REIA] * IMIW(K) ] +
;           + ( IMIB] * REIW(ZK) ] - REIB] * IMIW(ZK) ] )
; REIB' ]=( REIA] * REIW(K) ] + IMIA] * IMIW(K) ] -
;           - ( REIB] * REIW(ZK) ] + IMIB] * IMIW(ZK) ] )
; IMIB' ]=( IMIA] * REIW(K) ] - REIA] * IMIW(K) ] -
;           - ( IMIB] * REIW(ZK) ] - REIB] * IMIW(ZK) ] )
; ОБНФ: IMIW(K) ] = -IMIW(K), IMIW(ZK) ] = -IMIW(ZK)
;-----

```

```

BUTROT:  LHLD   ADR9
          MOV    C,M      ;REIA]
          LHLD   ADR11
          MOV    E,M      ;REIW(K) ]
          CALL   MULTIPLICATION ;REIA] * REIW(K) ]
          LXI   H,ADR12
          MOV    M,B
          LHLD   ADR9
          INX   H
          MOV    C,M      ;IMIA]
          LHLD   ADR11
          INX   H
          MOV    E,M      ;IMIW(K) ]
          MVI   A,CONST
          CPI   01H
          JZ    MUL1T
          MOV    A,E
          CMA
          INR   A
          MOV    E,A
MUL1T:   CALL   MULTIPLICATION ;IMIA] * IMIW(K) ]
          LDA   ADR12
          ADD   B          ;REIA] * REIW(K) ] + IMIA] * IMI(K) ]
          STA   ADR12
          LHLD   ADR9
          INX   H
          MOV    C,M      ;IMIA]
          LHLD   ADR11
          MOV    E,M      ;REIW(K) ]
          CALL   MULTIPLICATION ;IMIA] * REIW(K) ]
          LXI   H,ADR13
          MOV    M,B

```

```

LHLD   ADR9
MOV    C,M      ;REIA]
LHLD   ADR11
INX    H
MOV    E,M      ;IMIW(K)]
MVI    A,CONST
CPI    01H
JZ     MUL2T
MOV    A,E
CMA
INR    A
MOV    E,A
MUL2T: CALL MULTIPLICATION ;REIA]*IMIW(K)]
LDA    ADR12
LHLD   ADR9
MOV    M,A
LDA    ADR13
SUB    B        ;IMIA]*REIW(K)]-REIA]*IMIW(K)]
INX    H
MOV    M,A
LHLD   ADR10
MOV    C,M      ;REIB]
LHLD   ADR11
INX    H
INX    H
MOV    E,M      ;REIW(3K)]
CALL   MULTIPLICATION ;REIB]*REIW(3K)]
LXI    H,ADR12
MOV    M,B
LHLD   ADR10
INX    H
MOV    C,M      ;IMIB]
LHLD   ADR11
INX    H
INX    H
INX    H
MOV    E,M      ;IMIW(3K)]
MVI    A,CONST
CPI    01H
JZ     MUL3T
MOV    A,E
CMA
INR    A
MOV    E,A
MUL3T: CALL MULTIPLICATION ;IMIB]*IMIW(3K)]
LDA    ADR12
ADD    B        ;REIB]*REIW(3K)]+IMIB]*IMIW(3K)]
STA    ADR12
LHLD   ADR10
INX    H
MOV    C,M      ;IMIB]
LHLD   ADR11
INX    H
INX    H
MOV    E,M      ;REIW(3K)]
CALL   MULTIPLICATION ;IMIB]*REIW(3K)]
LXI    H,ADR13
MOV    M,B
LHLD   ADR10

```

```

MOV      C,M      ;REIB]
LHLD    ADR11
INX     H
INX     H
INX     H
MOV     E,M      ;IMIW(ЗК)]
MVI     A,CONST
CPI     01H
JZ      MUL4T
MOV     A,E
CMA
INR     A
MOV     E,A
MUL4T:  CALL     MULTIPLICATION ;REIB]*IMIW(ЗК)]
LDA     ADR13
SUB     B        ;IMIB]*REIW(ЗК)]-REIB]*IMIW(ЗК)]
MOV     B,A
LXI     H,ADR12
MOV     D,M      ;(REIB]*REIW(ЗК)]+IMIB]*I[W(ЗК)]
LHLD    ADR9
MOV     A,M      ;(REIA]*REIW(K)]+IMIA]*IMIW(K)]
MOV     E,A
ADD     D        ;(REIA]*REIW(K)]+IMIA]*IMIW(K)]+
MOV     M,A      ;+(REIB]*REIW(ЗК)]+IMIB]*IMIW(ЗК)]
INX     H
MOV     A,M      ;(IMIA]*REIW(K)]-REIA]*IMIW(K)]
MOV     C,A
ADD     B        ;(IMIA]*REIW(K)]-REIA]*IMIW(K)]+
MOV     M,A      ;+(IMIB]*REIW(ЗК)]-REIB]*IMIW(ЗК)]
LHLD    ADR10
MOV     A,E
SUB     D        ;(REIA]*REIW(K)]+IMIA]*IMIW(K)]-
MOV     M,A      ;-(REIB]*REIW(ЗК)]+IMIB]*IMIW(ЗК)]
INX     H
MOV     A,C
SUB     B        ;(IMIA]*REIW(K)]-REIA]*IMIW(K)]-
MOV     M,A      ;-(IMIB]*REIW(ЗК)]-REIB]*IMIW(ЗК)]
RET

```

```

*****
;*      FFTROF- ПРОГРАММА БПФРО (ПРОРЕЖИВАНИЕ ПО ЧАСТОТЕ)      *
*****

```

```

FFTROF:  MVI     A,1      ;ЗАГР.НАЧ.ЗНАЧЕНИЯ СЧЕТЧИКА
          STA     ADR14   ;МАСШТАБИРОВАНИЯ
          LXI     H,DIS1  ;НАЧ.СМЕЩЕНИЕ И КОЛ-ВО ОБР.
          SHLD    ADR3    ;ПАР В БЛОКЕ
          LXI     H,DIS2  ;НАЧ.КОЛ-ВО ОДИНАК. ПАР В
          SHLD    ADR15   ;БЛОКЕ "0"
          LXI     H,1     ;НАЧ.КОЛ-ВО БЛОКОВ НА ЭТАПЕ
          SHLD    ADR1
          SHLD    ADR2
          LXI     H,2     ;НАЧ.СМЕЩЕНИЕ ДЛЯ КОЭФ.
ITERROF: SHLD    ADR5
          MVI     A,9     ;НАЧ.ЗНАЧЕНИЕ СЧЕТЧИКА ОБР.БИТОВ
          STA     ADR17   ;В БАЙТЕ
          LXI     H,INFVEC;НАЧ.ЗНАЧЕНИЕ АДРЕСА БАЙТА INFVEC
          SHLD    ADR18
          MOV     A,M

```

```

STA      ADR19
CALL     SCALING ;МАСШТАБИРОВАНИЕ
LXI      H,BEGINSIGNAL
DCX      H
DCX      H
SHLD     ADR9
CONTRF:  LDA      ADR17      ;ОПРЕДЕЛЕНИЕ ВИДА ОБР. БЛОКА
DCR      A
JNZ      CONF
LHLD     ADR18
INX      H
SHLD     ADR18
MOV      A,M
STA      ADR19
MVI      A,8
CONF:    STA      ADR17
LDA      ADR19
RLC
STA      ADR19
JC       BLOCK1F
LHLD     ADR15
SHLD     ADR16
BLOCKOOF: LHLD     ADR3      ;ВЫПОЛНЕНИЕ ВЫЧИСЛЕНИИ В БЛОКЕ "0"
DAD      H      ;С КОЭФ. W=1
XCHG
LHLD     ADR9
INX      H
INX      H
SHLD     ADR9
DAD      D
SHLD     ADR10
CALL     BUT1
LHLD     ADR16      ;ПРОВЕРКА КОЛ-ВА ОБР. ПАР
DCX      H
SHLD     ADR16
MOV      A,H
ORA      L
JNZ      BLOCKOOF
LHLD     ADR15
SHLD     ADR16
BLOCKO1F: LHLD     ADR3      ;ВЫПОЛНЕНИЕ ВЫЧИСЛЕНИИ В БЛОКЕ "0"
DAD      H      ;С КОЭФ. W=J
XCHG
LHLD     ADR9
INX      H
INX      H
SHLD     ADR9
DAD      D
SHLD     ADR10
CALL     BUT2F
LHLD     ADR16      ;ПРОВЕРКА КОЛ-ВА ОБР. ПАР
DCX      H
SHLD     ADR16
MOV      A,H
ORA      L
JNZ      BLOCKO1F
NEXTF:   LHLD     ADR1      ;ПРОВЕРКА КОЛ-ВА ОБР. БЛОКОВ
DCX      H
SHLD     ADR1

```



```

MOV      A,H
ORA      L
JZ       EXF
LHLD     ADR10
SHLD     ADR9
JMP      CONTRF
BLOCK1F: LHLD     ADR3      ;ВЫПОЛНЕНИЕ ВЫЧИСЛЕНИИ В БЛОКЕ "1"
DAD      H              ;С КОЭФ. W=1
XCHG
LHLD     ADR9
INX      H
INX      H
SHLD     ADR9
DAD      D
SHLD     ADR10
CALL     BUT1
LHLD     ADR3
DCX      H
SHLD     ADR4
LXI      H,BEGINCOEFF-4
SHLD     ADR11
BLOCK11F: LHLD     ADR5      ;ВЫПОЛНЕНИЕ ВЫЧИСЛЕНИИ В БЛОКЕ "1"
XCHG     ;С ПРОИЗВОЛЬНЫМИ КОЭФ. W(K) И W(ЗК)
LHLD     ADR11
DAD      D
SHLD     ADR11
LHLD     ADR3
DAD      H
XCHG
LHLD     ADR9
INX      H
INX      H
SHLD     ADR9
DAD      D
SHLD     ADR10
CALL     BUTROF
LHLD     ADR4      ;ПРОВЕРКА КОЛ-ВА ОБР. ПАР
DCX      H
SHLD     ADR4
MOV      A,H
ORA      L
JNZ      BLOCK11F
JMP      NEXTF
EXF:     LHLD     ADR2      ;ОП-Е СМЕЩЕНИИ И КОЛ-ВА ОБР.
DAD      H          ;ПАР НА СЛЕД. ЭТАПЕ
SHLD     ADR1
SHLD     ADR2
LHLD     ADR3
MOV      A,H
RAR
MOV      H,A
MOV      A,L
RAR
MOV      L,A
CPI      1
JZ       LASTST
SHLD     ADR3
MOV      A,H
RAR

```

```

MOV      H,A
MOV      A,L
RAR
MOV      L,A
SHLD    ADR15
LHLD    ADR5
DAD      H
JMP      ITERROF
LASTST:  CALL    SCALING ;ВЫПОЛНЕНИЕ ВЫЧИСЛЕНИИ НА ПОСЛЕДНЕМ
LXI      H,BEGINSIGNAL ;ЭТАПЕ С КОЭФ. W=1
DCX      H
DCX      H
SHLD    ADR9
CYCLF:   LXI      D,2
LHLD    ADR9
DAD      D
SHLD    ADR9
DAD      D
SHLD    ADR10
CALL    BUT1
LHLD    ADR10
SHLD    ADR9
LHLD    ADR1      ;ПРОВЕРКА КОЛ-ВА ОБР. БЛОКОВ
DCX      H
SHLD    ADR1
MOV      A,H
ORA      L
JNZ     CYCLF
RET

```

```

;*****
;* BUTROF- ПОДПРОГРАММА "БАБОЧКА", W(K) И W(ZK)- ПРОИЗВОЛЬНЫЕ *
;*****
; REIA' ]=( REIA ]+REIB ])*REIW(K) ]+( IMIA ]+IMIB ])*IMIW(K) ]
; IMIA' ]=( IMIA ]+IMIB ])*REIW(K) ]-( REIA ]+REIB ])*IMIW(K) ]
; REIB' ]=( REIA ]-REIB ])*REIW(ZK) ]+( IMIA ]-IMIB ])*IMIW(ZK) ]
; IMIB' ]=( IMIA ]-IMIB ])*REIW(ZK) ]-( REIA ]-REIB ])*IMIW(ZK) ]
; ОБПФ: IMIW(K) ]=-IMIW(K) ], IMIW(ZK) ]=-IMIW(ZK) ]
;-----

```

```

BUTROF:  MOV      B,M      ;REIB ]
INX      H
MOV      C,M      ;IMIB ]
LHLD    ADR9
MOV      A,M      ;REIA ]
MOV      D,A
ADD      B          ;REIA ]+REIB ]
MOV      M,A
INX      H
MOV      A,M      ;IMIA ]
MOV      E,A
ADD      C          ;IMIA ]+IMIB ]
MOV      M,A
LHLD    ADR10
MOV      A,D
SUB      B          ;REIA ]-REIB ]
MOV      M,A
INX      H
MOV      A,E
SUB      C          ;IMIA ]-IMIB ]
MOV      M,A

```

```

MOV      C,A
LHLD    ADR11
INX     H
INX     H
INX     H
MOV     E,M      ;IMIW(3K)]
MVI     A,CONST
CPI     01H
JZ      MUL1F
MOV     A,E
CMA
INR     A
MOV     E,A
MUL1F:  CALL    MULTIPLICATION ;(IM[A]-IM[B])*IMW(3K)]
LXI     H,ADR12
MOV     M,B
LHLD    ADR10
MOV     C,M      ;(RE[A]-RE[B])
LHLD    ADR11
INX     H
INX     H
MOV     E,M      ;REIW(3K)]
CALL    MULTIPLICATION ;(RE[A]-RE[B])*REIW(3K)]
LDA     ADR12
ADD     B      ;(RE[A]-RE[B])*REIW(3K)]+
STA     ADR12  ;+(IM[A]-IM[B])*IMW(3K)]
LHLD    ADR10
INX     H
MOV     C,M      ;(IM[A]-IM[B])
LHLD    ADR11
INX     H
INX     H
MOV     E,M      ;REIW(3K)]
CALL    MULTIPLICATION ;(IM[A]-IM[B])*REIW(3K)]
LXI     H,ADR13
MOV     M,B
LHLD    ADR10
MOV     C,M      ;(RE[A]-RE[B])
LHLD    ADR11
INX     H
INX     H
INX     H
MOV     E,M      ;IMIW(3K)]
MVI     A,CONST
CPI     01H
JZ      MUL2F
MOV     A,E
CMA
INR     A
MOV     E,A
MUL2F:  CALL    MULTIPLICATION ;(RE[A]-RE[B])*IMW(3K)]
LDA     ADR12
LHLD    ADR10
MOV     M,A
LDA     ADR13
SUB     B      ;(IM[A]-IM[B])*REIW(3K)]-
INX     H      ;-(RE[A]-RE[B])*IMW(3K)]
MOV     M,A
LHLD    ADR9

```

```

INX      H
MOV      C,M      ;( IM[A] + IM[B] )
LHLD    ADR11
INX      H
MOV      E,M      ; IM[W(K)]
MVI      A,CONST
CPI      01H
JZ       MUL3F
MOV      A,E
CMA
INR      A
MOV      E,A
MUL3F:  CALL     MULTIPLICATION ;( IM[A] + IM[B] ) * IM[W(K)]
LXI      H,ADR12
MOV      M,B
LHLD    ADR9
MOV      C,M      ;( RE[A] + RE[B] )
LHLD    ADR11
MOV      E,M      ; RE[W(K)]
CALL     MULTIPLICATION ;( RE[A] + RE[B] ) * RE[W(K)]
LDA      ADR12
ADD      B      ;( RE[A] + RE[B] ) * RE[W(K)] +
STA      ADR12  ;+( IM[A] + IM[B] ) * IM[W(K)]
LHLD    ADR9
INX      H
MOV      C,M      ;( IM[A] + IM[B] )
LHLD    ADR11
MOV      E,M      ; RE[W(K)]
CALL     MULTIPLICATION ;( IM[A] + IM[B] ) * RE[W(K)]
LXI      H,ADR13
MOV      M,B
LHLD    ADR9
MOV      C,M      ;( RE[A] + RE[B] )
LHLD    ADR11
INX      H
MOV      E,M      ; IM[W(K)]
MVI      A,CONST
CPI      01H
JZ       MUL4F
MOV      A,E
CMA
INR      A
MOV      E,A
MUL4F:  CALL     MULTIPLICATION ;( RE[A] + RE[B] ) * IM[W(K)]
LDA      ADR12
LHLD    ADR9
MOV      M,A
LDA      ADR13
SUB      B      ;( IM[A] + IM[B] ) * RE[W(K)] -
INX      H      ;-( RE[A] + RE[B] ) * IM[W(K)]
MOV      M,A
RET

```

```

;*****
;* ПРОГРАММЫ БЫСТРОГО ПРЕОБРАЗОВАНИЯ КАРТЛИ С РАСЩЕПЛЕННЫМ *
;* ОСНОВАНИЕМ *
;*****

```

```

SIZE:      EQU      256      ;РАЗМЕР ОБРАБАТ. ПОСЛЕД-ТИ N
DIS1:      EQU      128      ;N/2
DIS2:      EQU      64       ;N/4
BEGNSIGNAL: EQU      4000H    ;НАЧ. АДР. ПОСЛЕД-ТИ
BEGINCOEFF: EQU      0780H    ;НАЧ. АДР. ТАБЛ. КОЭФФИЦ.
INFVEC:    EQU      0760H    ;НАЧ. АДР. ИНФОРМАЦ. ВЕКТОРА
ADR1:      EQU      BEGNSIGNAL+SIZE ;АДР. КОЛ-ВА БЛОКОВ
ADR2:      EQU      ADR1+2    ;АДР.КОЛ-ВА ОБР. БЛОКОВ
ADR3:      EQU      ADR2+2    ;АДР.КОЛ-ВА ПАР В БЛОКЕ
ADR4:      EQU      ADR3+2    ;АДР.КОЛ-ВА ОБР. ПАР В БЛОКЕ
ADR5:      EQU      ADR4+2    ;АДР.СМЕЩЕНИЯ ДЛЯ КОЭФФИЦ.
ADR6:      EQU      ADR5+2    ;АДР.СМЕЩЕНИЯ ДЛЯ КОЭФФИЦ.+2
ADR8:      EQU      ADR6+2    ;АДР.НАЧ.АДРЕСА ТЕК. БЛОКА
ADR9:      EQU      ADR8+2    ;АДР. ОПЕРАНДА В
ADR10:     EQU      ADR9+2    ;АДР. ОПЕРАНДА А
ADR11:     EQU      ADR10+2   ;АДР. ОПЕРАНДА W
ADR12:     EQU      ADR11+2   ;АДР. ВСПОМ. ЯЧЕЙКИ ПАМЯТИ
ADR13:     EQU      ADR12+1   ;АДР. ВСПОМ. ЯЧЕЙКИ ПАМЯТИ
ADR14:     EQU      ADR13+1   ;АДР. СЧЕТЧИКА МАСШТАБИРОВАНИЯ
ADR15:     EQU      ADR14+1   ;АДР. КОЛ-ВА ОДИНАК. ПАР В БЛОКЕ
ADR16:     EQU      ADR15+2   ;АДР. КОЛ-ВА ОБР. ОДИНАК. ПАР
ADR17:     EQU      ADR16+2   ;АДР. СЧЕТЧИКА БИТОВ В БАЙТЕ
ADR18:     EQU      ADR17+1   ;АДР. ТЕКУЩЕГО БАЙТА INFVEC
ADR19:     EQU      ADR18+2   ;АДР. ОБР. БАЙТА INFVEC
ADR20:     EQU      ADR19+1   ;АДР. ОПЕРАНДОВ ПАР, ОБРАБАТЫ-
ADR21:     EQU      ADR20+2   ;ВАЕМЬК СОВМЕСТНО В БЛОКЕ "1"
ADR22:     EQU      ADR21+2
ADR23:     EQU      ADR22+2

```

```

;*****
;* FHTROT- ПРОГРАММА БПХРО (ПРОРЕЖИВАНИЕ ПО ВРЕМЕНИ) *
;*****

```

```

FHTROT:    MVI      A,1      ;НАЧ.ЗАГР.СЧЕТЧИКА МАСШТАБИР.
           STA      ADR14
           LXI      H,DIS1   ;НАЧ. КОЛ-ВО БЛОКОВ НА ЭТАПЕ
           SHLD    ADR2
           SHLD    ADR6
           LXI      H,1      ;НАЧ. КОЛ-ВО СОВМЕСТНО ОБР. ГРУПП
           SHLD    ADR15    ;ОПЕРАНДОВ
           DAD     H        ;НАЧ.СМЕЩЕНИЕ И КОЛ-ВО ПАР В БЛОКЕ
           SHLD    ADR3
           CALL    SCALING  ;ВЫПОЛНЕНИЕ ВЫЧИСЛЕНИЯ НА ПЕРВОМ
           LXI      D,BEGNSIGNAL ;ЭТАПЕ
           LXI      H,BEGNSIGNAL+1
CYCLHT:    LDA     D
           MOV     B,A
           ADD     M
           STAX   D
           MOV     A,B
           SUB    M
           MOV     M,A

```

```

INX      D
INX      D
INX      H
INX      H
PUSH     H
LHLD     ADR2      ;ПРОВЕРКА КОЛ-ВА ОБР. БЛОКОВ
DCX      H
SHLD     ADR2
MOV      A,H
ORA      L
POP      H
JNZ      CYCLHT
LXI      H,DIS2
SHLD     ADR1
SHLD     ADR2
ITEROHT: MVI      A,9      ;НАЧ.ЗНАЧЕНИЕ СЧЕТЧИКА ОБР.БИТОВ
          STA      ADR17   ;В БАЙТЕ
          LXI      H,INFVEC;НАЧ.ЗНАЧЕНИЕ АДРЕСА БАЙТА INFVEC
          SHLD     ADR18
          MOV      A,M
          STA      ADR19
          CALL     SCALING ;МАСШТАБИРОВАНИЕ
          LXI      H,BEGINSIGNAL
          SHLD     ADR8
CONTRHT: LDA      ADR17   ;ОПРЕДЕЛЕНИЕ ВИДА БЛОКА
          DCR      A
          JNZ      CONHT
          LHLD     ADR18
          INX      H
          SHLD     ADR18
          MOV      A,M
          STA      ADR19
CONHT:   MVI      A,8
          STA      ADR17
          LDA      ADR19
          RLC
          STA      ADR19
          JC      BLOC1HT
          LHLD     ADR3
          SHLD     ADR4
          LHLD     ADR8
          KCHG
          LHLD     ADR3
          DAD      D
          LDAX     D      ;ВЫПОЛНЕНИЕ ВЫЧИСЛЕНИЯ В БЛОКЕ "0"
          MOV      B,A
          ADD      M
          STAX     D
          MOV      A,B
          SUB      M
          MOV      M,A
          INX      D
          INX      H
          PUSH     H
          LHLD     ADR4   ;ПРОВЕРКА КОЛ-ВА ОБР. ПАР
          DCX      H      ;В БЛОКЕ "0"
          SHLD     ADR4
          MOV      A,H
          ORA      L

```

	POP	H	
	JNZ	BLOCOHT	
PRENEXTT:	SHLD	ADRS	;ЗАПОМ. НАЧ. АДРЕСА СЛЕД. БЛОКА
NEXTHT:	LHLD	ADR2	;ПРОВ-КА КОЛ-ВА ОБР. БЛОКОВ
	DCX	H	
	SHLD	ADR2	
	MOV	A,H	
	ORA	L	
	JZ	EXHT	
	JMP	CONTRHT	
BLOC1HT:	LHLD	ADRS	;ВЫПОЛНЕНИЕ ВЫЧИСЛЕНИЯ В БЛОКЕ "1"
	KCHG		
	LHLD	ADR3	
	DAD	D	
	LDAX	D	
	MOV	B,A	
	ADD	M	
	STAX	D	
	MOV	A,B	
	SUB	M	
	MOV	M,A	
	LHLD	ADR15	
	DAD	D	
	KCHG		
	PUSH	D	
	LHLD	ADR3	
	DAD	D	
	PUSH	H	
	MOV	C,M	
	MVI	E,5BH	
	CALL	MULTIPLICATION	
	MOV	A,B	
	ADD	A	
	POP	H	
	MOV	M,A	
	POP	H	
	PUSH	H	
	MOV	C,M	
	MVI	E,5BH	
	CALL	MULTIPLICATION	
	MOV	A,B	
	ADD	A	
	POP	H	
	MOV	M,A	
	LHLD	ADR15	
	DCX	H	
	SHLD	ADR16	
	MOV	A,H	
	ORA	L	
	JNZ	ALONGT	
	LHLD	ADRS	
	KCHG		
	LHLD	ADR3	
	DAD	H	
	DAD	D	
	JMP	PRENEXTT	
ALONGT:	LHLD	ADR3	;ОП-Е АДР. СОВМЕСТНО ОБР. ОПЕРАНДОВ
	KCHG		
	LHLD	ADRS	

```

SHLD  ADR20
DAD   D
SHLD  ADR21
SHLD  ADR22
DAD   D
SHLD  ADR23
SHLD  ADR8 ;ЗАПОМ. НАЧ.АДР. СЛЕД. БЛОКА
LXI   H,BEGINCOEFF-2
SHLD  ADR11
BLOC11HT:
LHLD  ADR20 ;ВЫПОЛНЕНИЕ ПОДЭТАПА УМНОЖЕНИИ
INX   H
SHLD  ADR20
SHLD  ADR10
LHLD  ADR21
DCX   H
SHLD  ADR21
SHLD  ADR9
LHLD  ADR11
KCHG
LHLD  ADR5
DAD   D
SHLD  ADR11
CALL  BUTH
LHLD  ADR22
INX   H
SHLD  ADR22
SHLD  ADR10
LHLD  ADR23
DCX   H
SHLD  ADR23
SHLD  ADR9
LHLD  ADR11
INX   H
INX   H
SHLD  ADR11
CALL  BUTH
LHLD  ADR20 ;ВЫПОЛНЕНИЕ ПОДЭТАПА СЛОЖЕНИИ-
KCHG ;ВЫЧИТАНИИ
LHLD  ADR22
LDAX  D
MOV   B,A
ADD   M
STAX  D
MOV   A,B
SUB   M
MOV   C,A
LHLD  ADR23
KCHG
LHLD  ADR21
LDAX  D
MOV   B,A
ADD   M
STAX  D
MOV   A,B
SUB   M
MOV   M,C
LHLD  ADR22
MOV   M,B
LHLD  ADR16 ;ПРОВЕРКА КОЛ-ВА ОПР. ПАР

```



```

DCX      H
SHLD     ADR16
MOV      A,H
ORA      L
JNZ      BLOC11HT
JMP      NEXTHT
EKHT:    LHL     ADR1      ;ОПР-Е СМЕЩЕНИИ И КОЛ-ВА ОБР.
MOV      A,H      ;ПАР НА СЛЕД. ЭТАПЕ
RAR
MOV      H,A
MOV      A,L
RAR
MOV      L,A
RC
SHLD     ADR1
SHLD     ADR2
LHL     ADR3
SHLD     ADR15
DAD      H
SHLD     ADR3
LHL     ADR6
MOV      A,H
RAR
MOV      H,A
MOV      A,L
RAR
MOV      L,A
SHLD     ADR6
DCX      H
DCX      H
SHLD     ADR5
JMP      ITERROHT

;*****
;*      FHTROF- ПРОГРАММА БИКРО (ПРОРЕЖИВАНИЕ ПО ЧАСТОТЕ)      *
;*****

FHTROF:  MVI      A,1      ;НАЧ.ЗАГР.СЧЕТЧИКА МАСШТАБИР.
STA      ADR14
LXI      H,DIS1      ;НАЧ.СМЕЩЕНИЕ И КОЛ-ВО ОБР. ПАР
SHLD     ADR3      ;В БЛОКЕ "0"
LXI      H,1      ;НАЧ. КОЛ-ВО БЛОКОВ НА ЭТАПЕ
SHLD     ADR1
SHLD     ADR2
DAD      H
SHLD     ADR6

ITERROHF: MVI      A,9      ;НАЧ.ЗНАЧЕНИЕ СЧЕТЧИКА ОБР.БИТОВ
STA      ADR17      ;В БАЙТЕ
LXI      H,INFVEC;НАЧ.ЗНАЧЕНИЕ АДРЕСА БАЙТА INFVEC
SHLD     ADR18
MOV      A,M
STA      ADR19
CALL     SCALING ;МАСШТАБИРОВАНИЕ
LXI      H,BEGINSIGNAL
SHLD     ADR8

CONTRHF: LDA      ADR17      ;ОПРЕДЕЛЕНИЕ ВИДА БЛОКА
DCR      A
JNZ      CONHF
LHL     ADR18

```

```

      INX      H
      SHLD    ADR18
      MOV     A,M
      STA     ADR19
      MVI     A,B
CONHF:  STA     ADR17
      LDA     ADR19
      RLC
      STA     ADR19
      JC      BLOC1HF
      LHLD    ADR3
      SHLD    ADR4
      LHLD    ADR8
      XCHG
      LHLD    ADR3
      DAD     D
BLOC0HF: LDAX   D      ;ВЫПОЛНЕНИЕ ВЫЧИСЛЕНИЯ В БЛОКЕ "0"
      MOV     B,A
      ADD     M
      STAX   D
      MOV     A,B
      SUB     M
      MOV     M,A
      INX     D
      INX     H
      PUSH   H
      LHLD    ADR4      ;ПРОВЕРКА КОЛ-ВА ОБР. ПАР
      DCX     H      ;В БЛОКЕ "0"
      SHLD    ADR4
      MOV     A,H
      ORA     L
      POP     H
      JNZ     BLOC0HF
PRENEKTF: SHLD   ADR8      ;ЗАПОМ. НАЧ. АДРЕСА СЛЕД. БЛОКА
NEKTHF:  LHLD   ADR2      ;ПРОВ-КА КОЛ-ВА ОБР. БЛОКОВ
      DCX     H
      SHLD    ADR2
      MOV     A,H
      ORA     L
      JZ      EXHF
      JMP     CONTRHF
BLOC1HF: LHLD   ADR8      ;ВЫПОЛНЕНИЕ ВЫЧИСЛЕНИЯ В БЛОКЕ "1"
      XCHG
      LHLD    ADR3
      DAD     D
      LDAX   D
      MOV     B,A
      ADD     M
      STAX   D
      MOV     A,B
      SUB     M
      MOV     M,A
      LHLD    ADR15
      DAD     D
      XCHG
      PUSH   D
      LHLD    ADR3
      DAD     D
      PUSH   H

```

```

MOV      C,M
MVI      E,5BH
CALL     MULTIPLICATION
MOV      A,B
ADD      A
POP      H
MOV      M,A
POP      H
PUSH     H
MOV      C,M
MVI      E,5BH
CALL     MULTIPLICATION
MOV      A,B
ADD      A
POP      H
MOV      M,A
LHLD     ADR15
DCX      H
SHLD     ADR16
MOV      A,H
ORA      L
JNZ      ALONGF
LHLD     ADR8
XCHG
LHLD     ADR3
DAD      H
DAD      D
JMP      PRENEXTF
ALONGF:  LHLD     ADR3      ;ОПР-Е АДР. СОВМЕСТНО ОБР. ОПЕРАНДОВ
XCHG
LHLD     ADR8
SHLD     ADR20
DAD      D
SHLD     ADR21
SHLD     ADR22
DAD      D
SHLD     ADR23
SHLD     ADR8      ;ЗАПОМ. НАЧ.АДР. СЛЕД. БЛОКА
LXI      H,BEGINCOEFF-2
SHLD     ADR11
BLOC11HF: LHLD     ADR20      ;ВЫПОЛНЕНИЕ ПОДЭТАПА СЛОЖЕНИЯ-
INX      H          ;ВЫЧИТАНИЯ
SHLD     ADR20
XCHG
LHLD     ADR21
DCX      H
SHLD     ADR21
LDAX     D
MOV      B,A
ADD      M
STAX     D
MOV      A,B
SUB      M
MOV      C,A
LHLD     ADR23
DCX      H
SHLD     ADR23
XCHG
LHLD     ADR22

```

```

INX      H
SHLD    ADR22
LDAX    D
MOV     B,A
ADD     M
STAX    D
MOV     A,B
SUB     M
MOV     M,C
LHLD    ADR21
MOV     M,A
SHLD    ADR9      ;ВЫПОЛНЕНИЕ ПОДЭТАПА УМНОЖЕНИЯ
LHLD    ADR20
SHLD    ADR10
LHLD    ADR11
KCHG
LHLD    ADR5
DAD     D
SHLD    ADR11
CALL    BUTH
LHLD    ADR22
SHLD    ADR10
LHLD    ADR23
SHLD    ADR9
LHLD    ADR11
INX     H
INX     H
SHLD    ADR11
CALL    BUTH
LHLD    ADR16     ;ПРОВЕРКА КОЛ-ВА ОБР. ПАР
DCX     H
SHLD    ADR16
MOV     A,H
ORA     L
JNZ     BLOC11HF
JMP     NEXTHF
EXHF:   LHLD    ADR1      ;ОПР-Е СМЕЩЕНИИ И КОЛ-ВА ОБР.
DAD     H          ;ПАР НА СЛЕД. ЭТАПЕ
SHLD    ADR1
SHLD    ADR2
LHLD    ADR3
MOV     A,H
RAR
MOV     H,A
MOV     A,L
RAR
MOV     L,A
CPI     1
JZ      LASTHF
SHLD    ADR3
MOV     A,H
RAR
MOV     H,A
MOV     A,L
RAR
MOV     L,A
SHLD    ADR15
LHLD    ADR6
DAD     H

```

	SHLD	ADR6	
	DCX	H	
	DCX	H	
	SHLD	ADR5	
	JMP	ITERROHF	
LASTHF:	CALL	SCALING ;ВЫПОЛНЕНИЕ ВЫЧИСЛЕНИЯ НА ПОСЛЕДНЕМ	
	LXI	D,BEGINSIGNAL ;ЭТАПЕ	
	LXI	H,BEGINSIGNAL+1	
CYCLHF:	LDAX	D	
	MOV	B,A	
	ADD	M	
	STAX	D	
	MOV	A,B	
	SUB	M	
	MOV	M,A	
	INX	D	
	INX	D	
	INX	H	
	INX	H	
	PUSH	H	
	LHLD	ADR2 ;ПРОВЕРКА КОЛ-ВА ОБР. БЛОКОВ	
	DCX	H	
	SHLD	ADR2	
	MOV	A,H	
	ORA	L	
	POP	H	
	JNZ	CYCLHF	
	RET		

```

*****
;*      FFTR- ПРОГРАММА-ДОПОЛНЕНИЕ К ПРОГРАММЕ FFT1      *
;*      ДЛЯ БФ ВЕЩЕСТВЕННОГО СИГНАЛА                    *
*****
;      ДЛЯ ВЫПОЛНЕНИЯ БФ ВЕЩЕСТВЕННОГО СИГНАЛА В ПРОГРАММАХ
;      FFT1 И SCALING СИМВОЛ. ИМЯ BEGINSIGNAL НЕОБХОДИМО
;      ЗАМЕНИТЬ НА BEGINSIGNAL+SIZE-DIS2, SIZE2 - НА SIZE,
;      DIS1 - НА DIS2
;
-----

```

```

FFTR:      CALL      SCALING
           LXI      H,BEGINSIGNAL ;НАЧ. АДР. А(K)
           XCHG
           LXI      H,BEGINSIGNAL+SIZE-DIS2 ;НАЧ. АДР. Z(K)
           MOV      A,M          ;RE[Z(0)]=RE[X(0)]
           MOV      B,A
           INX      H
           MOV      C,M          ;IM[Z(0)]=RE[Y(0)]
           ADD      C          ;RE[X(0)]+RE[Y(0)]=RE[A(0)]
           STAX     D
           INX      D
           XRA      A          ;IM[A(0)]=0
           STAX     D
           MOV      A,B
           SUB      C          ;RE[X(0)]-RE[Y(0)]=RE[A(N/2)]
           INX      D
           STAX     D
           INX      D
           XRA      A          ;IM[A(N/2)]=0
           STAX     D
           INX      H
           MOV      A,M          ;RE[Z(N/2)]=RE[X(N/2)]=
           MOV      B,A          ;=RE[A(N/4)]
           INX      D
           STAX     D
           INX      D
           INX      H
           MOV      A,M          ;IM[Z(N/2)]=RE[Y(N/2)]=
           MOV      C,A          ;=-IM[A(N/4)]
           CMA
           INR      A
           STAX     D
           INX      D
           MOV      A,B          ;RE[Z(N/2)]=RE[X(N/2)]=
           STAX     D          ;=RE[A(N/4+N/2)]
           INX      D
           MOV      A,C          ;RE[Y(N/2)]=IM[A(N/4+N/2)]
           STAX     D
           INX      D
           INX      H
           SHLD     ADR1        ;ТЕК. АДР. Z(K)
           XCHG
           SHLD     ADR9        ;ТЕК. АДР. А(K)
           LXI      H,DIS2     ;ОПР-Е КОНЦА ВЫЧИСЛЕНИЯ
           XRA      A
           MOV      A,H

```

```

RAR
MOV      H,A
MOV      A,L
RAR
MOV      L,A
RC
SHLD     ADR4
LXI      H,BEGINCOEFF ;НАЧ. АДР. ТАБЛ. КОЭФФИЦ.
SHLD     ADR11
LXI      H,1
SHLD     ADR3
BLOCKR:  LHLD     ADR3      ;ОПР-Е СМЕЩЕНИЯ В ОБЛ. Z(K)
DAD      H
SHLD     ADR3
DCX      H
DAD      H
XCHG
LHLD     ADR1      ;ОПР-Е ТЕК. АДР. Z(N-K)
DAD      D
SHLD     ADR2
LHLD     ADR9      ;ОПР-Е СМЕЩЕНИЯ В ОБЛ. A(K)
XCHG     ;И ТЕК. АДР. A(N-K)
DAD      H
INX      H
INX      H
DAD      D
SHLD     ADR10
CON:     LHLD     ADR2
MOV      D,M      ;RE(N-K)
INX      H
MOV      E,M      ;IM(N-K)
LHLD     ADR1
MOV      B,M      ;RE(K)
INX      H
MOV      C,M      ;IM(K)
MOV      A,C
SUB      E      ;(IM[Z(K)]-IM[Z(N-K)])/2=
RLC      ;=IM[X(K)]
RAR
RAR
MOV      M,A
MOV      A,B      ;(RE[Z(K)]+RE[Z(N-K)])/2=
ADD      D      ;=RE[X(K)]
RLC
RAR
RAR
DCX      H
MOV      M,A
MOV      A,C      ;(IM[Z(K)]+IM[Z(N-K)])/2=
ADD      E      ;=RE[Y(K)]
RLC
RAR
RAR
LHLD     ADR2
MOV      M,A
MOV      A,D      ;(RE[Z(N-K)]-RE[Z(K)])/2=
SUB      B      ;=IM[Y(K)]
RLC

```

```

RAR
RAR
INX      H
MOV     M,A
MOV     C,A      ;IM[Y(K)]
LHLD   ADR11
INX     H
MOV     E,M      ;IM[W(K)]
CALL   MULTIPLICATION ;IM[Y(K)]*IM[W(K)]
LXI    H,ADR12
MOV     M,B
LHLD   ADR2
MOV     C,M      ;RE[Y(K)]
LHLD   ADR11
MOV     E,M      ;RE[W(K)]
CALL   MULTIPLICATION ;RE[Y(K)]*RE[W(K)]
LDA    ADR12
ADD     B      ;RE[Y(K)]*RE[W(K)]+
STA    ADR12  ;+IM[Y(K)]*IM[W(K)]
LHLD   ADR2
INX     H
MOV     C,M      ;IM[Y(K)]
LHLD   ADR11
MOV     E,M      ;RE[W(K)]
CALL   MULTIPLICATION ;IM[Y(K)]*RE[W(K)]
LXI    H,ADR13
MOV     M,B
LHLD   ADR2
MOV     C,M      ;RE[Y(K)]
LHLD   ADR11
INX     H
MOV     E,M      ;IM[W(K)]
CALL   MULTIPLICATION ;RE[Y(K)]*IM[W(K)]
LDA    ADR13
SUB     B      ;IM[Y(K)]*RE[W(K)]-
MOV     B,A    ;-RE[Y(K)]*IM[W(K)]
LXI    H,ADR12
MOV     D,M
LHLD   ADR1
MOV     A,M
MOV     E,A
ADD     D      ;RE[X(K)]+(RE[Y(K)]*RE[W(K)]+
MOV     C,A    ;+IM[Y(K)]*IM[W(K)])
MOV     A,E
SUB     D      ;RE[X(K)]-(RE[Y(K)]*RE[W(K)]+
MOV     E,A    ;+IM[Y(K)]*IM[W(K)])
INX     H
MOV     A,M    ;IM[X(K)]
MOV     D,A
SUB     B      ;IM[X(K)]-(IM[Y(K)]*RE[W(K)]-
MOV     H,A    ;-RE[Y(K)]*IM[W(K)])
MOV     A,D
ADD     B      ;IM[X(K)]+(IM[Y(K)]*RE[W(K)]-
MOV     B,A    ;-RE[Y(K)]*IM[W(K)])
MOV     D,H
CMA
INR    A
LHLD   ADR9    ;ВЫВОД РЕЗУЛЬТАТОВ В ПАМЯТЬ

```



```

MOV      M,C
INX      H
MOV      M,B
INX      H
MOV      M,E
INX      H
MOV      M,D
LHLD     ADR10
INX      H
MOV      M,A
DCX      H
MOV      M,C
DCX      H
MOV      A,D
CMA
INR      A
MOV      M,A
DCX      H
MOV      M,E
LHLD     ADR1      ; ОПР-Е СЛЕД. АДР. Z(K) В
INX      H          ; БЛОКЕ И ПРОВЕРКА КОНЦА
INX      H          ; БЛОКА
SHLD     ADR1
XCHG
LHLD     ADR2
MOV      A,E
CMP      L
JNZ      GROOP
MOV      A,D
CMP      H
JNZ      GROOP
LHLD     ADR4      ; ПРОВЕРКА КОНЦА ВЫЧИСЛЕНИЯ
MOV      A,H
RAR
MOV      H,A
MOV      A,L
RAR
MOV      L,A
RC
SHLD     ADR4
LHLD     ADR3      ; ОПР-Е АДР. Z(K) В СЛЕД.
XCHG     ; БЛОКЕ
LHLD     ADR2
DAD      D
SHLD     ADR1
LHLD     ADR11     ; ОПР-Е АДР. СЛЕД. КОЭФ. W
INX      H
INX      H
DAD      D
SHLD     ADR11
XCHG     ; ОПР-Е АДР. A(K) В СЛЕД.
DCX      H          ; БЛОКЕ
DAD      H
XCHG
LHLD     ADR10
DAD      D
SHLD     ADR9
JMP      BLOCKR

```

GROUP: DCX H ; ОПР-Е СЛЕД. АДР. Z(N-K) В
DCX H ; БЛОКЕ
SHLD ADR2
LHLD ADR11 ; ОПР-Е СЛЕД. АДР. КОЭФ. W
INX H ; В БЛОКЕ
INX H
SHLD ADR11
LHLD ADR9 ; ОПР-Е СЛЕД. АДР. A(K) В
INX H ; БЛОКЕ
INX H
INX H
INX H
SHLD ADR9
LHLD ADR10 ; ОПР-Е СЛЕД. АДР. A(N-K) В
DCX H ; БЛОКЕ
DCX H
DCX H
DCX H
SHLD ADR10
JMP CON

```

;*****
;* MULTIPLICATION- ПОДПРОГРАММА УМНОЖЕНИЯ ДРОБНЫХ ЧИСЕЛ *
;*****
;      МНОЖИТЕЛИ НАХОДЯТСЯ В РЕГИСТРАХ С И Е, РЕЗУЛЬТАТ
;      ПРОИЗВЕДЕНИЯ - В РЕГИСТРЕ В
;      ФРАГМЕНТ ПОДПРОГРАММЫ УМНОЖЕНИЯ, ПОМЕЧЕННЫЙ "====",
;      В ПРОГРАММАХ БПХ2МОЖНО ИСКЛЮЧИТЬ
;-----

```

```

MULTIPLICATION: XRA      A          ;ОЧИСТКА РЕГИСТРОВ D,В,L
                 MOV      D,A
                 MOV      В,A
                 MOV      L,A
                 ADD      E
;-----

```

```

                 JP       J0
                 CMA
                 INR      A          ;ОПР-Е АБСОЛ. ВЕЛИЧИНЫ СОДЕР-
                 INR      В          ;ЖИМОГО РЕГИСТРА Е И УЧЕТ ЕГО
                                     ;ЗНАКА
;-----

```

```

J0:              RAL
                 MOV      E,A
                 XRA      A          ;ОПР-Е АБСОЛ. ВЕЛИЧИНЫ СОДЕР-
                 ADD      C          ;ЖИМОГО РЕГИСТРА С И УЧЕТ ЕГО
                 JP       J1          ;ЗНАКА
                 CMA
                 INR      A
                 INR      В

```

```

J1:              RAL
                 MOV      H,A
                 DAD      H          ;ТЕЛО ПОДПРОГРАММЫ УМНОЖЕНИЯ
                 JNC      Q
                 DAD      D

```

```

Q:              DAD      H
                 JNC      Q1
                 DAD      D

```

```

Q1:             DAD      H
                 JNC      Q2
                 DAD      D

```

```

Q2:             DAD      H
                 JNC      Q3
                 DAD      D

```

```

Q3:             DAD      H
                 JNC      Q4
                 DAD      D

```

```

Q4:             DAD      H
                 JNC      Q5
                 DAD      D

```

```

Q5:             DAD      H
                 JNC      J2
                 DAD      D

```

```

J2:             XRA      A          ;ОКРУГЛЕНИЕ РЕЗУЛЬТ. УМНОЖЕНИЯ
                 ADD      L
                 JP       J3
                 INR      H

```

```

J3:             MOV      A,В          ;ВОССТАНОВЛЕНИЕ РЕЗУЛЬТ. УМНОЖ

```

```

RRC                ;С УЧЕТОМ ЗНАКА МНОЖИТЕЛЕЙ
MOV                B,H
RNC
MOV                A,H
CMA
INR                A
MOV                B,A
RET

```

```

;*****
;*                SCALING- ПОДПРОГРАММА МАСШТАБИРОВАНИЯ                *
;*****
;                В ПРОГРАММАХ БЛХ ВМЕСТО SIZE2 ИСПОЛЬЗУЕТСЯ SIZE
;-----

```

```

SCALING:           XRA        A            ;ОПР-Е МАКС. ЗНАЧЕНИЯ ОБР.
                   MOV        B,A        ;ПОСЛ-ТИ
                   LXI        H,SIZE2
                   XCHG
REPEAT:           LXI        H,BEGINSIGNAL
                   XRA        A
                   ADD        M
                   JP         COMPARISON
COMPARISON:       CMP        B
                   JC         INCREASE
INCREASE:         MOV        B,A
                   INX        H
                   DCX        D
                   MOV        A,D
                   ORA        E
                   JNZ        REPEAT
                   XRA        A            ;ПРОВЕРКА НЕОБХ. МАСШТАБ.
                   ADD        B
                   CPI        2EH
                   RC
                   LXI        H,SIZE2
                   XCHG
                   LXI        H,BEGINSIGNAL
                   MOV        B,A
                   LDA        ADR14      ;РАБОТА СЧЕТЧИКА МАСШТАБ.
                   DCR        A
                   STA        ADR14
                   JNZ        SHIFT1
                   MOV        A,B
                   CPI        5BH
                   JNC        SHIFT2
SHIFT1:           MOV        A,M        ;СДВИГ ОБР. ПОСЛ-ТИ НА 1 РАЗР.
                   RLC
                   RAR
                   RAR
                   MOV        M,A
                   INX        H
                   DCX        D
                   MOV        A,D
                   ORA        E
                   JNZ        SHIFT1
                   RET
SHIFT2:           MOV        A,M        ;СДВИГ ОБР. ПОСЛ-ТИ НА 2 РАЗР.

```

```

RLC
RAR
RAR
RLC
RAR
RAR
MOV    M,A
INX    H
DCX    D
MOV    A,D
ORA    E
JNZ    SHIFT2
RET

```

```

;*****
; UNSCRAMBLING- ПОДПРОГРАММА ДВОИЧНО-ИНВЕРСНОЙ ПЕРЕСТАНОВКИ *
;*****
; ФРАГМЕНТЫ ПОДПРОГРАММЫ, ПОМЕЧЕННЫЕ "====", В ПРОГРАММАХ
; БПХ ДОЛЖНЫ БЫТЬ ИСКЛЮЧЕНЫ
;-----

```

```

UNSCRAMBLING:  LXI    H,SIZE ;ОПР-Е N-1
                DCX    H
                SHLD   ADR12
                LXI    H,DIS2;КОЛ-ВО ОБР. ПАР
                PUSH   H
                LXI    B,1 ;НАЧ. АДР. В ПРЯМОМ ПОР-КЕ
                LXI    H,DIS1 ;СМЕЩЕНИЕ = N/2
                XCHG
                JMP    SEND2
BEGIN:         PUSH   B ;СРАВНЕНИЕ ПРЯМОГО И ДВОИЧНО-
                PUSH   D ;ИНВЕРСНОГО АДРЕСОВ
                MOV    A,B
                CMP    D
                JC     SEND1
                JNZ    BIGGER
                MOV    A,C
                CMP    E
                JC     SEND1
                JZ     EQUAL
BIGGER:        LHLD   ADR12 ;ОПР-Е (N-1)-АДР.В ПРЯМОМ ПОР-
                MOV    A,L ;КЕ И (N-1)-АДР. В ДВ.-ОБР.
                SUB    E ;ПОР-КЕ
                MOV    E,A
                MOV    A,H
                SBB   D
                MOV    D,A
                MOV    A,L
                SUB    C
                MOV    C,A
                MOV    A,H
                SBB   B
                MOV    B,A
SEND1:         MOV    H,B ;ПЕРЕСТАНОВКА СОДЕРЖИМОГО
                MOV    L,C ;ЯЧЕЕК ПАМЯТИ ПО АДР. ОПР.ВЫШЕ
                LXI    B,BEGINSIGNAL
;1=====
                DAD    H
;1=====
                DAD    B

```

```

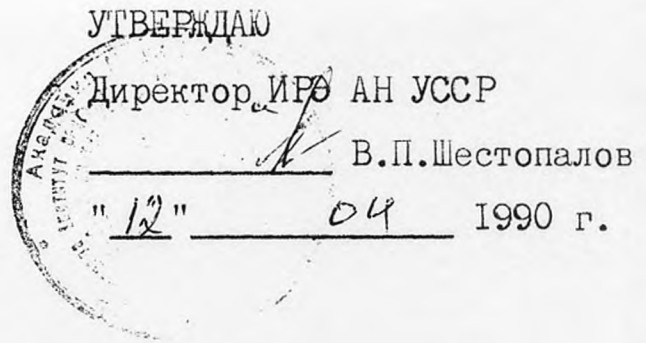
MOV      A,M
XCHG
;2=====
DAD      H
;2=====
DAD      B
MOV      B,M
MOV      M,A
;3=====
INX      H
MOV      C,M
;3=====
XCHG
MOV      M,B
;4=====
INX      H
MOV      A,M
MOV      M,C
XCHG
MOV      M,A
;4=====
EQUAL:   POP      D      ;ОПР-Е ПР. И ДВ.-ОБР. АДР.
POP      B      ;СЛЕД. ПАРЫ
INX      B
LXI      H,DIS1
DAD      D
XCHG
SEND2:   PUSH     B      ;ПЕРЕСТАНОВКА СОДЕРЖИМОГО
MOV      H,B      ;ЯЧЕЕК ПАМЯТИ ОП ОПР. АДР.
MOV      L,C
LXI      B,BEGINSIGNAL
;5=====
DAD      H
;5=====
DAD      B
MOV      A,M
XCHG
;6=====
DAD      H
;6=====
DAD      B
MOV      B,M
MOV      M,A
;7=====
INX      H
MOV      C,M
;7=====
XCHG
MOV      M,B
;8=====
INX      H
MOV      A,M
MOV      M,C
XCHG
MOV      M,A
;8=====
POP      B
POP      H      ;ПРОВЕРКА КОЛ-ВА ОБР. ПАР

```

```

DCX      H
MOV      A,H
ORA      L
RZ
PUSH     H
INX      B      ;ОПР-Е СЛЕД. АДР. В ПР. ПОР-КЕ
PUSH     B      ;И СООТВ. ЕМУ АДР. В ДВ.-ИНВ.
LXI      H,DIS1 ;ПОР-КЕ
LXI      D,0
BITREVERS: MOV    A,B
RAR
MOV      B,A
MOV      A,C
RAR
MOV      C,A
MOV      A,E
RAL
MOV      E,A
MOV      A,D
RAL
MOV      D,A
MOV      A,H
RAR
MOV      H,A
MOV      A,L
RAR
MOV      L,A
JNC      BITREVERS
POP      B
JMP      BEGIN

```



А К Т

о внедрении результатов диссертационной
работы Волынца Виктора Ивановича
"Разработка эффективных алгоритмов обра-
ботки сигналов для микропроцессорных
анализаторов спектра"

Настоящий акт удостоверяет, что в рамках НИР "Спектр-2", выполнявшейся кафедрой микроэлектроники Винницкого политехнического института по х/д 4329, в ИРЭ АН УССР внедрен микропроцессорный анализатор спектра, при разработке которого были использованы результаты научных исследований, полученные Волынцем В.И. при выполнении его диссертационной работы "Разработка эффективных алгоритмов обработки сигналов для микропроцессорных анализаторов спектра", в частности:

1. Эффективный алгоритм быстрого преобразования Хартли для обработки вещественных последовательностей, выполняемый с замещением и требующий минимального объема вычислений и памяти.

2. Выражения для оценки вычислительных ошибок алгоритмов быстрого преобразования Хартли при представлении чисел в дополнительном коде в системе счисления с фиксированной запятой.

3. Эффективная программная реализация алгоритма быстрого преобразования Хартли на языке ассемблера микропроцессора КР580 ВМ30А.

4. Структурные и принципиальные схемы устройств для повышения производительности микропроцессорных систем для обработки информации.

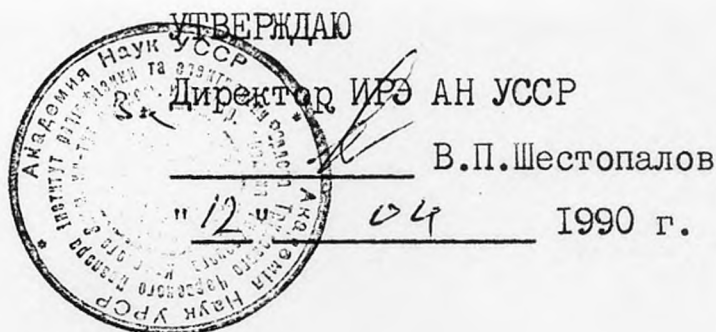
Ожидаемый годовой экономический эффект от внедрения микро-

процессорного анализатора спектра составляет 29700 руб. Расчет экономического эффекта прилагается.

Старший научный сотрудник
Старший научный сотрудник
Младший научный сотрудник



Г.И.Хлопов
А.А.Костенко
А.А.Могила



РАСЧЕТ

годового экономического эффекта от внедрения результатов диссертационной работы Волынца Виктора Ивановича "Разработка эффективных алгоритмов обработки сигналов для микропроцессорных анализаторов спектра"

В 1986-1989 гг. в рамках НИР "Спектр-2", выполнявшейся на кафедре микроэлектроники ВПИ по х/д № 4329, разработан и внедрен в ИРЭ АН УССР микропроцессорный анализатор спектра, в основу которого положены результаты диссертационной работы Волынца Виктора Ивановича "Разработка эффективных алгоритмов обработки сигналов для микропроцессорных анализаторов спектра".

Расчет годового экономического эффекта выполнен с использованием "Методики определения экономической эффективности использования в народном хозяйстве новой техники, изобретений и рационализаторских предложений.-М.:ВНИИПИ, 1986".

Расчет годового экономического эффекта производится по формуле:

$$\mathcal{E} = \left[Z_1 \alpha \beta + \frac{(U_1' - U_2') - \varepsilon_n (K_2' - K_1')}{P_2 + \varepsilon_n} - Z_2 \right] \cdot A_2 ,$$

где Z_1 и Z_2 - приведенные затраты единицы соответственно базового и нового приборов, руб; α - коэффициент учета роста качественных показателей нового прибора по сравнению с базовым; β - коэффициент учета изменения срока службы нового прибора по срав-

нению с базовым; $\frac{[(C_1' - C_2') - \varepsilon_n(K_2' - K_1')]}{(P_2 + \varepsilon_n)}$ - экономия потребителя на текущих издержках эксплуатации C_2' и сопутствующих капитальных вложениях K_2' нового прибора по сравнению с базовым (C_1' и K_1'), руб; ε_n - нормативный коэффициент эффективности равный 0,1; P_2 - доли отложений от балансовой стоимости на реновацию нового прибора; A_2 - годовой объем производства новых приборов.

В свою очередь приведенные затраты Z определяются по формуле:

$$Z = C + \varepsilon_n K,$$

где C - себестоимость изделия, руб; K - удельные капитальные вложения, руб.

Исходные данные для расчетов сведем в таблицу. При этом в качестве базового прибора принимается отечественный серийно выпускаемый анализатор спектра СК4-72/2.

Таблица

№ п/п	Показатель	Обозначение	Ед. изм.	Базовый прибор	Новый прибор
1	Годовой объем производства	A	шт.	-	1
2	Себестоимость изделия	C	руб.	17000	3500
3	Удельные капитальные вложения	K	руб.	-	8000
4	Срок службы	T_c	лет	10	10
5	Показатель качества	β	ед.	1	2

Приведенные затраты Z_1 и Z_2 определяются следующим образом:

$$Z_1 = C_1 = 17000 \text{ руб.}$$

$$Z_2 = C_2 + \varepsilon_n K_2 = 3500 + 0,1 \times 8000 = 4300 \text{ руб.}$$

Поскольку срок службы базового и нового приборов одинаков ($\beta = 1$), а также равны текущие издержки эксплуатации и сопутствующие капитальные вложения базового и нового приборов, годовой

экономический эффект составит:

$$\mathcal{E} = [Z_1 - Z_2] = 17000 \times 2 - 4300 = 29700 \text{ руб.}$$

Настоящий расчет не является основанием для получения денежного вознаграждения.

Гл. бухгалтер ИРЭ АН УССР

Нач. ТЭО

Зав. НИИ группой, с.н.с.

Павлова
Ищенко
Хлопов

В.В.Павлинова

А.Е.Ищенко

Г.И.Хлопов