

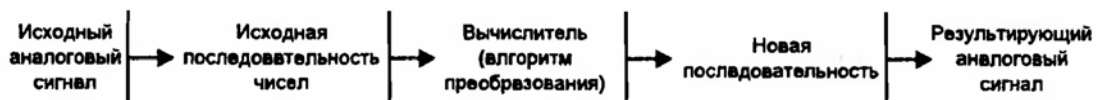


1	Цифровая обработка сигналов.....	1
1.1	Обобщенная схема цифровой обработки аналоговых сигналов.....	1
1.2	Способы реализации ЦОС.....	3
2	Особенности DSP.....	5
2.1	Определение.....	5
2.2	Принцип работы.....	5
2.3	Особенности архитектуры.....	6
2.4	Области применения.....	6
2.5	История.....	7
2.6	Основные параметры ЦСП.....	9
3	Устройство (архитектура).....	12
3.2	Классификация ЦСП по архитектуре.....	12
3.3	ЦСП с архитектурой VLIW.....	15
3.4	Суперскалярные ЦСП.....	15
3.5	Гибридные ЦСП.....	16
3.6	Классификация ЦСП по назначению.....	17
3.7	Программирование ЦСП.....	17
4	Основные производители DSP.....	18
4.1	Ссылки на основных производителей.....	18
4.2	DSP от Microchip dsPIC30F и dsPIC33F.....	18
4.3	Отечественные ЦСП.....	21
5	Алгоритмы ЦОС.....	23
5.1	Ких фильтр.....	23
5.2	БИХ фильтр.....	24
5.3	БПФ.....	25

1 Цифровая обработка сигналов

ЦОС- цифровая обработка сигналов

1.1 Обобщенная схема цифровой обработки аналоговых сигналов



- аналоговый антиэлайсинговый фильтр низких частот (АФНЧ);
- аналого-цифровой преобразователь (АЦП);
- устройство цифровой обработки сигналов (вычислитель);
- цифро-аналоговый преобразователь (ЦАП);
- аналоговый сглаживающий фильтр низких частот (СФНЧ).

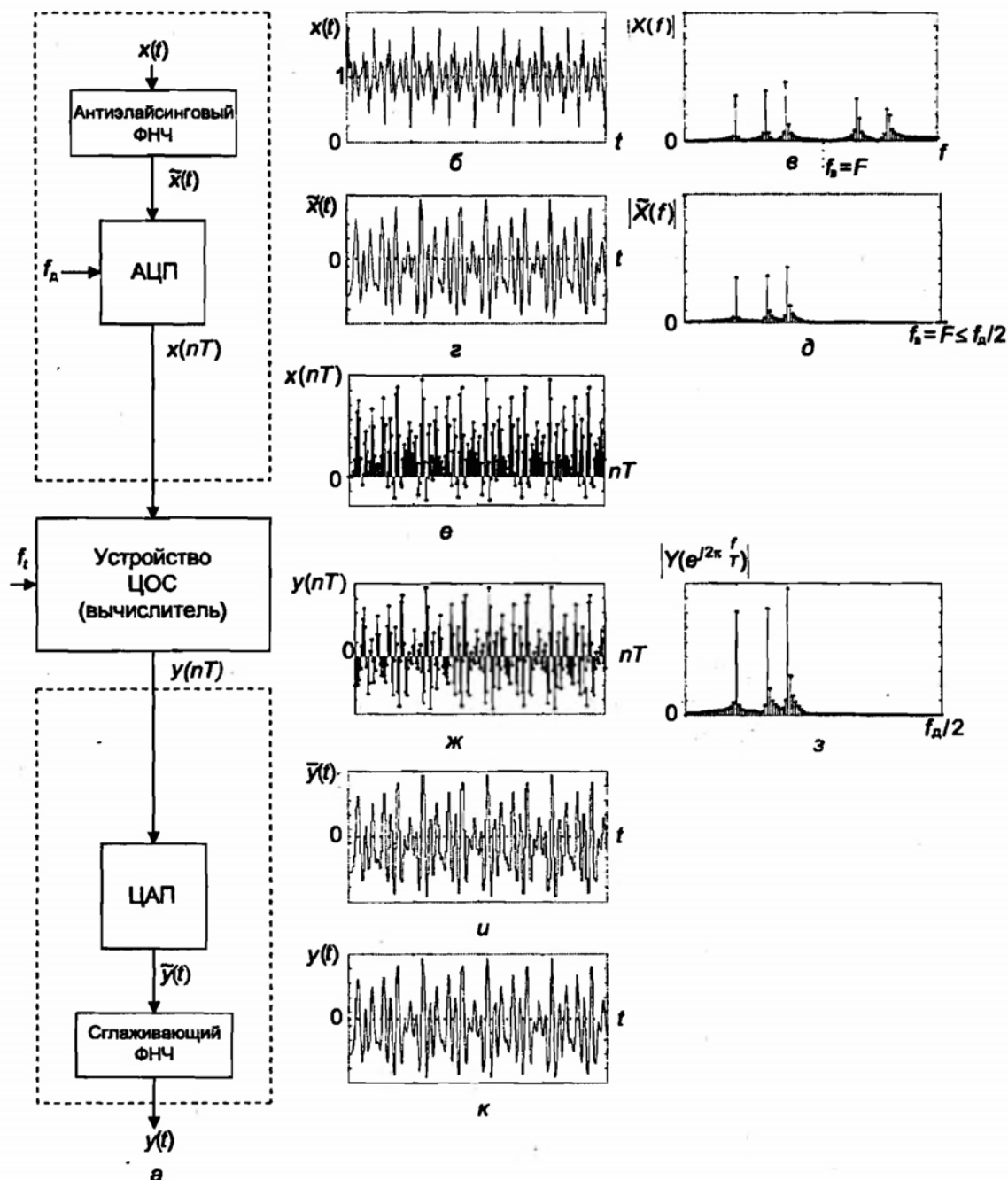


Рис. 1.2. Обобщенная схема цифровой обработки аналоговых сигналов

Устройство, объединяющее АФНЧ и АЦП, называется *кодером*. Кодер формирует последовательность чисел, соответствующую обрабатываемому аналоговому сигналу.

Устройство ЦОС (вычислитель — это и есть ЦСП (DSP),

Устройство, объединяющее ЦАП и СФНЧ, называется *декодером*. Декодер по принятому цифровому сигналу формирует аналоговый сигнал, т. е. производит преобразования, обратные происходившим в кодере.

Дискретизация по времени (или *дискретизация*) представляет собой процедуру взятия мгновенных значений сигнала $x(t)$ через равные промежутки времени T . Мгновенные значения $x(nT)$ называются *выборками*, или *отсчетами*, время T — *периодом дискретизации*, n указывает порядковый номер отсчета.

Квантование отсчетов по уровням (или *квантование*) производится с целью формирования последовательности чисел: весь диапазон изменения величины



отсчетов разбивается на некоторое количество дискретных уровней, и каждому отсчету по определенному правилу присваивается значение одного из двух ближайших уровней квантования, между которыми оказывается данный отсчет.

В результате получается последовательность чисел $x(nT) = X(N)$, представляемых в двоичном коде. Количество уровней определяется разрядностью b АЦП; так, если $b = 3$, всего можно иметь $K = 2^b = 2^3 = 8$ уровней квантования, а минимальное и максимальное значения отсчетов равны соответственно $0 \Leftrightarrow 000$ и $7 \Leftrightarrow 111$. Ясно, что квантованный отсчет отличается от выборки $x(nT)$. Это отличие выражается ошибкой квантования

$$\varepsilon_{\text{кв}} = x_u(nT) - x(nT),$$

которая тем больше, чем меньше b . Максимальная ошибка квантования при использовании округления в качестве приближения равна половине шага квантования Q

$$\max |\varepsilon_{\text{кв}}| = Q/2, \text{ где } Q = q_1 = 2^{-b}.$$

1.2 Способы реализации ЦОС

Ни рисунке 1.20 показаны способы реализации ЦОС

1.2.1 Программная

Программная реализация подразумевает представление алгоритма в виде программы, которую последовательно от команды к команде выполняет один или одновременно несколько независимых блоков. Программа должна быть написана на языке программирования, соответствующем конкретному операционному блоку. Так, для персонального компьютера это будет любой из языков высокого уровня (Pascal, C++, Java и др.), а для микропроцессорного комплекта или цифрового процессора — соответствующий язык ассемблера.

К достоинствам программной реализации относятся:

- неизменная структура системы при различных алгоритмах и областях применения;
- хорошая гибкость, позволяющая достаточно легко изменять алгоритмы работы системы за счет коррекции или изменения программы;
- существенное ускорение, облегчение и удешевление проектирования, изготовления и отладки системы, поскольку вместо прибора разрабатывается программа.

Недостатком программной реализации является относительно низкое быстродействие по причине последовательного выполнения операций программы в одном процессоре: как бы ни увеличивали скорость выполнения команд, она будет оставаться ниже производительности соответствующего устройства, реализованного аппаратно. Отсюда вытекает задача обеспечения реального времени, которая подразумевает два обстоятельства:

- 1) время обработки одного отсчета или группы отсчетов сигнала $t_{\text{об}}$ должно быть меньше допустимого времени задержки $t_{\text{доп}}$: $t_{\text{об}} < t_{\text{доп}}$;
контроль за выполнением этого условия осуществляется как при написании программы, так и при ее отладке;
- 2) цикл работы программы и моменты поступления отсчетов входного сигнала $x(n)$ должны быть строго согласованы по времени, т. е. начало обработки очередного отсчета $x(n)$ должно совпадать или следовать за поступлением этого отсчета; с другой стороны, результат обработки $y(n)$ должен быть выведен согласно темпу работы внешнего устройства, который не обязан совпадать с темпом поступления отсчетов $x(n)$.

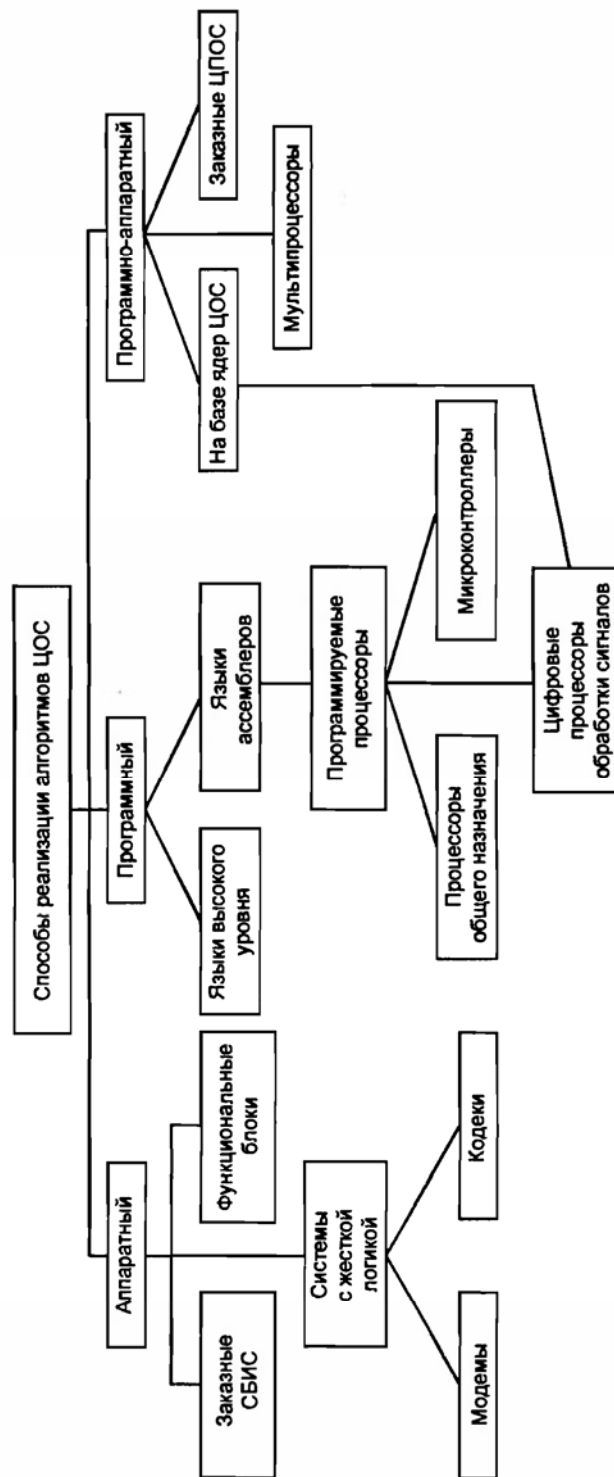


Рис. 1.20. Способы реализации алгоритмов и систем ЦОС

1.2.2 Аппаратно-программная реализация

Аппаратно-программная реализация подразумевает, что часть функций системы ЦОС выполняется аппаратно (аналого-цифровое и цифро-аналоговое преобразования, умножение, умножение с накоплением, прием/передача данных и др.), а другая часть функций выполняется программно. Пример аппаратно-программной реализации показан на рис. ???, где к процессору, работающему по заданной программе, подключены:

- аналого-цифровой (АЦП) и цифро-аналоговый (ЦАП) преобразователи;
- модули внешней памяти, хранящие программы, разнообразные константы и таблицы функций (например, \sin и \cos), что позволяет заменять длительное их вычисление быстрым обращением к памяти (такая реализация является прекрасным примером обмена скорости

вычисления на дополнительное оборудование, т. е. действует закон, согласно которому невозможно получить абсолютный выигрыш: любой выигрыш требует платы за себя);

2 Особенности DSP

2.1 Определение

Цифровой сигнальный процессор (ЦСП) ([англ.](#) Digital signal processor, DSP; сигнальный микропроцессор (СМП); процессор цифровых сигналов (ПЦС)) — специализированный [микропроцессор](#), предназначенный для [цифровой обработки сигналов](#) (обычно в [реальном масштабе времени](#))

Главное отличие ЦСП от МП и МК заключается в особенности архитектуры (см далее) ориентированной на максимально быстрое выполнение операции умножение с накоплением над массивом чисел. Основным можно считать:

1) введение специальных MAC (умножение с накоплением) команд, которые за один такт выполняют действия $Y := X + A \times B$, равнозначные 2-5 обычным командам МП и МК.

2) Кроме того оптимизирован доступ к элементам массива- как аргументам MAC команды. Оцифрованные отсчёты входного сигнала записываются в аппаратно организованные кольцевые буферы. Кольцевой буфер похож на стек, но в отличие от стека он никогда не переполняется. Изменение указателей в кольцевом буфере на запись в буфер и (стирание старых записей, добавление новых) и чтение может выполняться аппаратным ФМ, поддерживающим кольцевые буфера. Элементы кольцевого буфера могут быть аргументами MAC команды, причём переход к следующему элементу кольцевого буфера происходит автоматически как автоинкрементирование указателей при выполнении текущей MAC команды. При работе с данными во внешней памяти применяется прямой доступ к памяти (ПДП), используя контроллер ПДП или контроллер кольцевого буфера с ПДП. Для асинхронной записи и чтения в буфер часто используется 2-х портовая память.

Для работы с синфазными каналами в некоторых ЦСП существует 2 ядра и 2 набора буферов и контроллеров управления кольцевыми буферами.

MAC команды присутствуют в мультимедийных расширениях набора инструкций современных МП от Intel (MMX, SSE, SSE2, SSE3) и AMD (3DNow, 3DNow2). Но в эти процессоры они были добавлены как расширение команд MISC процессора и в результате выполняются намного медленнее, чем в ЦСП. В ЦСП MAC команды изначально сходят с состав RISC процессора.

2.2 .Принцип работы

КИХ фильтр

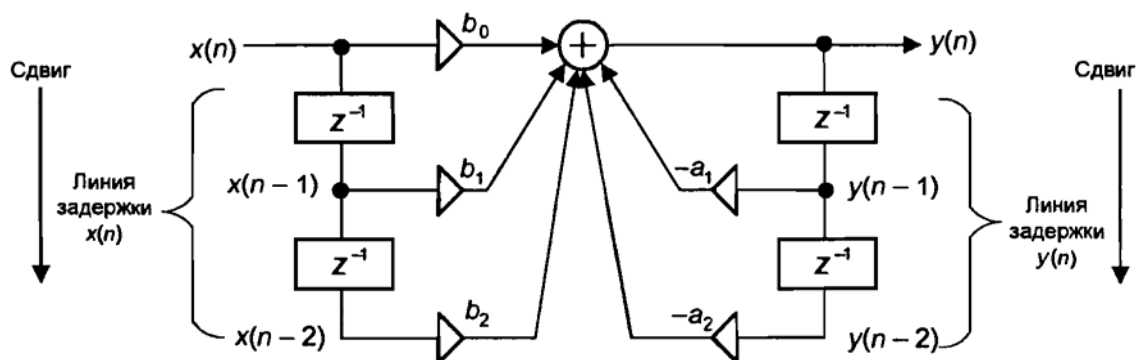


Рисунок Реализация аналогового КИХ фильтра

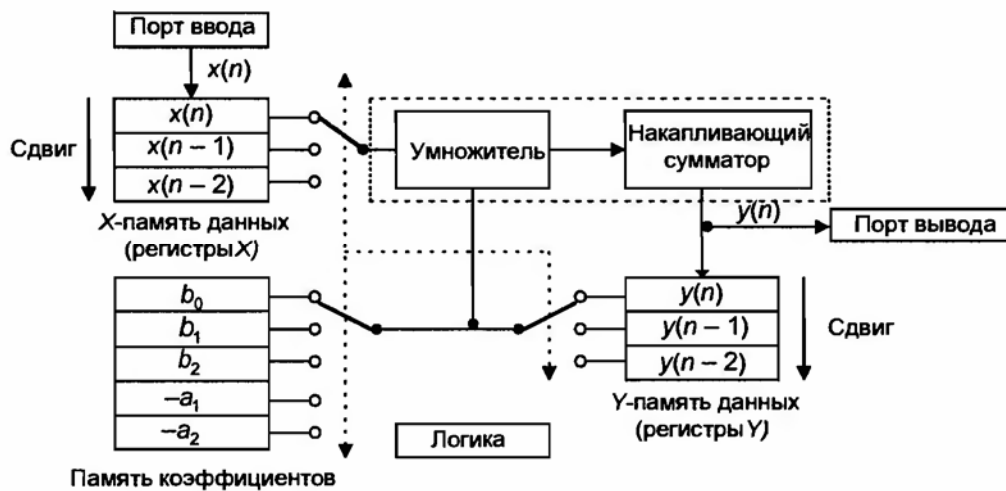


Рисунок Реализация Ких фильтра на ЦСП

2.3 Особенности архитектуры

Архитектура сигнальных процессоров, по сравнению с микропроцессорами настольных компьютеров, имеет некоторые особенности:

- [Гарвардская архитектура](#) (разделение памяти команд и данных), как правило [модифицированная](#);
 - Большинство сигнальных процессоров имеют встроенную [оперативную память](#), из которой может осуществляться выборка нескольких [машинных слов](#) одновременно. Нередко встроено сразу несколько видов оперативной памяти, например, в силу Гарвардской архитектуры бывает отдельная память для инструкций и отдельная - для данных.
 - Некоторые сигнальные процессоры обладают одним или даже несколькими встроенными [постоянными запоминающими устройствами](#) с наиболее употребительными подпрограммами, таблицами и т.п..
- Аппаратное ускорение сложных вычислительных инструкций, то есть быстрое выполнение операций, характерных для цифровой обработки сигналов, например, операция «умножение с накоплением» (MAC) ($Y := X + A \times B$) обычно выполняется за один такт. Обычно процедуры умножения, и основных функций (синуса, косинуса, логарифма и т.п.) выполняются чтением готового результата из таблиц значений.
- «Бесплатные» по времени [циклы](#) с заранее известной длиной. Поддержка векторно-конвейерной обработки с помощью генераторов адресных последовательностей.
- Детерминированная работа с известными временами выполнения команд, что позволяет выполнять планирование работы в [реальном времени](#).
- Сравнительно небольшая длина [конвейера](#), так что незапланированные [условные переходы](#) могут занимать меньшее время, чем в универсальных процессорах.
- Экзотический набор [регистров](#) и инструкций, часто сложных для [компиляторов](#). Некоторые архитектуры используют [VLIW](#).
- По сравнению с [микроконтроллерами](#), ограниченный набор периферийных устройств — впрочем, существуют «переходные» чипы, сочетающие в себе свойства DSP и широкую периферию микроконтроллеров.

2.4 Области применения

- Коммуникационное оборудование:
 - уплотнение каналов передачи данных;
 - кодирование аудио- и видеопотоков;
- Связь и телекоммуникация: базовые станции, DVB - приемники и т.д.;



- Сигнальная обработка: БПФ, фильтрация, корреляция, быстрая свертка;
- Системы гидро- и радиолокации;
- Фазированные антенные решетки;
- Распознавание речи и изображений;
- Речевые и музыкальные синтезаторы;
- Анализаторы спектра;
- Графические ускорители;
- Мультимедийная обработка изображений и цифровое телевидение (H.264/AVC (CIF), JPEG 2000 и т.д.);
- Мультимедийная обработка звука (MPEG-1 Audio Layer3 [MP3], AMR, WMA, AAC и другие звуковые кодеки);
- Управление объектами с использованием высокоточных адаптивных методов;
- Высокоточная обработка данных для малогабаритных мобильных и встраиваемых систем;
- Системы промышленного контроля.
- Управление технологическими процессами;
- Другие области, где необходима быстродействующая обработка сигналов, в том числе в реальном времени.

В настоящее время выделяют следующие основные направления ЦОС:

- линейная фильтрация;
- спектральный анализ;
- частотно-временной анализ;
- адаптивная фильтрация;
- нелинейная обработка;
- многоскоростная обработка.

1	Линейная фильтрация	Селекция сигнала в частотной области; синтез фильтров, согласованных с сигналами; частотное разделение каналов; цифровые преобразователи Гильберта и дифференциаторы; корректоры характеристик каналов
2	Спектральный анализ	Обработка речевых, звуковых, сейсмических, гидроакустических сигналов; распознавание образов
3	Частотно-временной анализ	Компрессия изображений, гидро- и радиолокация, разнообразные задачи обнаружения
4	Адаптивная фильтрация	Обработка речи, изображений, распознавание образов, подавление шумов, адаптивные антенные решетки
5	Нелинейная обработка	Вычисление корреляций, медианная фильтрация; синтез амплитудных, фазовых, частотных детекторов, обработка речи, векторное кодирование
6	Многоскоростная обработка	Интерполяция (увеличение) и децимация (уменьшение) частоты дискретизации в многоскоростных системах телекоммуникации, аудиосистемах

2.5 История

2.5.1 Предшествующие разработки

До 1980 года несколько компаний выпустили устройства, которые можно считать предшественниками ЦСП. Так, в 1978 [Intel](#) выпускает «процессор аналоговых сигналов» 2120. В его состав входили [АЦП](#), [ЦАП](#) и процессор обработки цифровых данных, однако аппаратная функция умножения отсутствовала. В 1979 [AMI](#) выпускает S2811 — периферийное устройство, управляемое основным процессором компьютера. Оба изделия не достигли успеха на рынке.

2.5.2 Первое поколение (начало 1980-х)

Основную историю ЦСП принято отсчитывать от 1979—1980 годов, когда [Bell Labs](#) представила первый однокристалльный ЦСП Mac 4, а также на «IEEE International Solid-State Circuits Conference '80» были показаны [μMPD7720](#) компании [NEC](#) и [DSP1](#) компании [AT&T](#), которые, однако, не получили широкого распространения. Стандартом де-факто стал выпущенный чуть позже кристалл [TMS32010](#) фирмы [Texas Instruments](#), по многим параметрам и удачным техническим решениям превосходящий изделия конкурентов. Вот некоторые его характеристики:



- [АЛУ](#):
 - Размер слова: 16 бит;
 - Разрядность вычислителя: 32 бит;
 - [Быстродействие](#): 5 млн операций сложения или умножения в секунду;
- Длительность командного цикла: 160—280 нс;
- Память:
 - [ОЗУ](#): 144—256 слов;
 - [ПЗУ](#) программ: 1,5—4 К слов;
 - [ППЗУ](#): до 4К слов (отдельные модели);
- Внешняя шина:
 - Разрядность: 16 бит;
 - Адресуемое пространство: 4К слов
 - Пропускная способность: 50 Мбит/с
- Устройства ввода-вывода: 8 портов по 16 разрядов;

2.5.3 Второе поколение (середина 1980-х)

Благодаря прогрессу в полупроводниковых технологиях, в этот период были выпущены изделия, имеющие расширенные функции по сравнению с первым поколением. К характерным отличиям можно отнести:

- Увеличение объема ОЗУ до 0,5 К слов;
- Добавлена возможность подключения внешней памяти программ и внешней памяти данных объемом до 128 К слов;
- Быстродействие повышено в 2—4 раза;
- Улучшенные подсистемы прерываний и ввода-вывода.

Много позднее также были выпущены устройства, формально относящиеся ко второму поколению, но имеющие следующие усовершенствования:

- Увеличена разрядность данных;
- Пониженное напряжение питания и, как следствие, энергопотребление;
- Введены режимы экономии энергии;
- Аппаратная поддержка мультипроцессорности (система совместного доступа к внешней памяти);
- Аппаратная поддержка кольцевых буферов;
- Аппаратная поддержка операций циклов;
- Расширены способы адресации;
- Две внутренние шины данных, что позволяет значительно ускорить парную обработку данных (координаты X/Y, действительная и мнимая часть и т. д.), либо виртуально удвоить разрядность обрабатываемых данных;
- Введена кэш-память.

2.5.4 Третье поколение (конец 1980-х)

Третье поколение ЦСП принято связывать с началом выпуска изделий, реализующих арифметику с плавающей запятой. Характерные особенности первых выпущенных образцов:

- Производительность: порядка 20-40 млн оп./сек. ([MIPS](#));
- Два блока ОЗУ по 1 К 32-разрядных слов с возможностью одновременного доступа;
- Кэш-память объемом 64 слова;
- Разрядность регистров: 32 бит;
- Разрядность АЛУ: 40 бит;
- Регистры для операций с повышенной точностью;
- Встроенные контроллеры ПДП;



- Разрядность шин: 32 бит для команд и 24 бит для адреса;

2.5.5 Четвёртое поколение

Четвёртое поколение ЦСП характеризуется значительным расширением наборов команд, созданием [VLIW](#) и [суперскалярных](#) процессоров. Заметно возросли тактовые частоты. Так, например, время выполнения команды MAC ($Y := X + A \times B$) удалось сократить до 3 нс.

2.5.6 Современные ЦСП

Лучшие современные ЦСП можно характеризовать следующими параметрами:

- Тактовая частота — 1 ГГц и выше;
- Многоядерность;
- Наличие двухуровневого кеша;
- Встроенные многоканальные контроллеры прямого доступа к памяти;
- Быстродействие порядка нескольких тысяч [MIPS](#) и [MFLOPS](#);
- Выполнение до 8 параллельных инструкций за такт;
- Совместимость со стандартными шинами ([PCI](#) и др.)

2.6 Основные параметры ЦСП

- **Тип арифметики.** ЦСП делятся на процессоры, обрабатывающие данные с фиксированной точкой и обрабатывающие данные с плавающей точкой. Устройства с плавающей точкой удобнее в применении, но они заметно сложнее по устройству и более дороги;
- **Разрядность данных.** Большинство ЦСП с фиксированной точкой обрабатывают данные с разрядностью 16 бит, процессоры с плавающей точкой — 32 бита. Многие модели могут обрабатывать данные с двойной точностью.
- **Быстродействие.** Быстродействие как интегральную характеристику определить достаточно сложно, поэтому скорость работы характеризуют несколькими параметрами, а также временем решения некоторых реальных задач.
 - **Тактовая частота и Время командного цикла.** Для современных ЦСП тактовая внутренняя частота может отличаться от внешней, поэтому могут указываться два значения. Время командного цикла указывает на время выполнения одного этапа команды, то есть время одного цикла конвейера команд. Так как команды могут исполняться за разное количество циклов, а также с учетом возможности одновременного исполнения нескольких команд, этот параметр может характеризовать быстродействие ЦСП достаточно приближённо.
 - **Количество выполняемых команд за единицу времени.** Различное время исполнения команд, а также исполнение нескольких команд одновременно не позволяют использовать этот параметр для надежной характеристики быстродействия.
 - **Количество выполняемых операций за единицу времени (MIPS).** Данный параметр учитывает одновременную обработку нескольких команд и наличие параллельных вычислительных модулей, поэтому достаточно хорошо может указывать на быстродействие ЦСП. Некоторой проблемой здесь остается то, что понятие «операции» четко не формализовано.
 - **Количество выполняемых операций с плавающей точкой за единицу времени.** Параметр аналогичен предыдущему и используется для процессоров с плавающей точкой.
 - **Количество выполняемых операций MAC за единицу времени.** Данная команда, с одной стороны, является базовой для многих вычислений, а с другой — достаточно проста. Поэтому время ее исполнения можно использовать в том числе и для оценки общей производительности ЦСП.



- **Виды и объём внутренней памяти.** Объем внутренней оперативной памяти показывает, сколько данных ЦСП может обработать без обращения к внешней памяти, что может характеризовать общее быстродействие системы а также возможность работать «в реальном времени». Тип ПЗУ определяет возможности по программированию устройства. Модели с обычным ПЗУ подходят для крупносерийного производства, ППЗУ (однократно программируемое) удобно для небольших тиражей, а применение Flash-памяти позволяет менять программу устройства многократно во время эксплуатации. На данный момент не очень мощные DSP чаще всего снабжены достаточно большой Flash-памятью (ее цена неуклонно снижается) и заметным объемом RAM и поэтому могут являться самодостаточными без добавления внешней памяти и на этапе разработки и на этапе производства, что повышает конкурентоспособность таких DSP в ряде сегментов рынка. Мощные DSP как правило полагаются на внешнюю память подключенную по достаточно быстрым шинам а размещение там Flash-памяти может быть технически проблематичным, например, из-за заметного выделения тепла.
- **Адресуемый объем памяти.** Объем адресуемой внешней памяти характеризуется шириной внешней шины адреса.
- **Способ начальной загрузки.**
- **Количество и параметры портов ввода-вывода.** Данный параметр показывает возможности ЦСП по взаимодействию с внешними по отношению к нему устройствами.
- **Состав внутренних дополнительных устройств.** В число внутренних могут входить разнообразные по назначению устройства, например, *общего применения* — таймеры, контроллеры ПДП и т. д., а также *проблемно-ориентированные* — АЦП, кодеки, компрессоры данных и другие.
- **Напряжение питания и потребляемая мощность.** Данная характеристика особенно важна для ЦСП, встраиваемых в переносные устройства. Обычно предпочтительнее низковольтные устройства (1,8-3,3В), которые имеют быстродействие аналогично 5В процессорам, но заметно экономнее в плане потребления энергии. Многие устройства имеют режимы экономии при простое, либо позволяют программно отключать часть своих устройств.
- **Состав и функциональность средств разработки и поддержки.**
 - Перечень языков программирования, для которых есть компиляторы под данную систему;
 - Наличие и возможности средств отладки готовых программ;
 - Доступность документации и технической поддержки;
 - Наличие библиотек стандартных подпрограмм и математических функций;
 - Наличие, доступность и возможности совместимых устройств — АЦП, ЦАП, контроллеры питания и т. д.
- **Допустимые параметры окружающей среды.**
- Другие, в зависимости от назначения.

Часто используются также интегральные характеристики ЦСП, например показатель «мощность/ток/быстродействие», например mA/MIPS (миллиампер на 1 млн инструкций в секунду), что позволяет оценить реальную потребляемую мощность в зависимости от сложности задачи, решаемой процессором в указанный момент.

Выбор ЦСП целиком определяется назначением разрабатываемой системы. Например, для массовых мобильных устройств важна дешевизна процессора, низкое энергопотребление, в то время как стоимость разработки системы отходит на второй план. С другой стороны, для измерительного оборудования, систем обработки звуковой и видеоинформации важны эффективность процессора, наличие развитых инструментальных средств, многопроцессорность и т. д.



<u>Архитектура</u>	CISC · EDGE · EPIC · MISC · URISC · RISC · VLIW · ZISC · Фон Неймана · Гарвардская · 32-bit · 64-bit · 128-bit
<u>Параллелизм</u>	Pipeline Конвейер · In-Order & Out-of-Order execution · Переименование регистров · Speculative execution
	Уровни Бит · Инструкций · Суперскалярность · Данных · Задач
	Нити Multithreading · Simultaneous multithreading · Hyperthreading · Superthreading
	Классификация по Флину SISD · SIMD · MISD · MIMD
Реализации	DSP · GPU · SoC · PPU · Векторный процессор · Математический сопроцессор · Микропроцессор · Микроконтроллер
Компоненты	Barrel shifter · FPU · BSB · MMU · TLB · register file · control unit · АЛУ · Демультимплексор · Мультиплексор · Микрокод · Тактовая частота · Корпус · Регистры · Кэш
Управление питанием	APM · ACPI · Clock gating · Динамическое изменение частоты · Динамическое изменение напряжения

2.6.1 Оценка и сравнение производительности

Как отмечено ранее, отдельные характеристики типа тактовой частоты, MIPS, MOPS, MFLOPS позволяют оценить быстродействие ЦСП достаточно неоднозначно. Поэтому для решения задачи измерения и сравнения характеристик разных ЦСП используют специальные наборы тестов, имитирующих некоторые распространенные задачи цифровой обработки сигналов. Каждый тест состоит из нескольких небольших программ, которые пишутся на ассемблере и оптимизируются под заданную архитектуру. Эти тесты могут включать реализацию:

- Фильтры [КИХ](#) и [БИХ](#);
- Перемножение [векторов](#);
- [Декодеры Витерби](#);
- [БПФ](#)

Наиболее авторитетным пакетом тестов на сегодняшний день является тест BTDMark2000 (<http://www.btdi.com/>), который кроме указанных алгоритмов включает также оценку используемой алгоритмом памяти, время разработки системы и другие параметры.



3 Устройство (архитектура)

3.1.1 Гарвардская архитектура

ЦСП строятся на основе Гарвардской архитектуры

Цифровые сигнальные процессоры строятся на основе т. н. «Гарвардской архитектуры», отличительной особенностью которой является то, что программы и данные хранятся в различных устройствах памяти — памяти программ и памяти данных. В отличие от [архитектуры фон Неймана](#), где процессору для выборки команды и двух операндов требуется минимум три цикла шины, ЦСП может производить одновременные обращения как к памяти команд, так и к памяти данных, и указанная выше команда может быть получена за два цикла шины. В реальности, благодаря продуманности системы команд и другим мерам, это время может быть сокращено до одного цикла. В реальных устройствах память команд может хранить не только программы, но и данные. В этом случае говорят, что ЦСП построен по модифицированной гарвардской архитектуре.

Память команд и память данных обычно располагаются на кристалле ЦСП. В связи с тем, что эта память имеет относительно небольшой объём, возникает необходимость в использовании внешних (относительно кристалла процессора) запоминающих устройств. Для таких устройств отдельные шины команд и данных не используются, так это потребовало бы значительно увеличить количество внешних выводов кристалла, что дорого и непрактично. Поэтому взаимодействие ЦСП с внешними запоминающими устройствами происходит по одному комплекту шин без разделения на команды и данные. Следует также заметить, что обращение к внешней памяти всегда занимает значительно больше времени, чем к внутренней, поэтому в приложениях, критичных ко времени исполнения, такие обращения необходимо минимизировать.

3.2 Классификация ЦСП по архитектуре

Следует отметить, что приведенная ниже классификация достаточно условна, так как разнообразие технических решений зачастую не позволяет однозначно отнести каждое конкретное устройство к одному из указанных типов. Поэтому нижесказанное следует скорее использовать как материал для понимания особенностей архитектуры ЦСП, чем для реальной классификации каких-либо изделий.

Особенности архитектуры ЦСП удобно рассматривать на примере конкретного алгоритма цифровой обработки данных, например, [КИХ-фильтра](#), выходной сигнал которого можно записать как:

$$y(n) = \sum_{i=0}^P b_i x(n-i), \text{ где}$$

- $x(n-i)$ — отсчеты входного сигнала;
- b_i — коэффициенты фильтра.

Как можно легко заметить, вычисление результата является классическим примером использования операции «умножение с накоплением» — МАС ($Y := X + A \times B$).

3.2.1 Стандартные ЦСП

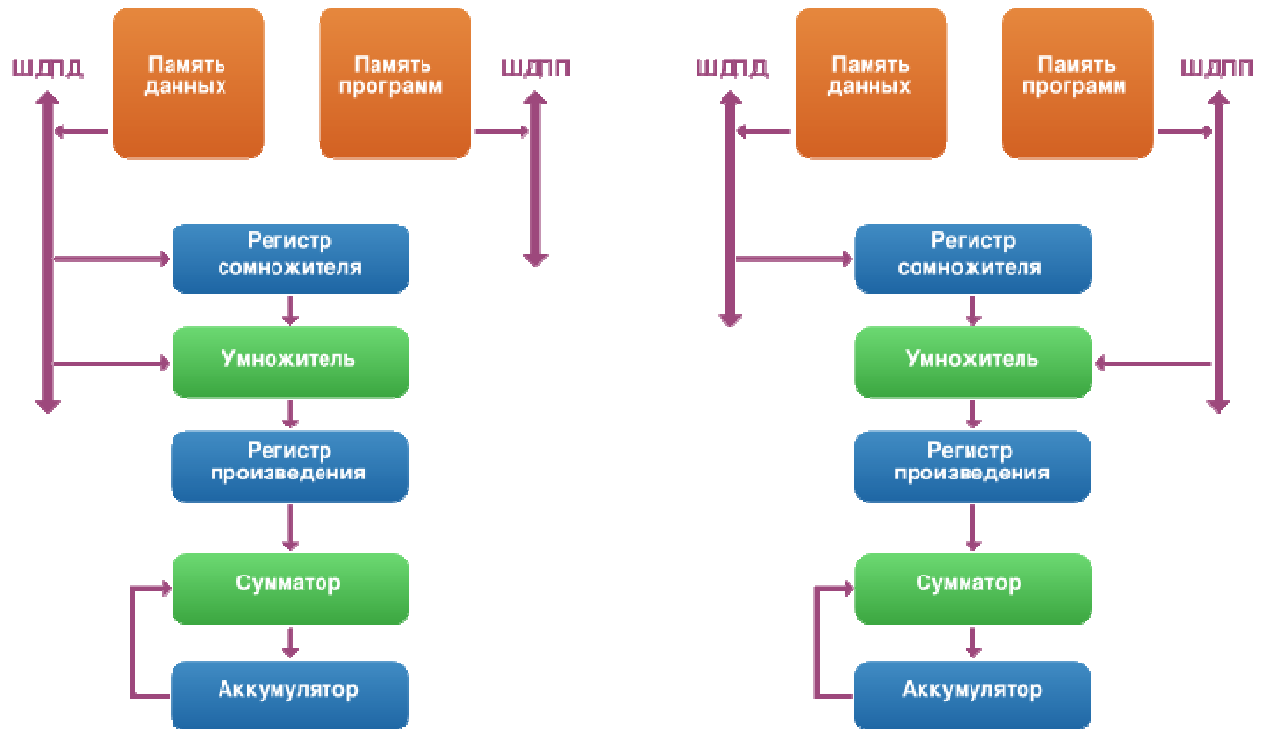


Рисунок Два варианта исполнения команды MAC на ЦСП Texas Instruments

На рисунке показано два варианта выполнения команды MAC на стандартном ЦСП. В первом варианте оба операнда хранятся в памяти данных, поэтому на их выборку требуется два такта, то есть время выполнения n сложений равно $2n$. Во втором случае один из операндов хранится в памяти программ, поэтому команда выполняется за один такт, и общее время выполнения цикла будет равно n тактов (следует уточнить, что в реальности для исполнения за один такт MAC должна исполняться внутри специальной команды цикла для исключения повторной выборки самого кода команды, что требует дополнительного такта). Здесь видно, что эффективная реализация алгоритма требует использования памяти программ для хранения данных, а максимальное быстродействие достигается только в однокомандном цикле.

Одним из вариантов, позволяющим отказаться от использования памяти программ для хранения данных, является применение т. н. «двухпортовой памяти», то есть памяти, имеющей два комплекта входных шин — двух шин адреса и данных. Такая архитектура позволяет произвести одновременное обращение по двум адресам (правда, при этом они должны находиться в разных адресуемых блоках). Данное решение применяется в ЦСП компаний Motorola (DSP56000) и Lucent (DSP1600).

При указанной архитектуре повысить производительность можно только увеличением тактовой частоты.

3.2.2 Улучшенные стандартные ЦСП

«Улучшенные стандартные ЦСП» для повышения производительности системы, по сравнению со стандартными ЦСП, используют следующие методы повышения параллелизма:

- Увеличение количества операционных и вычислительных устройств;
- Введение специализированных сопроцессоров;
- Расширение шин для увеличения количества передаваемых данных;

- Использование памяти с многократным доступом (несколько обращений за такт);
- Усложнение системы команд;

Многие из этих способов применялись уже начиная с самых первых процессоров, поэтому зачастую их невозможно однозначно классифицировать как «стандартные» или «улучшенные».

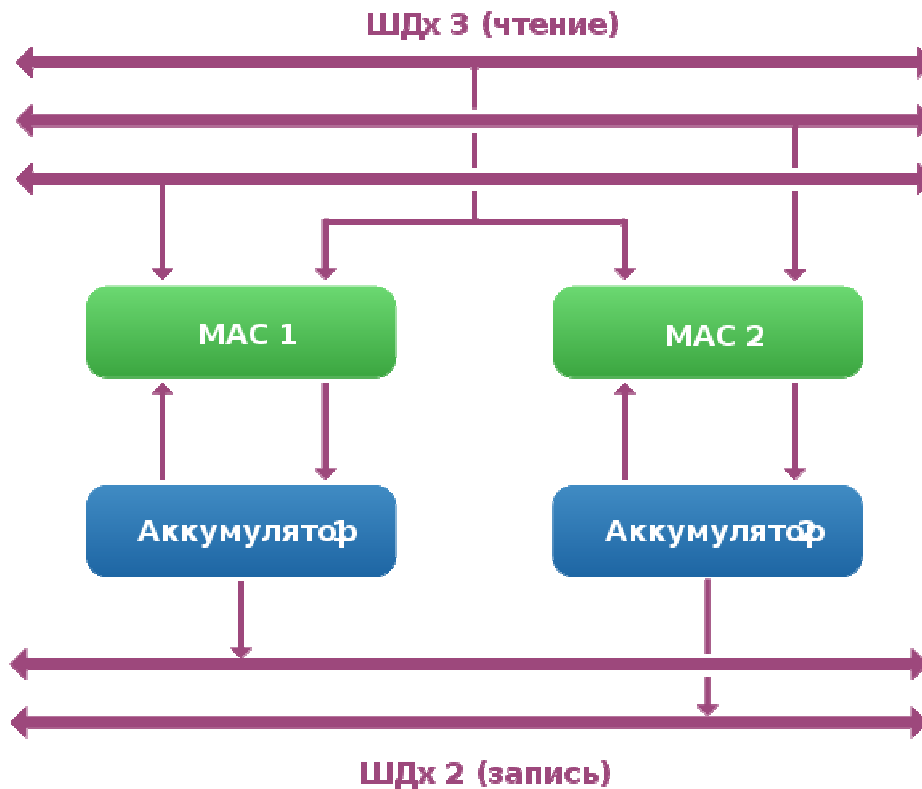


Рисунок Исполнение двух операций MAC на ЦСП Texas Instruments TMS320C55x

На рисунке показан пример реализации вычисления двух параллельных команд MAC. Для этого в ЦСП присутствуют два модуля MAC и два аккумулятора. Блоки MAC получают данные по трём шинам одновременно, причём одно из значений является для них общим. Таким образом, происходит одновременное исполнение двух команд:

- $AK1 := AK1 + D1 \times D2$
- $AK2 := AK2 + D1 \times D3$

Особенность показанного решения состоит в том, что к выполнению двух параллельных команд с одним общим сомножителем можно свести многие алгоритмы ЦОС, например:

- КИХ-фильтр с симметричными коэффициентами. В качестве одинаковых сомножителей используются коэффициенты фильтра, а на отдельные шины подаются два разных набора отсчетов сигнала, то есть параллельно рассчитываются две половины фильтра, которые затем суммируются.
- Двухканальная обработка. На общую шину подаются отсчеты сигнала, а на отдельные — наборы коэффициентов.

В некоторых процессорах (Lucent DSP16xxx, ADI ASDP-2116x) используются два одинаковых ядра, каждый со своей памятью, то есть одна команда выполняется одновременно в двух ядрах с



различными данными. Это позволяет обойти ограничение на использование полностью независимых данных.

Характерным недостатком таких процессоров можно считать необходимость в высокой квалификации разработчика, так как эффективное использование указанных особенностей требует программирования на языке ассемблера, хорошего знания архитектуры и системы команд, то есть эти устройства считаются «недружественными» к языкам высокого уровня.

3.3 ЦСП с архитектурой VLIW

Основное отличие VLIW-процессоров состоит в том, что коды команд еще на этапе компиляции собираются в большие «суперкоманды» и выполняются параллельно. Обычно такие процессоры используют RISC-архитектуру с фиксированной длиной команды, где каждая из них выполняется в отдельном операционном модуле. К характерным особенностям таких процессоров можно отнести:

- Большой набор операционных модулей, работающих независимо друг от друга. В состав таких модулей могут входить:
 - Арифметические:
 - Модули арифметических операций и операций сравнения;
 - Модули логических операций;
 - Модули умножения чисел с плавающей и фиксированной точкой;
 - Модули генерации констант.
 - Модули генерации адреса, в том числе для линейных и циклических буферов;
- Необходимость оптимизации компилятора под каждую модель процессора, так как между моделями может меняться состав и функции вычислительных блоков, что влечет за собой изменение перечня команд, которые могут выполняться одновременно;
- Необходимость в наличии сверхшироких шин данных (порядка 128 бит), чтобы код операции, состоящий из отдельных команд (до 8-ми) мог быть получен из памяти за одно обращение.
- Высокие требования к объему памяти программ, что также связано с большой длиной операции.

Обычно, если процессор имеет несколько одинаковых модулей, то при создании программы на ассемблере имеется возможность указания только типа необходимого операционного модуля, а конкретное устройство будет назначено компилятором. С одной стороны, это упрощает программирование таких устройств, а с другой стороны, позволяет достаточно эффективно использовать их ресурсы.

3.4 Суперскалярные ЦСП

Суперскалярные процессоры также характеризуются большим набором параллельных операционных модулей и возможностью одновременного исполнения нескольких команд. Однако, по сравнению с VLIW, они имеют две характерные особенности:

- Команды процессора не группируются в блоки, каждая из них поступает в процессор независимо;
- Команды для параллельного исполнения группируются внутри процессора на основе состава и текущей загруженности операционных блоков, а также зависимости между данными.

С помощью описанного подхода можно обойти следующие недостатки VLIW:

- Неэффективное использование памяти из-за большой длины групповой операции;
- Зависимость скомпилированного кода от состава операционных модулей конкретного процессора.

Платой за решение этих проблем становится значительное усложнение схемы процессора, в котором появляется модуль планирования выполнения команд.

Суперскалярные процессоры планируют исполнение команд не только на основе информации о загрузке операционных блоков, но и на основе анализа зависимостей между данными. К примеру, команда сохранения результата арифметической операции не может быть выполнена раньше самой операции вычисления, даже если модуль обращения к памяти в данный момент свободен. Эта особенность приводит в том числе к тому, что один и тот же набор команд может по-разному исполняться в различных местах программы, что делает невозможным точную оценку производительности. Особенно это важно для систем, работающих в реальном времени, ведь оценка по наихудшему результату приведет к тому, что ресурсы процессора будут использованы не полностью. Таким образом, в этих системах задача точной оценки производительности суперскалярных ЦСП остается открытой.

3.5 Гибридные ЦСП

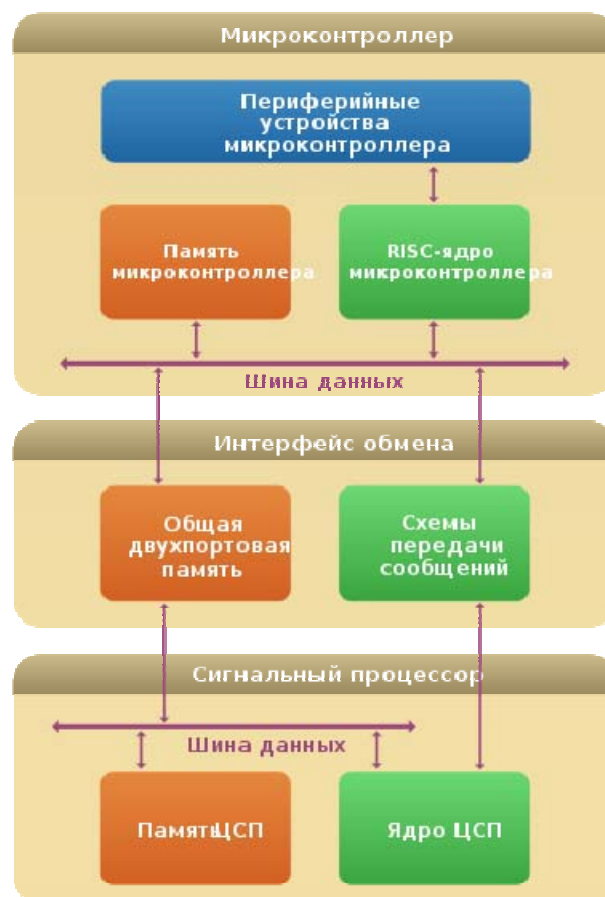


Рисунок Блок-схема гибридного ЦСП

Под гибридными ЦСП обычно понимают специализированные устройства, сочетающие в себе функцию [микроконтроллера](#) и цифрового сигнального процессора. Обычно такие изделия предназначены для выполнения одной функции — например, управления электрическими двигателями, или другими объектами в реальном времени. Другой широкой областью их применения в последнее время становится мобильная телефония, где ранее использовались два процессора — один обычный для управления функциями аппарата (дисплеем, клавиатурой), а другой для обработки голосовых сигналов (кодирование и т. д.).



3.6 Классификация ЦСП по назначению

В целом, по назначению ЦСП можно разделить на две группы:

- ЦСП общего назначения;
- Проблемно-ориентированные ЦСП.

«Проблемная ориентация» обычно относится не к дополнительным командам, а к набору встроенных специализированных периферийных устройств. Например, ЦСП, предназначенные для управления [электродвигателями](#), могут содержать на кристалле генераторы сигналов [ШИМ](#), контроллеры локальной промышленной сети и т. д. Процессоры, используемые для обработки голосовых сигналов, часто содержат модули манипуляции разрядами (VMU) и сопроцессоры исправления ошибок. В цифровых фото- и видеокамерах применяются ЦСП с модулями кодирования/декодирования MPEG1, MPEG4, JPG, MP3, AAC и др.

3.7 Программирование ЦСП

Для программирования ЦСП обычно используют один из двух языков — [ассемблер](#) и [С](#). Основные особенности ассемблеров ЦСП совпадают с аналогичными языками обычных [микропроцессоров](#) и, в целом, могут быть описаны как:

- Язык ассемблера является машинно-ориентированным, то есть каждое семейство процессоров имеет язык, отличающийся от языка других семейств;
- Одна команда на ассемблере обычно эквивалентна одной команде машинного языка;
- При программировании на ассемблере программисту доступны все ресурсы процессора и системы, что позволяет использовать их максимально эффективно;
- От программиста требуется хорошее знание архитектуры каждого конкретного процессора, с которым он работает, то есть требуемая квалификация персонала должна быть достаточно высокой;
- Создание и отладка программ на ассемблере — длительный трудоёмкий процесс, также требующий высокой квалификации.

С другой стороны, при использовании языков высокого уровня, в частности, С, можно заметно упростить и ускорить создание программ, но при этом ресурсы системы будут использоваться менее эффективно, по сравнению с программой, целиком написанной на ассемблере.

В реальности обычно используются подход, совмещающий достоинства как языков высокого уровня, так и эффективности программ на ассемблере. Выражается это в том, что стандартные библиотеки обычно создаются на ассемблере, как и критичные ко времени исполнения и объёму памяти части кода. В то же время вспомогательные модули могут создаваться на языке высокого уровня, ускоряя и упрощая разработку программной системы в целом.

3.7.1 Особенности ассемблеров ЦСП

К интересным особенностям ассемблеров ЦСП можно отнести следующее:

- Наличие двух форм записи многих команд — *мнемонической* и *алгебраической*. Мнемоническая форма аналогична записи команд для обычных микропроцессоров, например, `ADD dst, src`. Другая, алгебраическая, в ассемблерах стандартных микропроцессоров используется реже, в то время как на языке ЦСП упомянутая команда может быть записана в виде `dst = dst + src`. Обычно ассемблеры ЦСП понимают обе формы записи, но, например, ассемблеры ADI и Lucent используют только алгебраическую запись.



- Средства организации стандартных структур, например, специальных аппаратных команд повторения одной команды или блока кода. При этом, в отличие от команд повторения обычных процессоров, ЦСП может пропускать цикл выборки кода повторяемой команды, что уменьшает время выполнения каждого повторения как минимум на 1 цикл шины, что при двухцикловой команде даёт двойной выигрыш по времени.

3.7.2 Совместимость внутри семейств ЦСП

Обычно ЦСП выпускаются семействами, и изделия внутри семейств имеют аналогичные языки ассемблера, или даже совместимы на уровне машинных кодов. Также внутри семейства обычно используются одинаковые наборы библиотек подпрограмм. Как и в обычных микропроцессорах, зачастую старшие модели ЦСП могут исполнять машинный код младших моделей, либо их ассемблер включает все команды младших моделей как подмножество собственного набора команд.

3.7.3 Отладка программ

Обычно, отладка программ, написанных для ЦСП, производится с помощью специальных средств, включающих программные имитаторы и [эмуляторы](#). В их состав также часто входят средства профилирования (измерения скорости выполнения блоков кода).

4 Основные производители DSP

4.1 Ссылки на основных производителей

- [Texas Instruments\(США\)](#)
- [Analog Devices\(США\)](#)
- [Atmel Corp.\(США\)](#)
- [MicroChip\(США\)](#)
- [Freescale \(бывш. Motorola DSP\)\(США\)](#)
- [NEC Electronics Corporation\(Япония\)](#)
- [Renesas \(объединение отделений Mitsubishi Electric Corporation и Hitachi\)\(Япония\)](#)
- [ZiLOG, Inc.](#)
- [ГУП НПП «ЭЛВИС» \(РФ\)](#)
- [ГП «НИИ Электронной техники» \(РФ\)](#)
- [STMicroelectronics \(ЕС\)](#)
- [NXP \(основана Philips\)](#)
- [DSP «Stratix» и программируемые в ПЛИС DSP «NIOS» от Altera Corporation\(США\)](#)

4.2 DSP от Microchip dsPIC30F и dsPIC33F

Компания Microchip предлагает два семейства 16-ти разрядных Flash микроконтроллеров с поддержкой команд цифровой обработки сигналов — dsPIC30F и dsPIC33F. Высокое быстродействие (30 MIPS для dsPIC30F и 40 MIPS для dsPIC33F) и эффективная система команд позволяет использовать контроллеры в сложных системах реального времени. Ключевые особенности:

- расширенная система команд, включающая специфические команды поддержки цифровой обработки сигналов (DSP).
- 24-разрядные инструкции выполняются за 4 периода тактовой частоты у dsPIC30F и за 2 — у dsPIC33F, за исключением команд деления, переходов, команд пересылки данных из регистра в регистр и табличных команд.



- разрядность программного счетчика (24 бита) позволяет адресовать до 4М слов программной памяти (4М*24бит).
- аппаратная поддержка циклов типа DO и REPEAT, выполнение которых не требует дополнительных издержек программной памяти и времени на анализ условий окончания, в то же время эти циклы могут быть прерваны событиями прерывания в любой момент;
- 16 рабочих регистров, каждый регистр массива может выступать как данные, адрес или смещение адреса
- два класса команд: микроконтроллерные инструкции (MCU) и команды цифровой обработки сигналов (DSP). Оба этих класса равноправно интегрированы в архитектуру контроллера и обрабатываются одним ядром.
- различные типы адресации;
- система команд оптимизирована для получения максимальной эффективности при программировании на языке высокого уровня Си.

Если о PIC24F можно говорить, как об усеченном доработанном варианте dsPIC30F (без ядра ЦОС, с трехвольтовым питанием и переработанным конвейером), то PIC24H — это усеченный вариант dsPIC33F. Хотя в данном сравнении нарушены причинно-следственные связи, технически оно верно. Ядро dsPIC33F полностью аналогично ядру dsPIC30F, за исключение того что в dsPIC33F команда выполняется за два такта генератора. Семейства полностью совместимы по набору инструкций, программной модели и способам адресации, что позволяет использовать библиотеки и исходные коды программ, написанные для dsPIC30F. Особо следует отметить переработанную по сравнению с dsPIC30F систему тактирования. dsPIC33F, как и семейство PIC24H, имеют PLL с дробным коэффициентом умножения (конфигурируемым программно), что позволяет получить сетку частот от 12,5 МГц до 80 МГц с шагом 0,25 МГц при использовании кварцевого резонатора 4 МГц. Кроме того, контроллеры dsPIC33F и PIC24H имеют два внутренних высокостабильных RC-генератора с частотами 7,3728 МГц и 32,768 кГц. Отдельный делитель тактовой частоты ядра (модуль DOZE) присутствует во всех новых 16-битных семействах. Он позволяет уменьшить тактовую частоту, подаваемую на ядро независимо от тактовой частоты периферийных модулей, что необходимо для уменьшения потребления в энергоограниченных приложениях. Большой выбор по периферии контроллеров ЦОС:

Общего назначения:

- календарь и часы реального времени RTCC
- аппаратный подсчет CRC
- расширенная периферия (SPI, I2C, UART (с поддержкой IrDA, LIN), CAN (ECAN), USB OTG)
- 10-и и 12-битные АЦП
- компараторы
- 10-и и 16-битные ЦАП
- прямой доступ к памяти (DMA)
- ведущий параллельный порт (PMP)
- программное переназначение выводов (PPS)
- многоуровневая система защиты кода (Code Guard)

Для управления двигателями и преобразователями энергии

- специализированный ШИМ для управления приводом (Motor Control PWM)
- интерфейс квадратурного энкодера

Для импульсных источников питания (SMPS)

- Специализированный сверхбыстрый ШИМ с высоким разрешением (SMPS PWM)
- Специализированные сверхбыстрые АЦП (SMPS ADC)

Для работы со звуком:



- 12-битный АЦП
- 16-битный ЦАП
- специализированный ШИМ (output compare PWM)
- интерфейс кодирования данных DCI (I2S, AC97)

Для управления графическими дисплеями:

- ведущий параллельный порт PMP (QVGA)
- модуль измерения времени заряда CTMU (сенсорные дисплеи touch-screen)



•

4.3 Отечественные ЦСП

4.3.1 ГУП НПЦ «Элвис»

Параметры отечественных ЦСП фирмы ЭЛВИС смотри в отдельном файле- Каталог Элвис

Таблица 1. Цифровые сигнальные процессоры "Мультикор"

Микросхема	МС-12	МС-24	МС-0226	NVCom-01	МС-0428*)
Технология изготовления, мкм	0.25	0.25	0.25	0.13	0.18/0.13
Размер кристалла, мм*мм	10*10	10*10	12.3 *12.6	8.8*9.5	14*14 / -
Интеграция, млн. транзисторов	~18	~18	~26	~60	~65
Корпус	PQFP240	HSBGA292	HSBGA416	HSBGA400	Уточняется
Многопроцессорная MIMD - архитектура	2 процессора: RISCore32 + ELcore-14	2 процессора: RISCore32 + ELcore-24	3 процессора: RISCore32 + 2 x ELcore-26	3 процессора, RISCore32 (с FPU) + 2 x DELCore-30	5 процессоров, уточняется
Рабочая частота	80	80	100	300	250-340 / -
Пиковая производительность, MFLOPs, 32 бит	240	480	1200	3600	не менее 8000
Год выпуска	2004	2004	2006	2008	-

*) перспективная серия

В качестве процессорных блоков используются следующие типы IP-ядер из библиотеки платформы "МУЛЬТИКОР":

- процессорные RISC - ядра с архитектурой MIPS32, выполняющие функции центрального процессора системы CPU (Central Processing Unit);
- высокопроизводительные ядра процессоров-акселераторов для цифровой обработки сигналов (DSP - Digital Signal Processing) с плавающей/фиксированной точкой ELcore-xx (ELcore = Elvees's core).

Таблица 2. Производительность DSP-ядер процессоров "Мультикор"

Компания	ЭЛВИС				TI		ADI	
Процессор	MC-12	MC-24	MC-0226	NVCom-01	C6701	C6416	TS201	BF53
Технология, нм	250	250	250	130	180	130	130	130
Тактовая частота, МГц	80	80	100	300	166	600	600	600
Пиковая производительность по смеси арифметических операций (умножения, сложения, вычитания)								
• 16-разр. фиксированная точка, млн.оп/с	640	1280	3200	19200	-	4800	14400	3360
• 32-разр. плавающая точка, млн. флоп/с	240	480	1200	3600	1000	-	3600	-
• 32-разр. фиксированная точка, млн. оп/с	320	640	1600	4800	-	-	-	-
• 8-разр. фиксированная точка, млн. оп/с	1440	2880	7200	28800	-	-	-	4800
Пиковая производительность по операции MAC (умножение + накопление)								
• 16-разр. фиксированная точка, млн. оп/с	160	320	800	4800	-	-	4800	1200
• 32-разр. плавающая точка, млн. флоп/с	80	160	400	1200	-	-	-	-
• 32-разр. фиксированная точка, млн. оп/с	80	160	400	1200	-	-	1200	-
• 1-разр. фиксированная точка, млн.оп/с	1700	3400	8500	25600	-	-	32000	-
Пиковая производительность по операции CMAC (комплексное умножение+накопление)								
• 16-разр. фиксированная точка, млн. оп/с	-	-	-	1200	-	-	-	-
• 8-разр. фиксированная точка, млн. оп/с	160	320	800	2400	-	-	-	-
КИХ - фильтр, действительные данные и коэффициенты, 35 отводов, 1024 входных данных, без прореживания								
• формат 16•16+32/64, фиксированная точка, мкс	275	192	77	10	-	25,7	-	-
КИХ - фильтр, действительные данные и коэффициенты, 50 отводов, 1024 входных данных, без прореживания								
• формат 16•16+32/64, фиксированная точка, мкс	363	243	97	13,4	-	-	12	44
БПФ - 1024, комплексные данные								
• формат (16+j16), блочная плавающая точка, мкс	145	72	29	4,6	-	10	5	-
• формат (32+j32), плавающая точка, мкс	273	136	55	17,9	160	-	16,8	-
БПФ - 256, комплексные данные								
• формат (16+j16), блочная плавающая точка, мкс	28,9	14,5	5,8	1	-	-	1,5	-
Операция ACS (сложение+сравнение+выбор) – базовая операция декодера Витерби								
• 16-разр. метрики путей, нс/метрика	12,5	6,3	2,5	1,3	-	0,82	-	-

Архитектура микросхем "Мультикор" по организации потоков данных и инструкций поддерживает пиковую производительность на большинстве задач обработки сигналов/изображений реального времени. Это обеспечивается малым количеством стадий процессорного конвейера, а также наличием многоканального интеллектуального контроллера DMA с поддержкой режимов самосинхронизации ресурсов микросхемы и 32/64-разрядного по данным порта внешней памяти со встроенным SDRAM/FLASH/SRAM/ROM контроллером.

Микросхемы в зависимости от модели содержат от 2 до 8 Мбит внутренней памяти, периферийные SHARC-совместимые линки и последовательные порты, I2C, I2S, USB, Ethernet, PCI, UART, JTAG, а также гиперлинки типа SpaceWire и Serial RapidIO.



Важным преимуществом отечественных микросхем является плотность кодов для DSP-ядер, которая характеризуется длиной программ для той или иной функции. Для микросхем серии "Мультикор" данная характеристика, как правило, лучше, чем у зарубежных аналогов (см. табл.3).

Таблица 3. Размер программ для DSP процессоров "Мультикор" (на примере MC-24) по некоторым типовым процедурам обработки

Процедура	Размер*)	Дополнительные условия
КИХ-фильтр	10 - 15	32-разрядный формат плавающей точки, действительные данные и коэффициенты
FFT	150 - 230	32-разрядный формат плавающей точки, вход: комплексные данные, произвольная степень 2, выход: комплексная амплитуда/мощность, порядок 2-инверсный/прямой
DCT 8x8	50 - 80	16-разрядный целочисленный формат (блочная плавающая точка) или 32-разрядный формат плавающей точки
SIN/COS	30	32-разрядный формат плавающей точки

4.3.2 ГП «НИИЭТ»

- M1867BM1 [КМОП однокристалльный 16-разрядный процессор ЦОС с фиксированной запятой производительностью 5 млн. опер/сек](#)
- L1867BM2 [КМОП однокристалльный 16-разрядный процессор ЦОС с фиксированной запятой производительностью 10 млн.опер/сек](#)
- L1867BM3 [КМОП однокристалльный 32-разрядный процессор ЦОС с плавающей запятой производительностью 27млн. опер/сек](#)

5 Алгоритмы ЦОС

5.1 Ких фильтр

5.1.1 Динамические характеристики

[Разностное уравнение](#), описывающее связь между входным и выходным сигналами фильтра: $y(n) = b_0x(n) + b_1x(n-1) + \dots + b_Px(n-P)$ где P — порядок фильтра, $x(n)$ — входной сигнал, $y(n)$ — выходной сигнал, а b_i — коэффициенты фильтра. Иными словами, значение любого отсчета выходного сигнала определяется суммой масштабированных значений P предыдущих отсчетов. Можно сказать иначе: значение выхода фильтра в любой момент времени есть значение отклика на мгновенное значение входа и сумма всех постепенно затухающих откликов P предыдущих отсчетов сигнала, которые всё ещё оказывают влияние на выход (после P -отсчетов [импульсная переходная функция](#) становится равной нулю, как уже было сказано, поэтому все члены после P -го тоже станут равными нулю). Запишем предыдущее уравнение в более ёмком виде:

$$y(n) = \sum_{i=0}^P b_i x(n-i)$$



Для того, чтобы найти [ядро фильтра](#) положим

$$x(n) = \delta(n)$$

где $\delta(n)$ — [дельта-функция](#). Тогда импульсная характеристика КИХ-фильтра может быть записана как:

$$h(n) = \sum_{i=0}^P b_i \delta(n-i)$$

[Z-преобразование](#) импульсной характеристики даёт нам передаточную функцию КИХ-фильтра:

$$H(z) = \sum_{i=0}^P b_i z^{-i}$$

5.1.2 Свойства

КИХ-фильтр обладает рядом полезных свойств, из-за которых он иногда более предпочтителен в использовании, чем [БИХ-фильтр](#). Вот некоторые из них:

- КИХ-фильтры устойчивы.
- КИХ-фильтры при реализации не требуют наличия обратной связи.
- Фаза КИХ-фильтров может быть сделана линейной

5.2 БИХ фильтр

Фильтр с бесконечной импульсной характеристикой (Рекурсивный фильтр, БИХ-фильтр) — линейный [электронный](#) фильтр, использующий один или более своих выходов в качестве входа, то есть образует [обратную связь](#). Основным свойством таких фильтров является то, что их [импульсная переходная характеристика](#) имеет бесконечную длину во временной области, а [передаточная функция](#) имеет дробно-рациональный вид. Такие фильтры могут быть как [аналоговыми](#) так и [цифровыми](#).

Примерами БИХ-фильтров являются [фильтр Чебышева](#), [фильтр Баттерворта](#), [Фильтр Калмана](#) и [фильтр Бесселя](#).

5.2.1 Динамические характеристики

[Разностное уравнение](#), описывающее дискретный БИХ-фильтр, устанавливает связь между входным и выходным сигналами во временной области:

$$y(n) = b_0 x(n) + b_1 x(n-1) + \dots + b_P x(n-P) - a_1 y(n-1) - a_2 y(n-2) - \dots - a_Q y(n-Q)$$

где P порядок входного сигнала, b_i — коэффициенты входного сигнала, Q — порядок обратной связи, a_i — коэффициенты обратной связи, $x(n)$ — входной, а $y(n)$ — выходной сигналы. Более компактная запись разностного уравнения:

$$y(n) = \sum_{i=0}^P b_i x(n-i) - \sum_{k=1}^Q a_k y(n-k)$$



Для того, чтобы найти [ядро фильтра](#), положим

$$x(n) = \delta(n)$$

где $\delta(n)$ — [дельта-функция](#). Тогда импульсная переходная функция (ядро фильтра) записывается как

$$h(n) = \sum_{i=0}^P b_i \delta(n-i) + \sum_{k=1}^Q a_k h(n-k)$$

[Z-преобразование](#) импульсной переходной функции даёт передаточную функцию БИХ-фильтра:

$$H(z) = \frac{\sum_{i=0}^P b_i z^{-i}}{1 + \sum_{k=1}^Q a_k z^{-k}}$$

5.3 БПФ

Быстрое преобразование Фурье (БПФ) — это быстрый алгоритм вычисления дискретного преобразования Фурье (см. [Дискретное преобразование Фурье](#), ДПФ). То есть алгоритм вычисления за количество действий, меньше чем $O(N^2)$, требуемых для прямого (по формуле) вычисления ДПФ. Иногда под БПФ понимается один из быстрых алгоритмов, называемым алгоритмом прореживания по частоте/времени или алгоритмом по основанию 2, имеющего сложность $O(N \log(N))$.

5.3.1 Основной алгоритм

Покажем как выполнить дискретное преобразование Фурье за $O(N(p_1 + \dots + p_n))$ действий при $N = p_1 p_2 \dots p_n$. В частности, при $N = 2^n$ понадобится $O(N \log(N))$ действий.

Дискретное преобразование Фурье преобразует набор чисел a_0, \dots, a_{n-1} в набор чисел

b_0, \dots, b_{n-1} , такой, что $b_i = \sum_{j=0}^{n-1} a_j \varepsilon^{ij}$, где $\varepsilon^n = 1$ и $\varepsilon^k \neq 1$ при $0 < k < n$. Алгоритм быстрого преобразования Фурье применим к любым коммутативным ассоциативным [кольцам](#) с единицей.

Чаще всего этот алгоритм применяют к полю [комплексных чисел](#) (с $\varepsilon = e^{2\pi i/n}$) и к [кольцам вычетов](#).

Основной шаг алгоритма состоит в сведении задачи для N чисел к задаче для $p = N/q$ чисел, где q — делитель N . Пусть мы уже умеем решать задачу для N/q чисел. Применим преобразование Фурье к наборам $a_i, a_{q+i}, \dots, a_{q(p-1)+i}$ для $i = 0, 1, \dots, q-1$. Покажем теперь, как за

$$b_i = \sum_{j=0}^{q-1} \varepsilon^{ij} \left(\sum_{k=0}^{p-1} a_{kq+j} \varepsilon^{kq} \right)$$

$O(Np)$ действий решить исходную задачу. Заметим, что $\varepsilon^{kq} = \varepsilon^{k \cdot (N/q)}$. Выражения в скобках нам уже известны — это $i \pmod{p}$ -тое число после преобразования Фурье j -той группы. Таким образом, для вычисления каждого b_i нужно $O(q)$ действий, а для вычисления всех b_i — $O(Nq)$ действий, что и требовалось получить.

5.3.2 Обратное преобразование Фурье



Для обратного преобразования Фурье можно применять алгоритм прямого преобразования Фурье — нужно лишь использовать ε^{-1} вместо ε (или применить операцию комплексного сопряжения в начале к входным данным, а затем к результату полученному после прямого преобразования Фурье) и окончательный результат поделить на N .

5.3.3 Общий случай

Общий случай может быть сведён к предыдущему. Пусть $4N > 2^k \geq 2N$. Заметим, что

$$b_i = \varepsilon^{-i^2/2} \sum_{j=0}^{N-1} \varepsilon^{(i+j)^2/2} \varepsilon^{-j^2/2} a_j$$

. Обозначим

$$\bar{a}_i = \varepsilon^{-i^2/2} a_i, \bar{b}_i = \varepsilon^{i^2/2} b_i, c_i = \varepsilon^{(2N-2-i)^2/2}. \quad \bar{b}_i = \sum_{j=0}^{2N-2-i} \bar{a}_j c_{2N-2-i-j}, \quad \text{Тогда}$$

положить $\bar{a}_i = 0$ при $i \geq N$. , если

Таким образом задача сведена к вычислению [свёртки](#), но это можно сделать с помощью трёх преобразований Фурье для 2^k элементов. Выполняем прямое преобразование Фурье для $\{\bar{a}_i\}_{i=0}^{i=2^k-1}$ и $\{c_i\}_{i=0}^{i=2^k-1}$, перемножаем поэлементно результаты и выполняем обратное преобразование Фурье.

Вычисления всех \bar{a}_i и c_i требуют $O(N)$ действий, три преобразования Фурье требуют $O(M \log(N))$ действий, перемножение результатов преобразований Фурье требует $O(N)$ действий, вычисление всех b_i зная значения свертки требует $O(N)$ действий. Итого для дискретного преобразования Фурье требуется $O(M \log(N))$ действий для любого N .

Этот алгоритм быстрого преобразования Фурье может работать над кольцом только когда известны [первообразные корни](#) из единицы степеней $2N$ и 2^k .

5.3.4 Вывод преобразования из ДПФ

Дискретное преобразование Фурье для вектора \vec{x} состоящего из N элементов имеет вид:

$$\vec{X} = \hat{A} \vec{x}$$

элементы матрицы \hat{A} имеют вид:

$$a_N^{mn} = \exp\left(-2\pi i \frac{mn}{N}\right)$$

Пусть N чётно, тогда ДПФ можно переписать следующим образом:

$$X_m = \sum_{n=0}^{N-1} x_n a_N^{mn} = \sum_{n=0}^{N/2-1} x_{2n} a_N^{2nm} + \sum_{n=0}^{N/2-1} x_{2n+1} a_N^{(2n+1)m}$$

Коэффициенты a_N^{2nm} и $a_N^{(2n+1)m}$ можно переписать следующим образом ($M=N/2$):

$$a_N^{2nm} = \exp\left(-2\pi i \frac{2mn}{N}\right) = \exp\left(-2\pi i \frac{mn}{N/2}\right) = a_M^{nm}$$

$$a_N^{(2n+1)m} = \exp\left(-2\pi i \frac{m}{N}\right) a_M^{nm}$$



В результате получаем:

$$X_m = \sum_{n=0}^{M-1} x_{2n} a_M^{nm} + \exp\left(-2\pi i \frac{m}{N}\right) \sum_{n=0}^{M-1} x_{2n+1} a_M^{nm}$$

То есть дискретное преобразование Фурье от вектора состоящего из N отсчетов свелось к

линейной композиции двух ДПФ от $\frac{N}{2}$ отсчетов, и если для первоначальной задачи требовалось N^2

операций, то для полученной композиции — $\frac{N^2}{2}$. Если M четно, то это разделение можно продолжать рекурсивно до тех пор, пока не дойдем до двух точечного преобразования Фурье, которое вычисляется по следующим формулам:

$$\begin{cases} X_0 = x_0 + x_1 \\ X_1 = x_0 - x_1 \end{cases}$$