

Способы реализации алгоритмов ЦОС

Перцев Дмитрий Юрьевич
доцент кафедры ЭВМ БГУИР

2022

Способы реализации алгоритмов ЦОС

Независимо от сложности алгоритма вычисления осуществляются с помощью **базовых операций**: сложения, вычитания и умножения.

Поскольку вычислительные операции производятся с данными, задерживаемыми относительно друг друга на один и более периодов дискретизации T с помощью **элементов задержки**, представляющих собой регистры (ячейки память), объединяемые в линии задержки, необходимо иметь возможность осуществлять пересылки и сдвиги данных.

Кроме того, для управления вычислительным процессом необходимо предусмотреть и **логические операции**.

Этого достаточно для построения алгоритма.

Реальное время

Автономная обработка (off-line) и обработка в реальном времени (real-time).

Автономная обработка: входной сигнал целиком находится в компьютере, вся информация доступная для программы обработки.

- колебания земной поверхности во время землетрясения;
- медицинское обследование (компьютерная томография).

При обработке в реальном времени выходной сигнал формируется в то же время, как в системы поступает входной сигнал.

- телефонная связь, слуховые аппараты, радары и др. (информация должна быть доступна незамедлительно).

Реальное время

Определяющим свойством процедуры становится ее практическая вычислимость, т.е. ее способность вычислять отсчет $y(n)$ за разумное время, или, за **реальное время**, при этом имеется в виду обязательное достижение заданной точности.

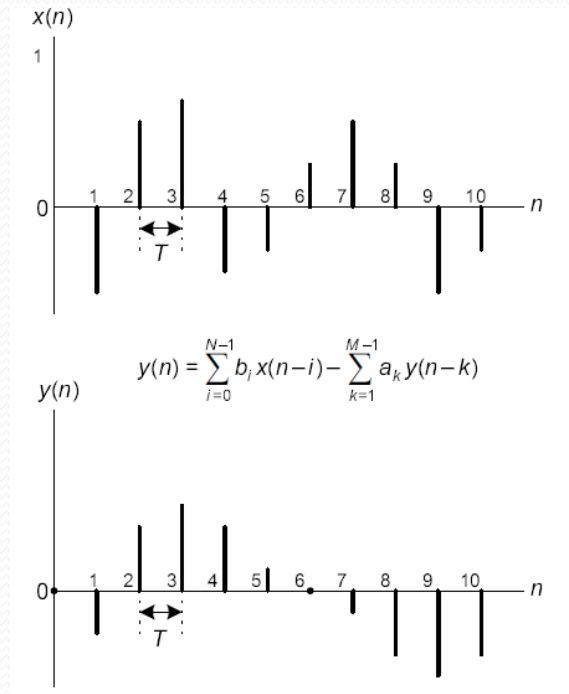
Определение **реального времени** **зависит от** конкретной задачи и **связано с** объемом вычислений алгоритма, точностью вычислений и частотой дискретизации (периода дискретизации).

Реальное время

Пусть

T – период дискретизации,

τ_a – время выполнения алгоритма.



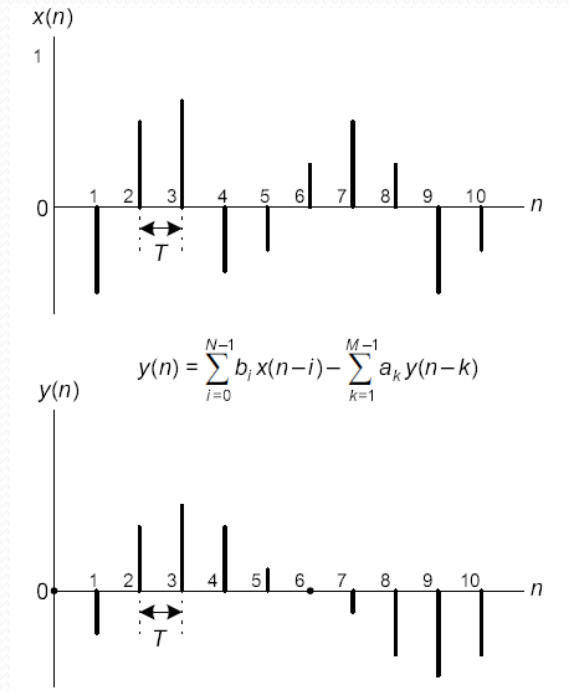
Реальное время

Пусть

T – период дискретизации,

τ_a – время выполнения алгоритма.

Определение. Говорят, что цифровая система работает в реальном времени, если время выполнения τ_a не превышает периода дискретизации.

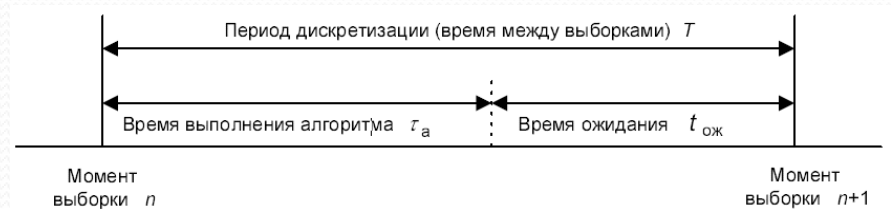


Реальное время

Это означает, что остается некоторый запас времени, обычно называемый *временем ожидания* $t_{\text{ож}}$.

Найти время выполнения алгоритма можно, если знать время выполнения элементарной (одноцикловой) команды t_k , называемое *командным циклом*, и количество командных циклов N_a , необходимое для выполнения алгоритма (это можно определить в процессе *отладки*). Тогда

$$\tau_a = t_k N_a, \quad t_{\text{ож}} = T - \tau_a$$



Тактовая частота

Тактовая частота $f_{\text{такт}}$ (тактовый период $\tau_{\text{такт}} = 1/f_{\text{такт}}$) показывает, как быстро процессор выполняет простейшую единицу работы, например пересылку в регистре из разряда в разряд. Тактовая частота должна существенно превышать частоту дискретизации.

Отношение тактовой частоты к частоте дискретизации относится к наиболее важным характеристикам, определяющим, каким образом будет реализована система. Это отношение частично определяет количество аппаратных средств, необходимое для реализации алгоритма заданной сложности в реальном времени.

Если отношение указанных частот падает, то количество и сложность аппаратных средств, требуемых для реализации алгоритма, увеличивается.

Способы реализации алгоритмов ЦОС

Возможны **три способа** реализации алгоритмов ЦОС:

- аппаратный;
- программный;
- аппаратно-программный



Аппаратная реализация

- планшет Speak&Spell
 - Texas Instruments
 - 1978



Аппаратная реализация

Аппаратная реализация подразумевает использование разнообразных **функциональных блоков**: регистров, сумматоров, шифраторов и дешифраторов, счетчиков, линий задержек, устройств памяти, умножителей, сдвигателей, логических элементов, интегральных и больших интегральных схем, программируемых логических матриц и т.п.

Совокупность функциональных блоков и связей между ними определяет реализуемый алгоритм.

Аппаратная реализация: пример

Рассмотрим пример аппаратной реализации БИХ-звена второго порядка, описываемого разностным уравнением

$$y(n) = b_0 x(n) + b_1 x(n - 1) + b_2 x(n - 2) + a_1 y(n - 1) + a_2 y(n - 2).$$

Вычисления по этому уравнению можно организовать несколькими алгоритмами, один из которых называется *прямой формой 1*.

Аппаратная реализация: пример

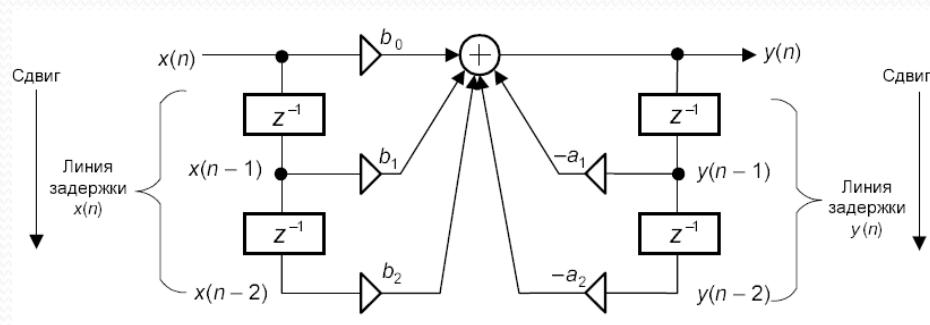
Обозначения на структурной схеме:

z^{-1} - задержка на один период дискретизации;

умножение – треугольник с соответствующим коэффициентом, а суммирование – кружок с внутренним знаком «+».

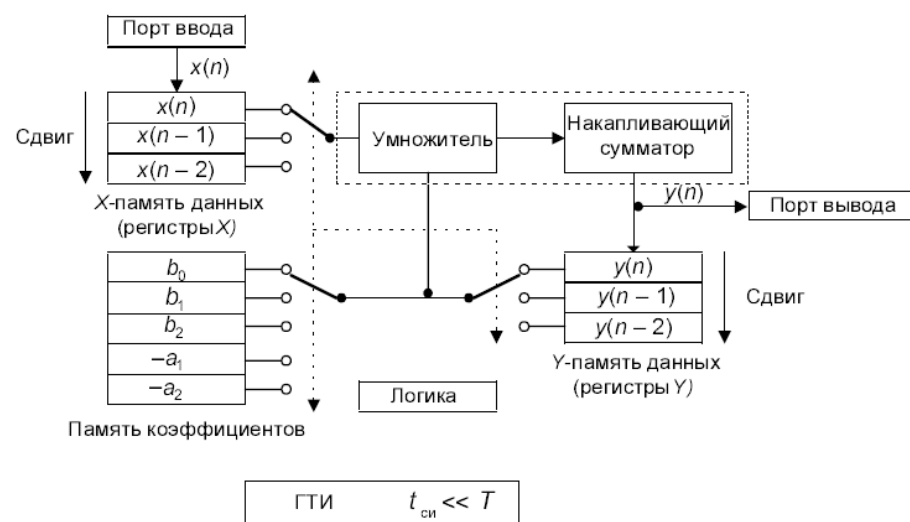
Цифровое устройство, реализующее данный алгоритм, должно иметь:

- память коэффициентов и память исходных и промежуточных данных;
- систему ввода и вывода исходных данных $x(n)$ и результата $y(n)$ соответственно;
- умножители;
- сумматор;
- генератор тактовой частоты (ГТЧ) с периодом следования импульсов $t_{\text{ти}} \ll T$



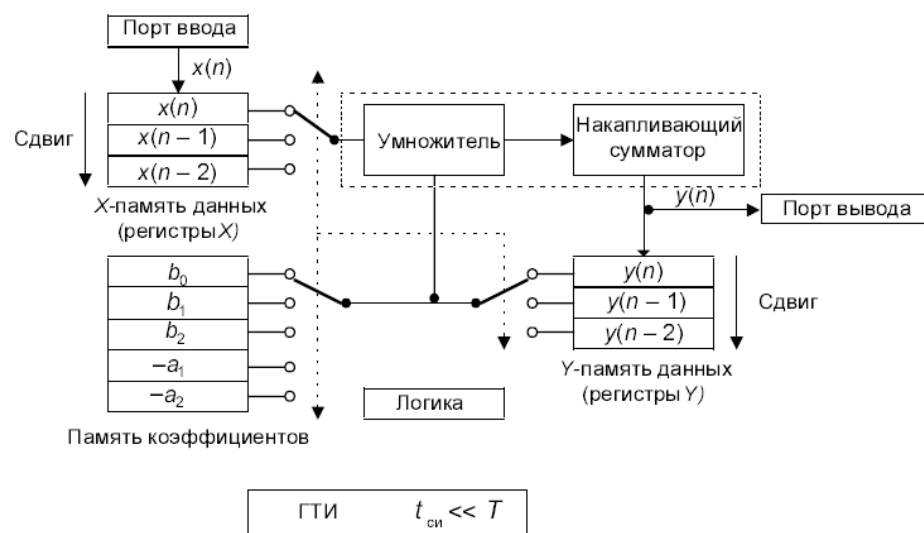
Аппаратная реализация: пример

Вместо нескольких умножителей используется один быстродействующий умножитель, на один вход которого последовательно согласно логике вычислений поступают данные $x(n)$ или $y(n)$, на другой – соответствующие коэффициенты. Получаемые произведения подаются в накапливающий сумматор, результат которого считывается один раз за период дискретизации T .



Аппаратная реализация: пример

Вместо линий задержек обычно организуют X-память и Y-память данных, а также память коэффициентов. По окончании вычислений очередного отсчета $y(n)$ осуществляется его пересылка в Y-память, откуда подается на устройство вывода. В X-памяти и Y-памяти происходит сдвиг данных, что означает готовность устройства к приему очередного отсчета $x(n)$. Ответственным за последовательность выполнения всех операций является логическое устройство. Обычно умножитель, сумматор и логическое устройство объединяются в один блок, называемый арифметико-логическим устройством (АЛУ). Связи между всеми блоками осуществляются с помощью шин, разрядность которых соответствует разрядности X-памяти и Y-памяти.



Аппаратная реализация

Достоинство аппаратной реализации – очень высокое быстродействие (за счет применения функциональных блоков на базе ТТ-логики, распараллеливаемых операций и узкой направленности создаваемых устройств).

Недостатки аппаратной реализации:

- 1) аппаратная реализация, ориентированная на решение узкоспециализированных задач, подразумевает создание систем с жесткой логикой, когда любое изменение алгоритма требует изменения структуры устройства;
- 2) аппаратная реализация также приводит к большому потреблению энергии и к необходимости организовывать теплоотвод;
- 3) высокая стоимость аппаратной реализации; проектирование, изготовление и отладка оказываются весьма трудоемкими при больших временных затратах.

Программная реализация

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 блоку.

Для персонального компьютера это будет любой из языков высокого уровня (C++, Java и др.), для микропроцессорного комплекта или цифрового процессора – соответствующий язык ассемблера.

Программная реализация

Достоинства программной реализации:

- 1) неизменная структура системы при различных алгоритмах и областях применения;
- 2) хорошая гибкость, позволяющая достаточно легко изменять алгоритмы работы системы за счет коррекции или изменения программы;
- 3) существенное ускорение, облегчение и удешевление проектирования, изготовления и отладки системы.

Недостатком программной реализации является относительно низкое быстродействие по причине последовательного выполнения операций программы в одном процессоре.

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.

Аппаратно-программная реализация



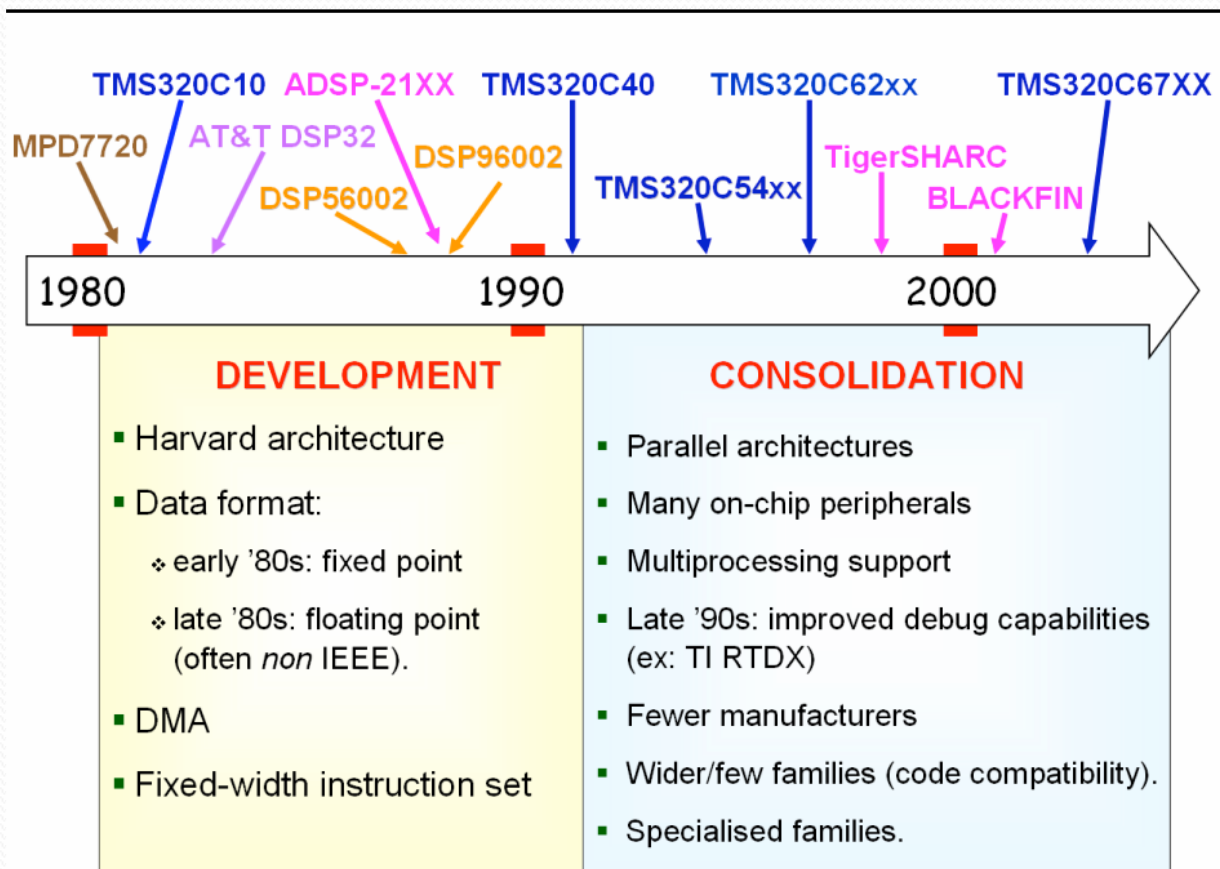
DSP vs Общего плана CPU



Skylake Xeon Platinum 8180M и TMS320C6713BZDP300

	CPU (Intel)	DSP (TI)
Частота	2.5 ГГц	500 МГц
Число ядер	28	1
Пиковая производительность	560 GIPS	1.8 GIPS
Энергопотребление	205 Вт	1 Вт
Out-of-order	Да	Нет
Цена	\$13K (+ система охлаждения)	\$35 (оптовая цена)
Target applications	Любые	<ul style="list-style-type: none">- Большое число циклов- Высокий параллелизм по данным- Регулярный паттерн доступа в память
Производительность / Вт / ядро	0.097 GIPS/Вт/ядро	1.7 GIPS/Вт/ядро (в 17 раз лучше Intel)
Производительность / Вт / ядро / \$	0.0075 MIPS/Вт/ядро/\$	0.051 GIPS/Вт/ядро/\$ (в 7000 раз лучше Intel)

Эволюция DSP



Эволюция DSP

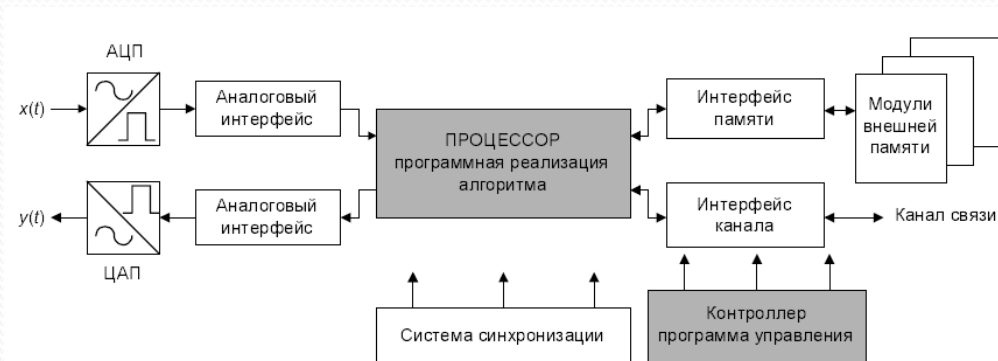
Table 2: Overview of DSP device characteristics as a function of time. The last column refers to expected values.

Characteristic \ Year		Year			
		1980	1990	2000	> 2010
Wafer size	[inches]	3	6	12	18
Die size	[mm]	50	50	50	5
Feature	[μm]	3	0.8	0.1	0.02
RAM	[Bytes]	256	2000	32000	1 million
Clock frequency	[MHz]	20	80	1000	10000
Power	[mW/MIPS]	250	12.5	0.1	0.001
Price	[USD]	150	15	5	0.15

Пример аппаратно-программной реализации системы ЦОС

К процессору, работающему по заданной программе подключены:

- АЦП и ЦАП;
- модули внешней памяти, хранящие программы, разнообразные константы и таблицы функций, что позволяет заменять длительное их вычисление быстрым обращением к памяти;
- интерфейс – специальные вспомогательные схемы, обеспечивающие согласование сигналов на стыках с модулями внешней памяти, АЦП, ЦАП, каналом связи;
- система синхронизации, обеспечивающая временное согласование всех элементов системы.



Аппаратно-программная реализация системы ЦОС

Аппаратно-программная реализация сочетает положительные свойства аппаратной и программной реализации. Разумное сочетание аппаратных и программных средств позволяют снизить требования к вычислительным возможностям элементной базы и упростить реализацию систем ЦОС.

Особенности ЦОС, влияющие на элементную базу

1. Высокая скорость поступления данных

Пример: отсчеты аудиосигнала поступают в устройство обработки со скоростью от 8000 до 20000 отсчетов в секунду.

Скорость обработки данных определяется производительностью процессора, которая выражается количеством миллионов одноцикловых команд, выполняемых в секунду: в MIPS (Million Instructions Per Second) для процессоров с ФТ и в MFLOPS (Million Float Operations Per Second) для процессоров с ПТ. Производительность, выражаемая в MIPS (FLOPS), является пиковой, т.е. предельно возможной для данного процессора.

Реальная производительность может быть значительно меньшей и потому ее оценивают временем выполнения стандартных алгоритмов; в частности, временем выполнения 1024-точечного БПФ.

Другой способ определения реальной производительности - BDTImark (см. <http://www.bdti.com>) состоит в тестировании ЦПОС на группе специальных задач. Результат тестирования выражается в относительных условных единицах: чем выше производительность, тем большим количеством единиц оценивается процессор.

Особенности ЦОС, влияющие на элементную базу

- нет пропорциональной зависимости реальной производительности от пиковой;
- процессоры с одинаковой пиковой производительностью не обязательно имеют одинаковую реальную производительность.

Процессор	Производительность	
	Пиковая (MIPS)	Реальная в единицах BDTmark
LUSENT DSP161210	100	36
Motorola DSP56303	100	25
TMS320VC549	100	25
ADSP-2189M	75	19
TMS320C6201	1000	600

Особенности ЦОС, влияющие на элементную базу

2. Широкий диапазон изменения входных/выходных данных.

Обычно диапазон данных составляет 40-80 дБ, а в радиоприемных устройствах может доходить до 100 дБ. Следовательно, в ряде случаев необходимо иметь такую элементную базу, которая обеспечивала бы организацию обработки данных большой разрядности. Если учесть, что один бит соответствует ≈ 6 дБ, то разрядность регистров сомножителей при различных диапазонах обязаны быть следующей

Динамический диапазон (дБ)	Разрядность регистров сомножителей	Разрядность регистра произведения
40	7	14
50	9	18
60	10	20
70	12	24
80	14	28
100	17	34

Особенности ЦОС, влияющие на элементную базу

3. Большое количество операций сложения, умножения и логических операций

Эти операции требуются для вычисления одного выходного отсчета. Все виды сложной обработки могут быть представлены композицией операторов: свертки, рекурсии, ДПФ, нелинейных и логических преобразований. Элементная база должна быть ориентирована на быстрое выполнение таких операторов. Должно быть организовано аппаратное умножение с накоплением и создана большая память данных и память программ с удобным и быстрым доступом к ним.

Особенности ЦОС, влияющие на элементную базу

4. **Необходимость обеспечения гибкости и перестройки** цифровых систем обработки сигналов, что связано с изменением разнообразных параметров, коэффициентов и данных в регулируемых и адаптивных системах.
5. **Параллелизм алгоритмов**, проявляющийся в том, что для каждого набора входных данных выполняются такие действия, которые могут совмещаться по времени.
6. **Регулярность алгоритмов**, т.е. повторяемость отдельных операций (например, «бабочка» в БПФ).

Основные свойства ЦПОС

Основные свойства ЦПОС, обеспечивающие эффективную реализацию алгоритмов ЦОС:

- быстрое выполнение типовых операций ЦОС;
- аппаратная реализация комплексной операции умножения с накоплением (суммирование локальных произведений);
- применение арифметики с ФТ и ПТ с разнообразной разрядностью;
- параллельное выполнение отдельных частей алгоритма, которое достигается аппаратной реализацией типовых алгоритмов;
- большая внутрикристалльная память данных и память программ;
- разнообразие режимов адресации применительно к различным задачам: организация буферов, поддержка бит-реверсной адресации в БПФ и др.
- обработка в реальном времени данных, поступающих с высокой скоростью;
- наличие внутрикристалльной периферии (последовательных и параллельных интерфейсов, портов ввода/вывода, таймеров);
- малое время обращения к элементам внешней периферии.

Основные свойства ЦПОС

Свойства	Применение
Быстрое умножение с накоплением	Большинство алгоритмов ЦОС (фильтрация, преобразования, спектральный анализ, нелинейная обработка и т. д.) насыщены операциями сложения и умножения
Архитектура с параллельным доступом к памяти	Увеличение производительности, поскольку многие операции ЦОС, работающие с большими объемами данных, требуют чтения команд программы и многократного обращения к данным во время каждого командного цикла
Режимы специальной адресации	Эффективная поддержка массивов данных и буферов типа FIFO ("первым вошел — первым вышел")
Управление специальными программами	Эффективное управление циклами в многоитеративных алгоритмах ЦОС; быстрое прерывание, поддерживающее часто повторяемые команды типа ввода/вывода
Внутрикристальная периферия и интерфейсы ввода/вывода	Внутрикристальная периферия, включающая в себя разнообразные устройства (компарандеры, кодеки, таймеры, интерфейсы ввода/вывода, приспособленные к внешней периферии общего назначения и др.), позволяет разрабатывать компактные системы малой стоимости

Цифровые сигнальные процессоры

Компьютеры, предназначенные для бизнеса или других применений общего назначения, не могут быть оптимальными для реализации таких алгоритмов, как цифровая фильтрация и анализ Фурье.

Цифровые сигнальные процессоры представляют собой класс микропроцессоров, специально разработанный для решения задач цифровой обработки сигналов.

Digital Signal Processor – цифровой сигнальный процессор, ЦСП (среди инженеров)

Digital Signal Processing – цифровая обработка сигналов, ЦОС (среди ученых)

ЦСП разрабатываются специально для выполнения математических вычислений, необходимых в цифровой обработке сигналов.

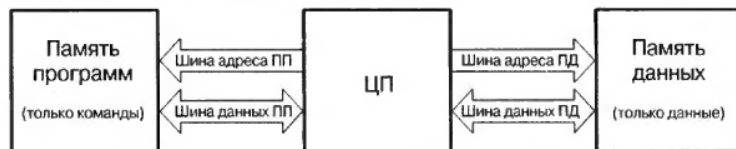
	Манипуляция данными	Математические вычисления
Типичные приложения	Обработка текста, управление базой данных, большие таблицы, операционные системы и т. п.	Цифровая обработка сигналов, управление движением, научное и техническое моделирование и т. п.
Основные операции	Перемещение данных ($A \rightarrow B$) Проверка значений (Если $A = B$, то ...)	Сложение ($A + B = C$) Умножение ($A \times B = C$)

Архитектура цифрового сигнального процессора

а) Архитектура фон Неймана (один модуль памяти)



б) Гарвардская архитектура (два модуля памяти)



в) Супергарвардская архитектура (два блока памяти, кэш-память команд, контроллер ввода/вывода)



Архитектура микропроцессоров:

а) Фон-неймановская архитектура использует одну память и для данных, и для команд;

б) Гарвардская архитектура использует отдельную память для команд и для данных, обеспечивая более высокую скорость обмена;

в) Супер Гарвардская архитектура улучшает гарвардскую конструкцию добавлением кэш-памяти команд и специального контроллера ввода/вывода.

Архитектура цифрового сигнального процессора

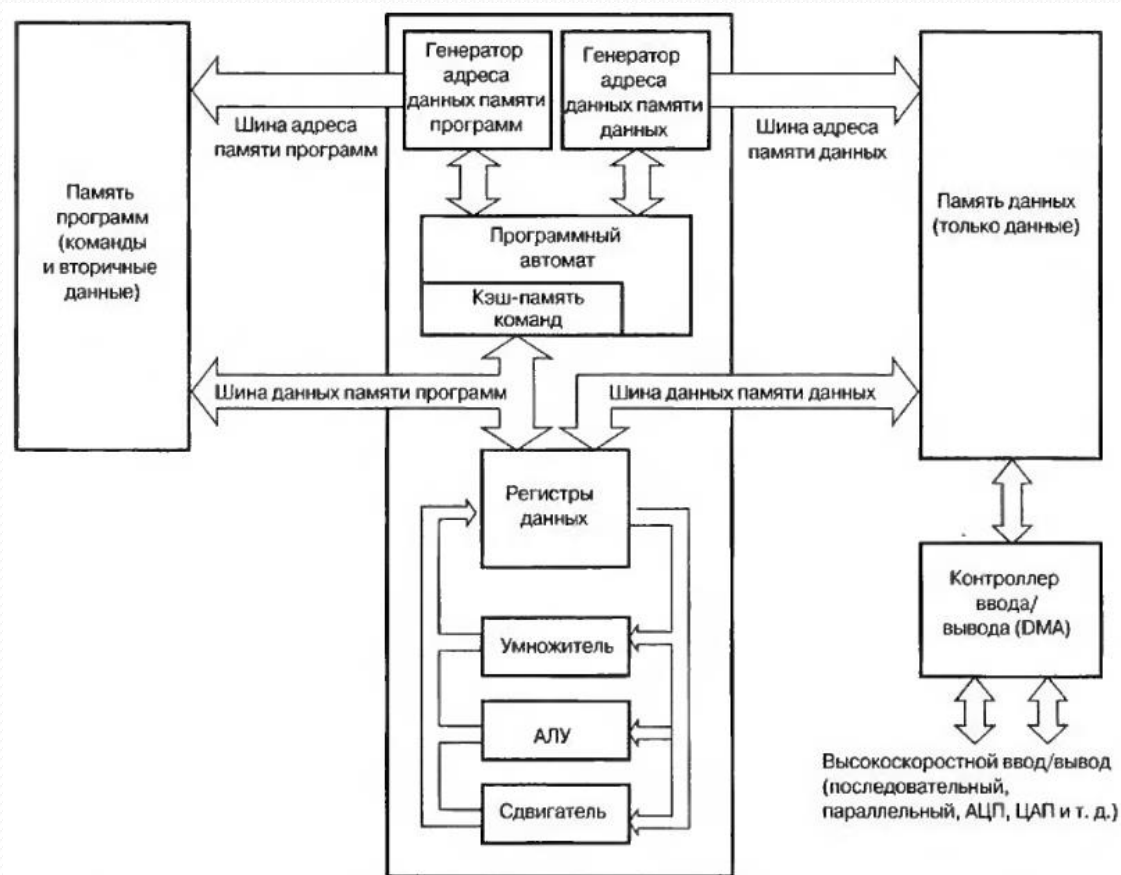
Большинство современных ЦСП используют двухшинную гарвардскую архитектуру.

Термин «супергарвардская архитектура» был введен фирмой Analog Devices для описания работы ЦСП, семейств ADSP-2106x и ADSP-21xx.

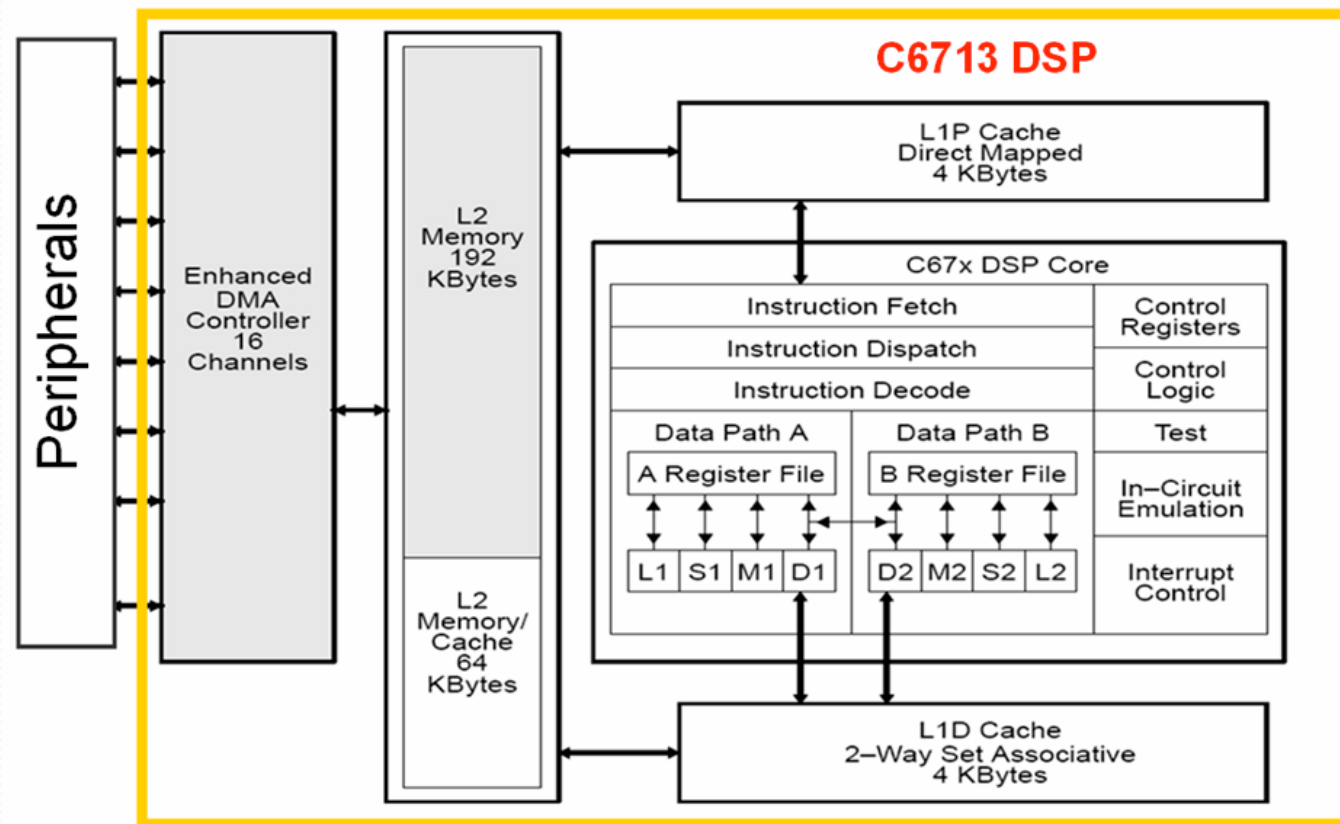
ЦСП, построенных по такой архитектуре, называли **SHARC**, что является сокращением от слов Super Harvard ARChiterture.

Архитектура ЦСП семейства SHARC оптимизирована по десяткам направлений, однако два из них играют особую роль – наличие кэш-памяти команд и контроллера ввода/вывода.

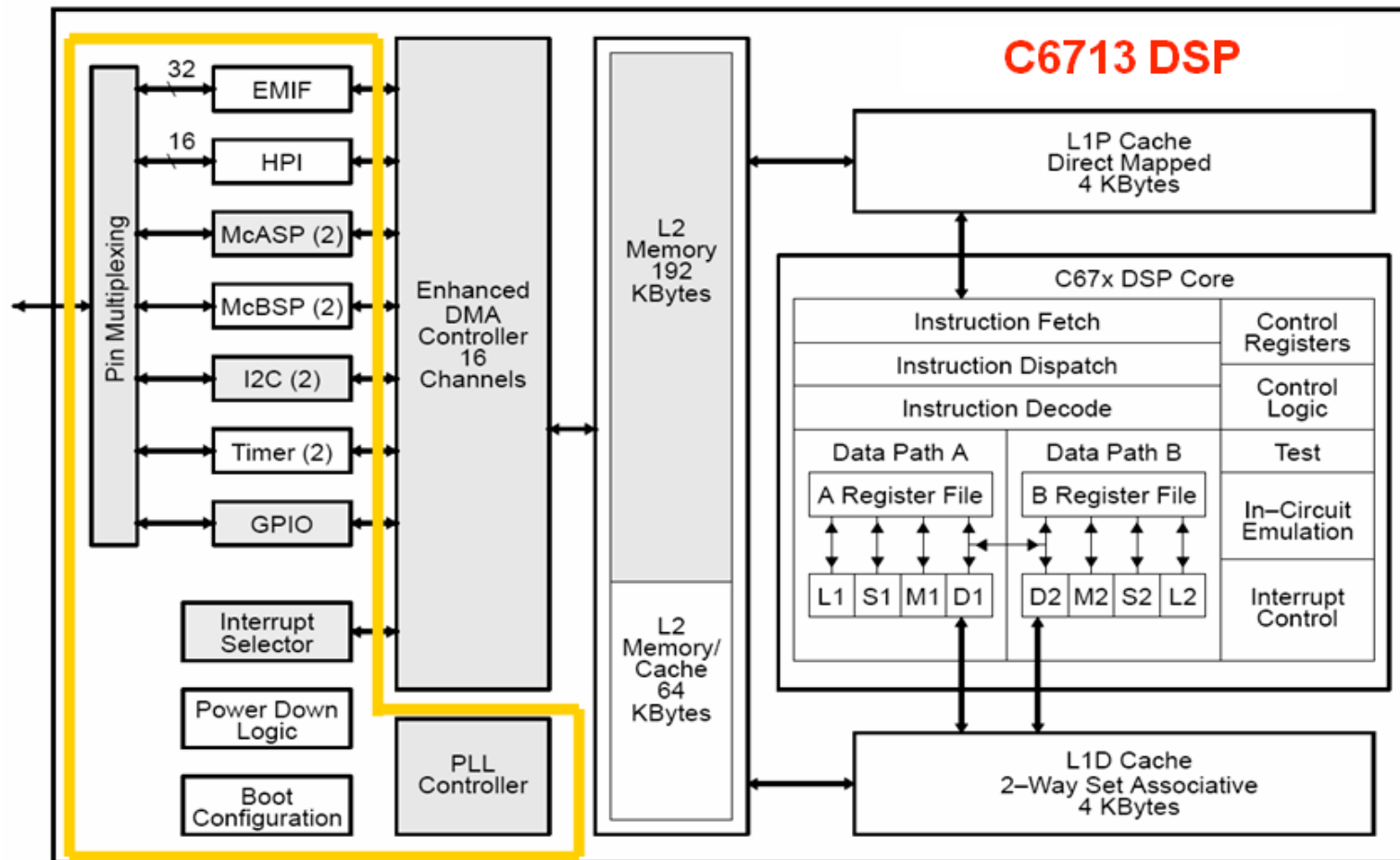
Упрощенная схема одного из процессоров семейства SHARC фирмы Analog Devices



TMS320C6713



TMS320C6713



Рынок цифровых сигнальных процессоров

Manufacturer	Family	Typical use and performance
TI	TMS320C2x	Digital signal controllers
	TMS320C5x	Power efficient
	TMS320C6x	High performance
ADI	SHARC	Medium performance. First ADI family (now three generations)
	TigerSHARC	High performance for multi-processor systems
	Blackfin	High performance and low power

Профессиональные журнал, посвященные рынку ЦСП:

- EDN (Electronic Design News, www.ednmag.com)
- ECN (Electronic Component News, www.ecnmag.com)