

Архитектура и программирование DSP-процессоров

- Примеры приложений и базовые алгоритмы ЦОС.
Форматы данных
- Обобщенная архитектура DSP-процессоров. Основные элементы архитектуры процессоров ADSP-21х60
- Обзор аппаратных средств ЦОС. Альтернативные средства ЦОС. Тенденции рынка DSP-процессоров
- Этапы и средства разработки ПО для DSP-процессоров
- Базовые подходы к программной реализации алгоритмов ЦОС на DSP

Хусаинов Наиль Шавкятович
к.т.н., доцент кафедры МОП ЭВМ ТРТУ
KhussainovNSh@mopevm.tsure.ru

1. Примеры приложений и базовые алгоритмы ЦОС. Форматы данных

1.1. Принципы ЦОС

1.2. Сравнение цифровых и аналоговых методов обработки сигналов

1.3. Задачи ЦОС

1.4. Функциональная схема системы ЦОС

1.5. Основные области применения ЦОС. Пример применения алгоритмов ЦОС в сотовой связи

1.6. Базовые алгоритмы ЦОС

1.7. Форматы данных

1.1. Принципы цифровой обработки сигналов

- цифровое представление сигналов с необходимой точностью
- выполнение в реальном или близком к реальному масштабе времени большого количества вычислительных операций над потоками данных (массивами), в том числе бесконечными во времени

1.2. Аналоговая и цифровая обработка

Недостатки цифровой обработки сигналов

- расширение полосы частот
- необходимость АЦП
- необходимость временной синхронизации
- несовместимость с существующими аналоговыми устройствами)

1.2. Аналоговая и цифровая обработка

Преимущества цифровой обработки сигналов-1

- возможность регенерации сигнала
- возможность работы при малых значениях отношения сигнал/шум
- простота передачи управляющей информации
- индифферентность к характеру нагрузки
- простота группообразования
- простота засекречивания
- высокая надежность и степень интеграции с другими устройствами (ЭВМ)

1.2. Аналоговая и цифровая обработка

Преимущества цифровой обработки сигналов-2

- эффективная обработка данных
 - а) программируемость
 - б) совместное использование
 - в) автоматический контроль
 - г) универсальность

1.3. Задачи, решаемые ЦОС

Основная задача: обработка сигнала с целью выделения из него нужной информации для ее последующего хранения, преобразования или передачи

- анализ сигналов
- синтез сигналов

1.3. Задачи, решаемые в ЦОС

Примеры приложений с анализом сигналов

- определение (распознавание) типа объекта по некоторой совокупности сигналов
- анализ речевого сигнала для идентификации говорящего
- выделение речевого сигнала в условиях помех
- анализ сейсмических сигналов
- анализ электрокардиограмм и т.п.
- анализ свойств материалов (рентгенография, спектроскопия)
- анализ трафика в сетях передачи данных
- анализ временных рядов (курсы акции и т.п.)

Общие требования:

- автоматический (автоматизированный) режим работы
- работа в реальном времени
- значимость извлекаемых параметров

1.3. Задачи, решаемые в ЦОС

Примеры приложений с синтезом сигналов

- компрессия сигналов
- удаление из сигнала нежелательных шумов, улучшение качества звучания звука
- изменение свойств изображения (контраста, цветовой насыщенности, резкости)

Варианты реализации:

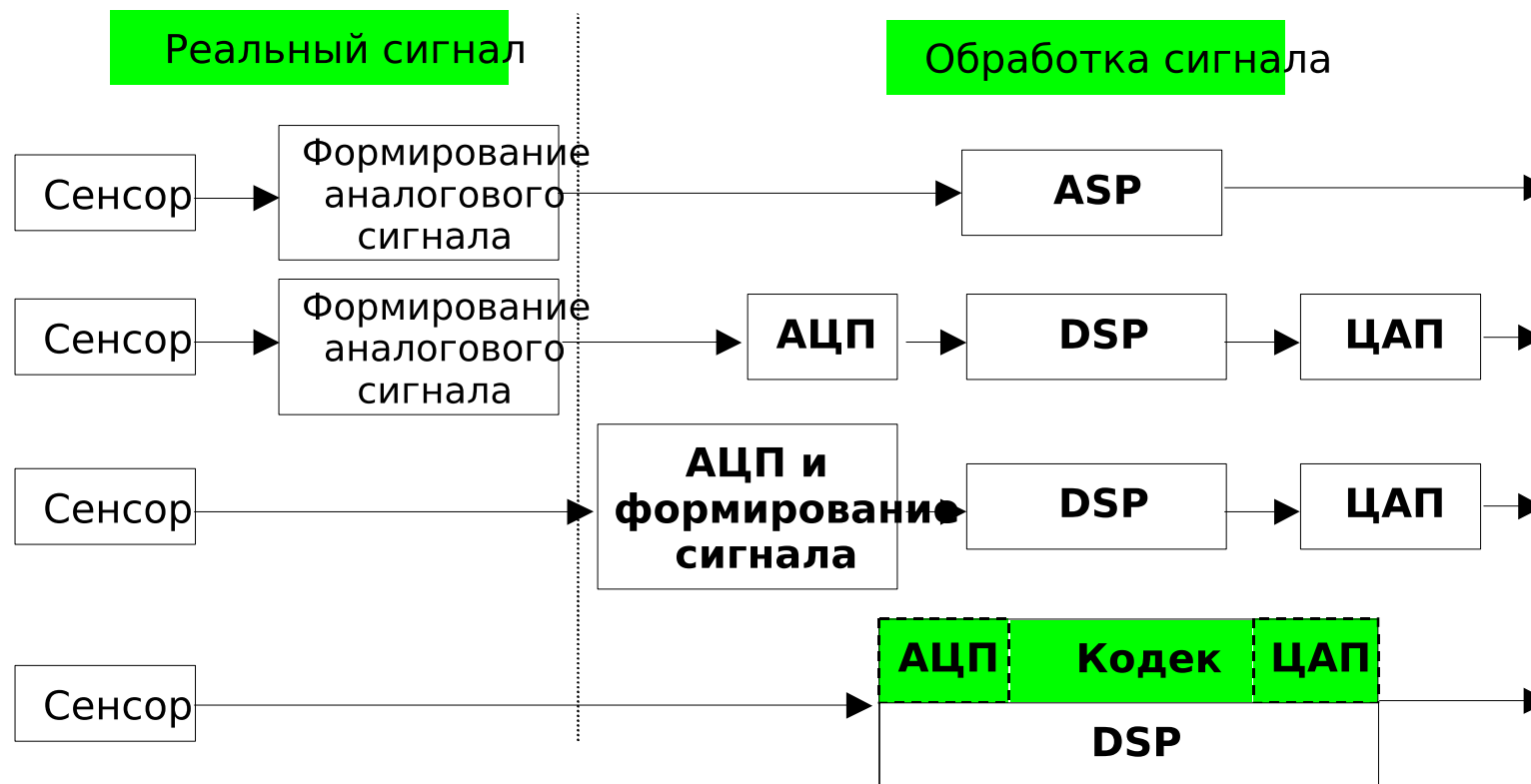
- полное / неполное восстановление
- симметричное / несимметричное преобразование

1.3. Связь ЦОС с другими областями науки

Цифровая обработка сигналов	Теория связи и телекоммуникаций
	Методы численного анализа
	Теория вероятностей и математическая статистика
	Аналоговая обработка сигналов
	Теория принятия решений
	Цифровая электроника
	Аналоговая электроника

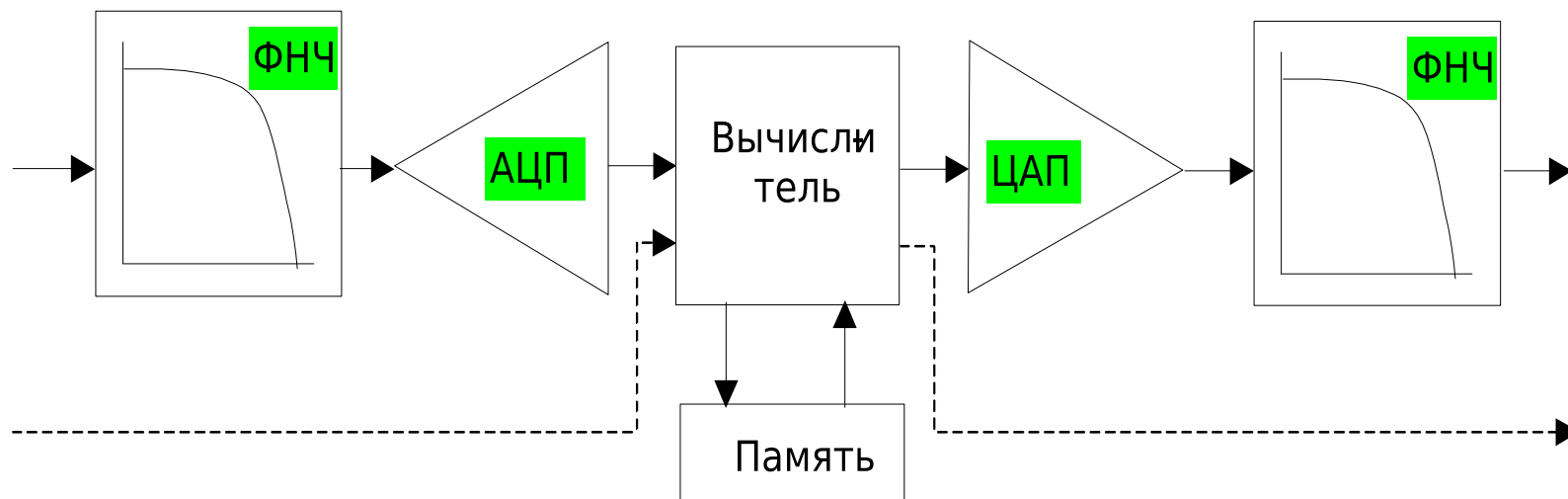
1.4. Функциональные схемы системы ЦОС

Реализация этапов аналоговой и цифровой обработки сигналов

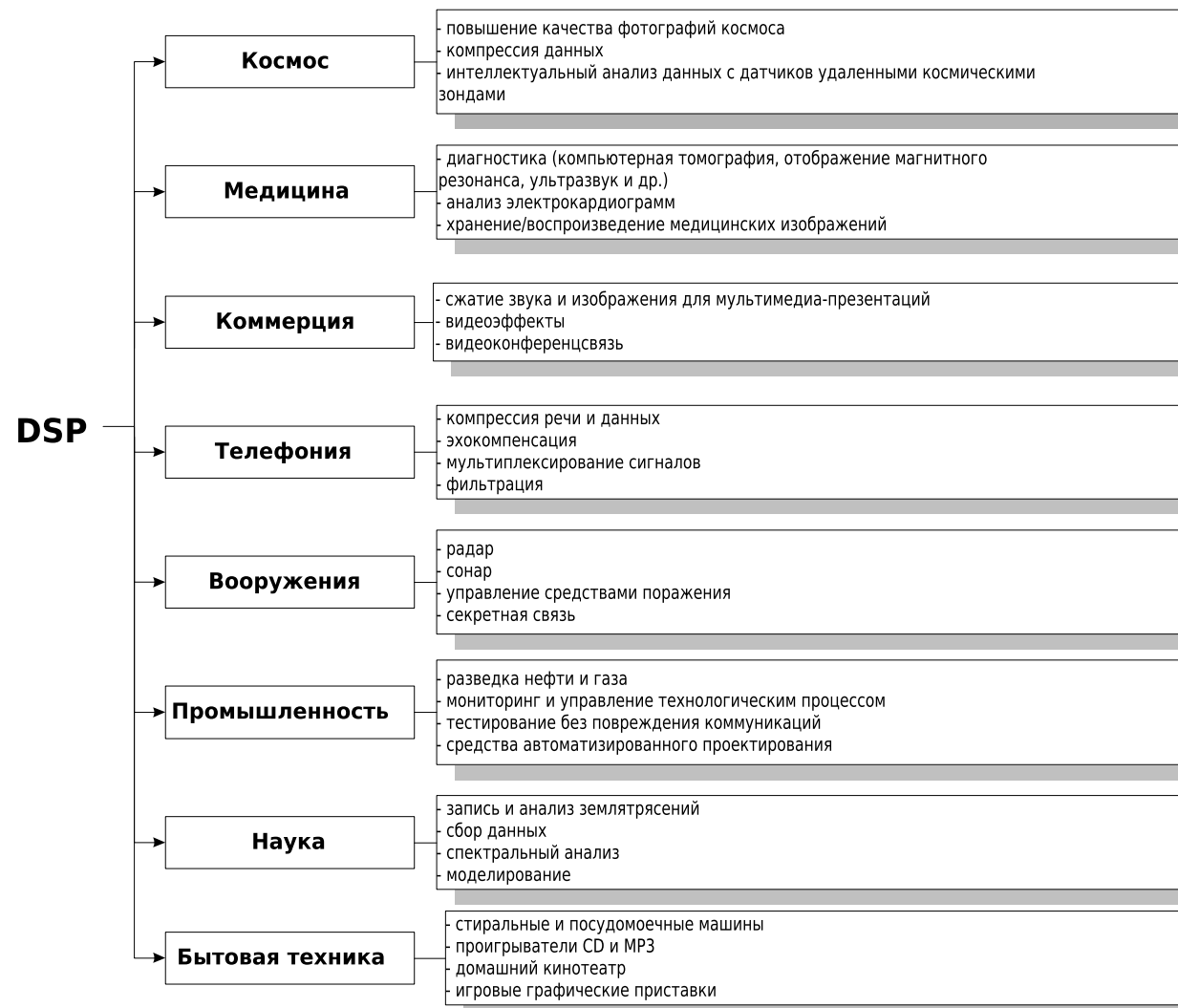


1.4. Функциональная схемы системы ЦОС

Состав системы ЦОС



1.5. Основные приложения ЦОС



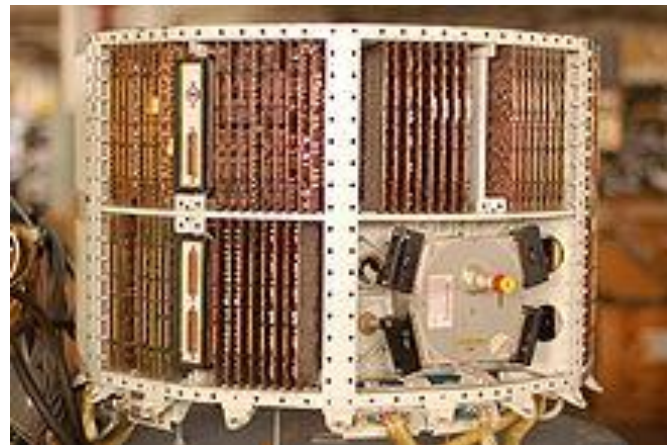
ПЕРВЫЕ ВСТРАИВАЕМЫЕ СИСТЕМЫ И ПРОЦЕССОРЫ

•Apollo Guidance Computer (1965г., 16-битный, 2 Кб ОЗУ, 36 Кб ПЗУ - навигация, наведение и управление космическим кораблем, программа Apollo)



ПЕРВЫЕ ВСТРАИВАЕМЫЕ СИСТЕМЫ И ПРОЦЕССОРЫ

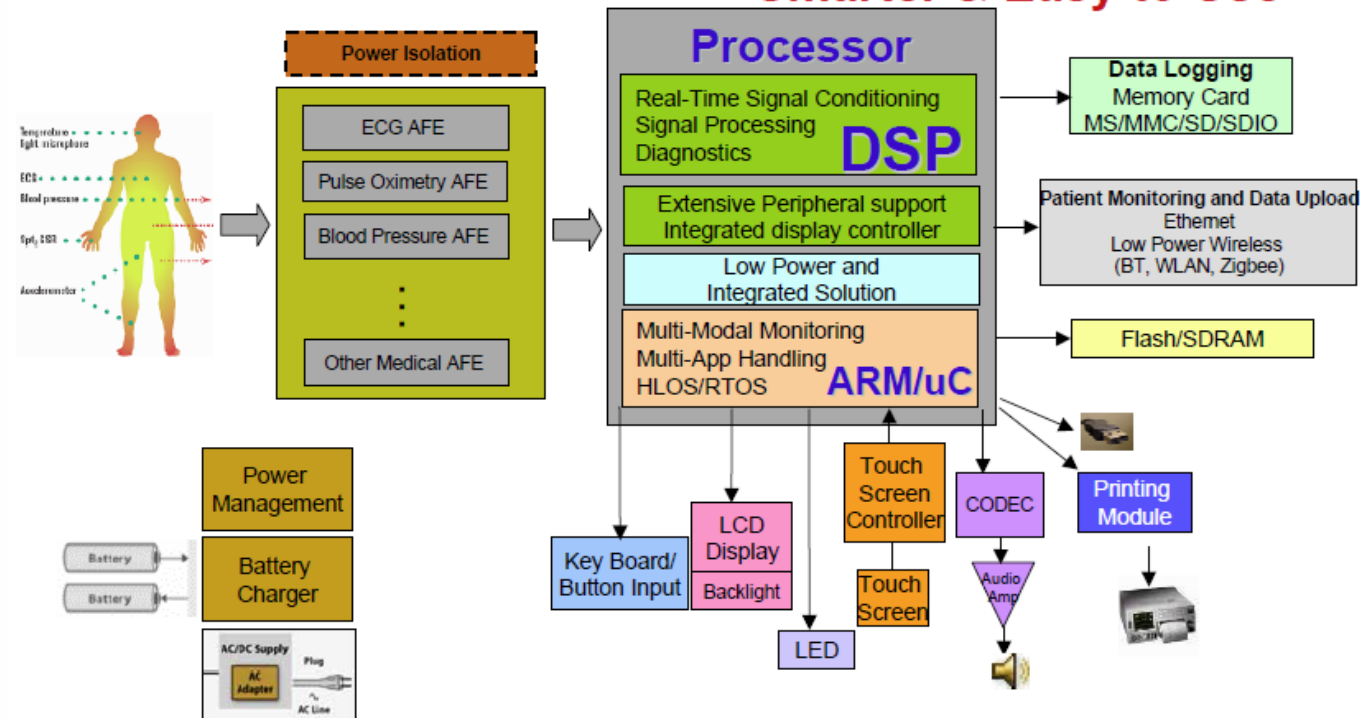
•Autonetics D-17 guidance computer (1962г., 24-битный, 345 Гц, 28 кг (с платформой стабилизации и источником питания) – решение уравнений навигации и наведения, управляемая межконтинентальная баллистическая ракета Minuteman I)



DSP-ПРОЦЕССОРЫ В МЕДИЦИНЕ

Typical Medical Device Block Diagram

Smarter & Easy to Use



medicalsolutions@arrow.com



DSP-ПРОЦЕССОРЫ В МЕДИЦИНЕ

Medical Devices Landscape



medicalsolutions@arrow.com

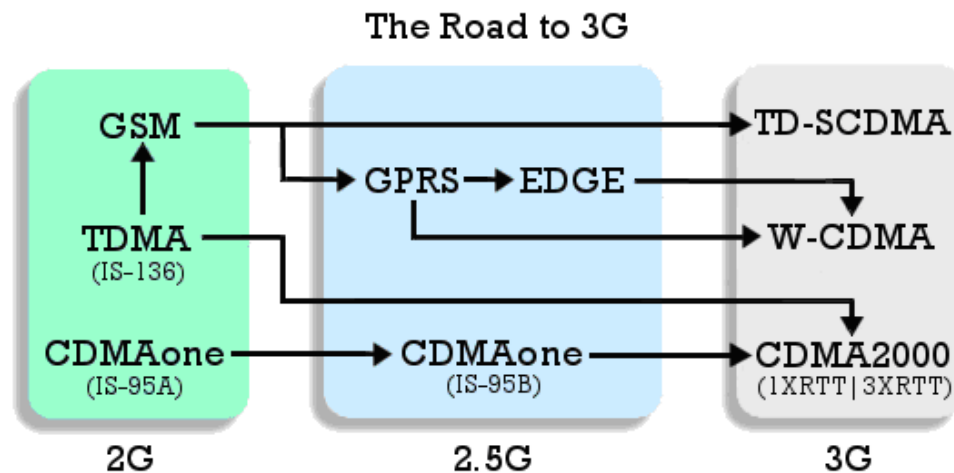


ОБЛАСТИ ПРИМЕНЕНИЯ И ТРЕБОВАНИЯ К ПРОЦЕССОРАМ ДЛЯ ВСТРАИВАЕМЫХ СИСТЕМ И СИСТЕМ ЦОС

Область применения	Требования к DSP/CPU/МС
Игровые консоли	Высокая производительность графической подсистемы (3D-графика, ПЗ-арифметика, быстродействующая память)
КПК, Handheld PC, автомобили, сетевое оборудование	Механизмы управления виртуальной памятью, стандартный набор периферии (последовательные порты, порты ввода/вывода общего назначения, контроллер дисплея)
Мобильные телефоны и коммуникаторы	Сверхнизкое энергопотребление, высокая производительность, поддержка алгоритмов ЦОС
Модемы, факсы, принтеры	Низкая стоимость
Ресиверы цифрового телевидения (set-top box), DVD	Высокая степень интеграции подсистем (блоков) процессора
Цифровые видеокамеры	Поддержка алгоритмов обработки растровой графики

1.5. Пример применения ЦОС в СОТОВОЙ СВЯЗИ

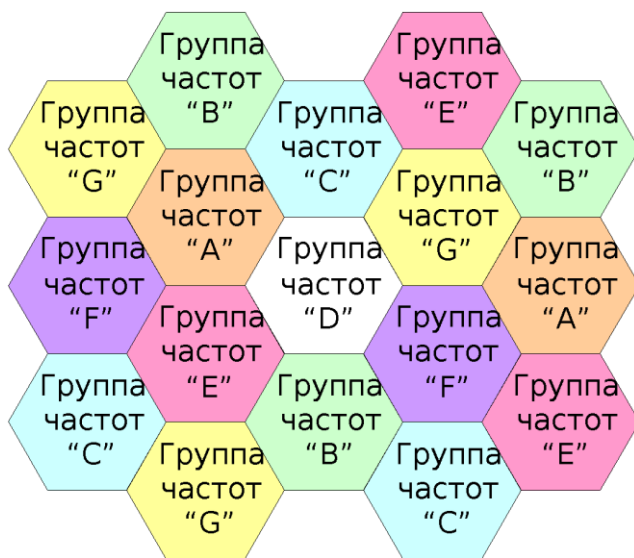
Эволюция и основные стандарты сетей СОТОВОЙ СВЯЗИ



- Ø FDMA - выделение каналов на основе частотных слотов (пример - система AMPS)
- Ø TDMA - выделение каналов на основе временных слотов (пример - суммарная пропускная способность примерно в 3 раза выше чем в FDMA) (пример - стандарт GSM)
- Ø CDMA - выделение и переключение каналов на основе технологии расширенного спектра (пример - плавное снижение качества при увеличении числа задействованных каналов)
- Ø В TDMA и CDMA интенсивно используются DSP-процессоры для кодирования речи и канального кодирования

1.5. Применение ЦОС в СОТОВОЙ СВЯЗИ

Структура и базовые принципы

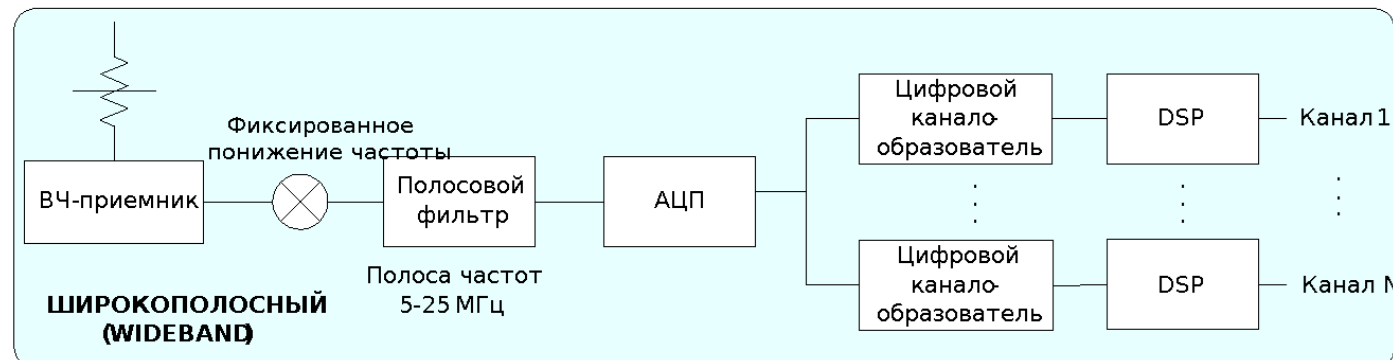
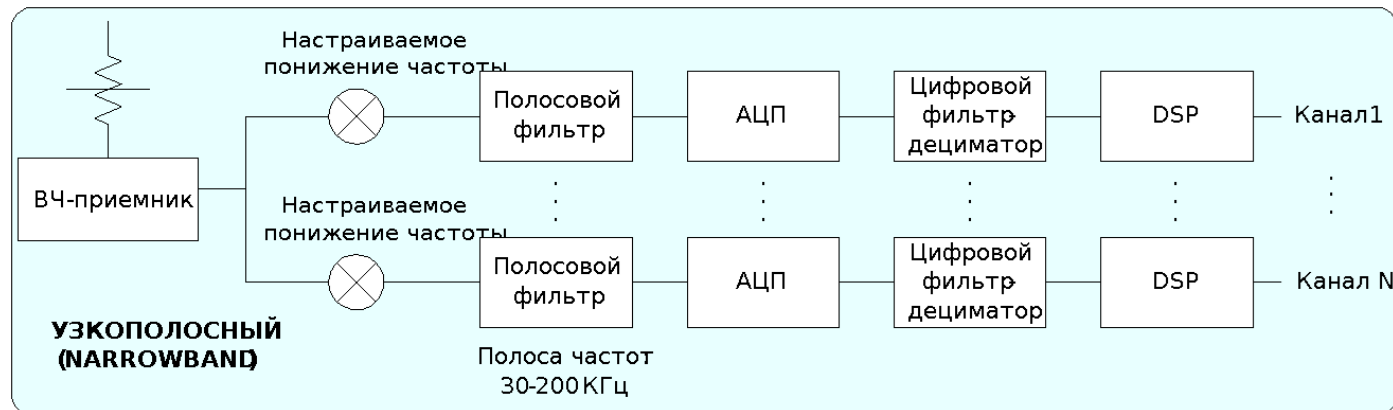


Многократное использование частот в сотовой связи

- Ø Радиус каждой соты порядка 1-15 км
- Ø В каждой соте есть собственная базовая станция (Basestation) для приема и ретрансляции сигнала
- Ø Каждая сота должна обрабатывать несколько вызовов одновременно
- Ø Взаимодействие с абонентом может переходить от одной соты к другой

1.5. Применение ЦОС в СОТОВОЙ СВЯЗИ

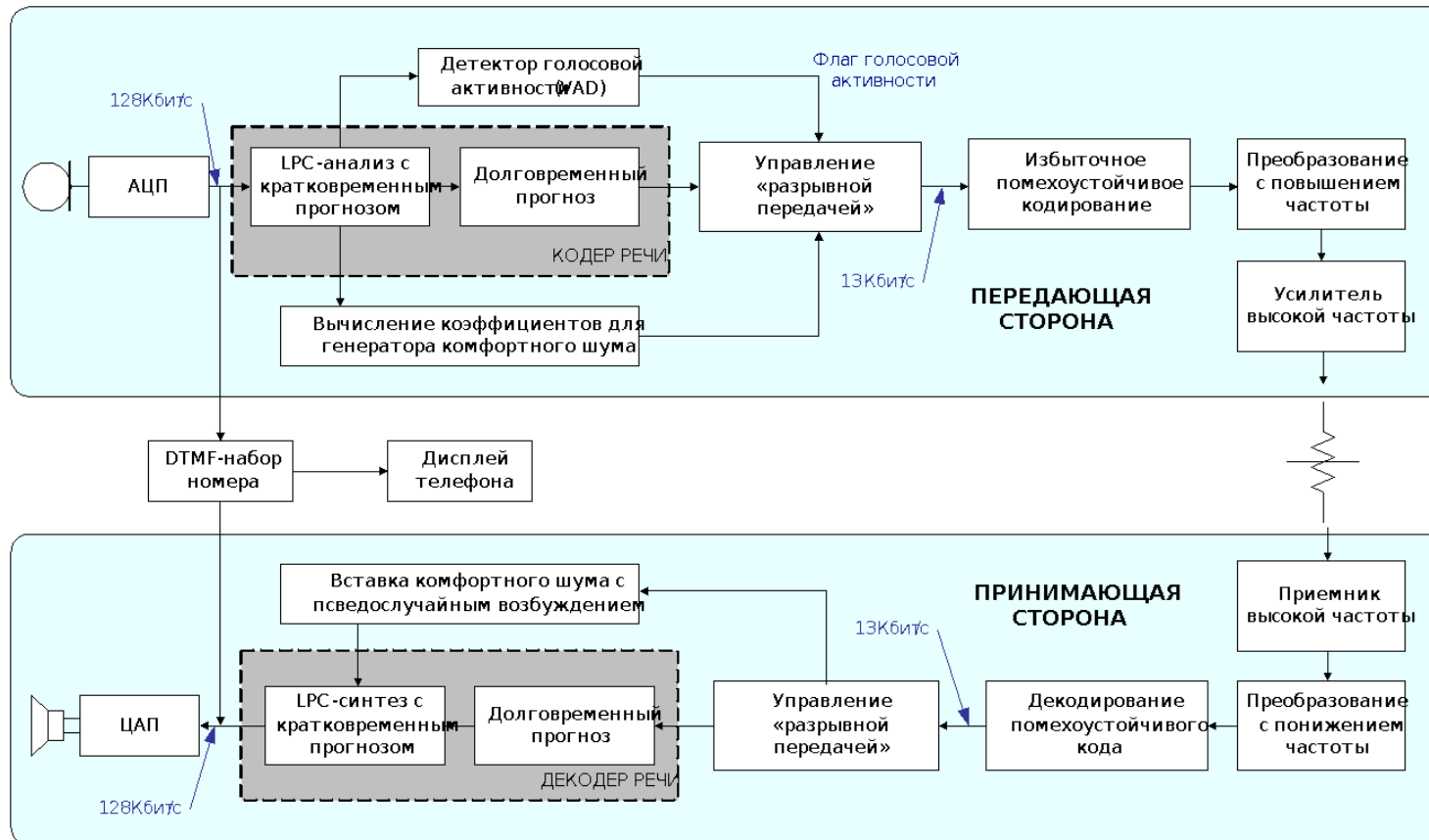
Базовая станция (Basestation)



Узкополосный и широкополосный цифровой приемник для базовой станции сотовой связи

1.5. Применение ЦОС в СОТОВОЙ СВЯЗИ

Телефон стандарта GSM (функциональные блоки)



ПОЛЕЗНЫЕ ССЫЛКИ ПО DSP И ВСТРАИВАЕМЫМ СИСТЕМАМ

- .EE Times Embedded.com <http://www.eetimes.com/design/embedded>
(сообщество разработчиков встраиваемых систем, статьи по HW и SW для embedded, вебинары)
- .Embedders.org (блоги разработчиков встраиваемых систем, рус.)
<http://embedders.org/>
- .Обзор книг (англ.) по встраиваемым системам
<http://www.barrgroup.com/embedded-systems/books>
- .DSPRelated.com (англ.) – сайт для инженеров и разработчиков в области ЦОС и DSP www.dsprelated.com
- .Steven W. Smith The Scientist and Engineer's Guide to Digital Signal Processing (англ., online) <http://www.dspguide.com>
- .Computers in Spaceflight: The NASA Experience (англ.)
<http://www.hq.nasa.gov/pao/History/computers/CompSPACE.html>
- .Журнал «Компоненты и технологии» <http://www.kit-e.ru/>
- .Журнал «Современные технологии автоматизации» www.cta.ru
- .Время электроники www.russianelectronics.ru
- .EmbeddedGurus. Experts on Embedded Software (англ.)
<http://embeddedgurus.com/>

1.6. Базовые алгоритмы ЦОС

Перечень основных алгоритмов ЦОС

- БИХ- и КИХ-фильтрация сигнала
- корреляционная функция двух сигналов или автокорреляционная функция сигнала
- прямое/обратное преобразование Фурье
- помехоустойчивое (канальное) кодирование
- формирование пакетов для передачи по каналам связи

1.6. Базовые алгоритмы ЦОС

Области применения вычислителей

Работа с данными	Математические вычисления
<ul style="list-style-type: none">- офисные приложения- СУБД	<ul style="list-style-type: none">- цифровая обработка сигналов- управление двигателями- техническое моделирование- обработка данных в реальном времени
<ul style="list-style-type: none">- пересылка данных ($A \rightarrow B$)- сравнение данных (ЕСЛИ $A=B$ ТО ... ИНАЧЕ ...)	<ul style="list-style-type: none">- сложение ($C = A + B$)- умножение ($C = A \times B$)
<ul style="list-style-type: none">- время исполнения не критично, заранее не нормируется	<ul style="list-style-type: none">- время исполнения критично, заранее нормируется

1.6. Базовые алгоритмы ЦОС

Фильтрация сигнала

Фильтр с конечной импульсной характеристикой:

$$y_n = \sum_{k=0}^{N-1} b_k x_{n-k} = b_0 x_n + b_1 x_{n-1} + b_2 x_{n-2} + \dots + b_{N-1} x_{n-N+1}$$

Фильтр с бесконечной импульсной характеристикой:

$$y_n = \sum_{k=0}^{\infty} b_k x_{n-k} + \sum_{k=1}^{\infty} a_k y_{n-k}$$

Для эффективной реализации необходимы:

- выполнение операций «умножения с накоплением»;
- одновременная выборка из памяти двух операндов;
- доступ к памяти с автоматической модификацией указателя;
- циклическая организация буфера (кольцевые буферы);
- аппаратная поддержка циклов

1.6. Базовые алгоритмы ЦОС

Фурье-преобразование-1

$$X(n) = \sum_{l=0}^{N-1} x(l)W_N^{nl} = \begin{cases} B(n) + W_N^n C(n), & n = 0, 1, \dots, N/2 - 1 \\ B(n) - W_N^n C(n), & n = N/2, N/2 + 1, \dots, N - 1 \end{cases}$$

$$B(n) = \sum_{k=0}^{N/2-1} x(2k)W_{N/2}^{kn} \quad C(n) = W_N^n \sum_{k=0}^{N/2-1} x(2k+1)W_{N/2}^{kn} \quad W_N^n = e^{j\frac{2\pi n}{N}} = \cos\frac{2\pi n}{N} + j\sin\frac{2\pi n}{N}$$

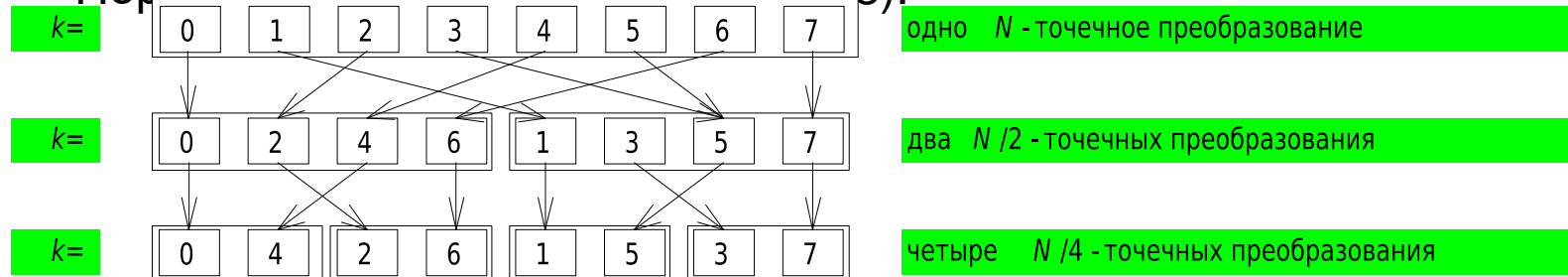
Трудоемкость при переходе от $1 \times N \rightarrow 2 \times (N/2) \rightarrow \dots \rightarrow (N/2) \times 2$

$$N^2 = 2 \times \frac{N}{2} \times 2 = \frac{N}{2} \log_2 N$$

1.6. Базовые алгоритмы ЦОС

Фурье-преобразование-2

Порядок адресации элементов ($N=8$):



Представим адреса элементов в двоичном коде и переставим биты в обратном (реверсном) порядке:

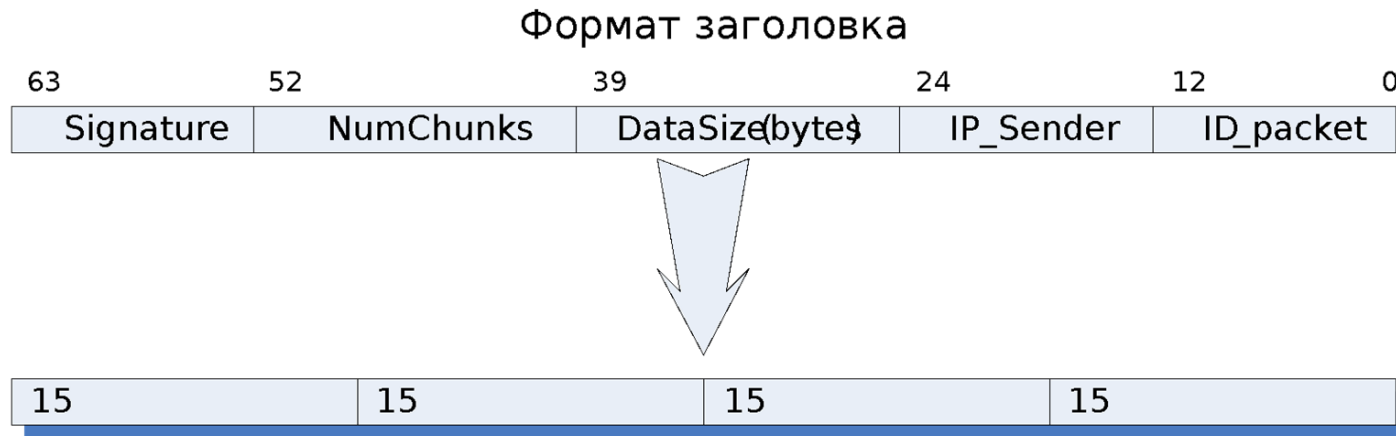
Прямой:	000	001	010	011	100	101	110	111
Реверсный:	000	100	010	110	001	101	011	111
Значение:	0	4	2	6	1	5	3	7

Для эффективной реализации необходимы:

- бит-реверсная адресация;
- дуальное сложение/вычитание;
- умножение с накоплением;
- табличная реализация функций \sin , \cos
- аппаратная поддержка циклов;

1.6. Базовые алгоритмы ЦОС

Формирование пакетов данных



Формат представления и обработки

Для эффективной реализации необходимы:

- выполнение сдвигов за одинаковое число тактов независимо от величины фактора сдвига
- депонирование и выделение битовых полей (доступ к битовым полям в пределах машинного слова

1.6. Базовые алгоритмы ЦОС

Обобщенные требования к DSP-процессорам

- выполнение операций «умножения с накоплением»;
- одновременная выборка из памяти двух операндов;
- доступ к памяти с автоматической модификацией указателя;
- циклическая организация буфера (кольцевые буферы);
- аппаратная поддержка циклов (автоматическая проверка условия выхода из цикла без потери тактов);
- дуальное сложение/вычитание;
- бит-реверсная адресация;
- повышенная точность представления операндов;
- табличная реализация элементарных функций;
- эффективный доступ к битовым полям

1.7. Форматы данных в системах ЦОС

1. Данные с фиксированной запятой (Fixed-Point Numbers)

а) целые (Integer)

- знаковые (signed)
- беззнаковые (unsigned)

б) дробные (Fractional)

- знаковые (signed)
- беззнаковые (unsigned)

2. Данные с плавающей запятой (Floating-Point Numbers)

1.7. Форматы данных в системах ЦОС

Целочисленные ФЗ-данные (N.0)

Бит	15	14	13		2	1	0
Вес	-2^{15}	2^{14}	2^{13}	...	2^2	2^1	2^0
	Знаковый бит						

Знаковое целое
(в дополнительном коде)

Двоичная точка

Бит	15	14	13		2	1	0
Вес	2^{15}	2^{14}	2^{13}	...	2^2	2^1	2^0

Беззнаковое целое

Преимущества

- простота, удобство в трактовке данных и результатов

Недостатки

- возникновение ошибки переполнения при обработке больших массивов данных;
- переполнение при умножении;
- невозможность одновременной обработки больших и малых значений;

1.7. Форматы данных в системах ЦОС

Дробные ФЗ-данные (1.N, 0.N)

Бит	15	14	13		2	1	0
Вес	-2^0	2^{-1}	2^{-2}	...	2^{-13}	2^{-14}	2^{-15}
	Знаковый бит		Знаковое дробное				
	(в дополнительном коде)						

Двоичная точка

Бит	15	14	13		2	1	0
Вес	2^{-1}	2^{-2}	2^{-3}	...	2^{-14}	2^{-15}	2^{-16}
	Беззнаковое дробное						

Преимущества

- согласованность с АЦП (соответствует доли сигнала от max);
- отсутствие переполнения при умножении;

Недостатки

- возникновение ошибки округления при выполнении операций умножения;
- невозможность одновременной обработки больших и малых значений

1.7. Форматы данных в системах ЦОС

ПЗ-данные (формат IEEE 754/854)



Преимущества

- большой динамический диапазон;
- простота использования (не требуются операции выделения и нормализации порядков);

Недостатки

- необходимость дополнительно преобразования ФЗ→ПЗ и ПЗ→ФЗ;
- меньшая точность представления данных по сравнению с ФЗ-форматами;
- сложность аппаратной реализации;
- наличие «особых» типов ПЗ-данных:

Тип	Экспонента	Мантисса	Значение
NAN	255	≠ 0	неопределено
Infinity	255	= 0	$(-1)^s * \infty$
Normal	$1 \leq e \leq 254$	любая	$(-1)^s * (1.f_{22-0})2^{e-127}$
Zero	0	0	$(-1)^s * 0$

1.7. Форматы данных в системах ЦОС

Источники возникновения ошибок при обработке данных

1. Ошибки, вызванные конечной разрядностью DSP-процессора

- ошибки округления
- ошибки переполнения

2. Распространение ошибок при выполнении математической обработки данных

1.7. Форматы данных в системах ЦОС

Ошибки округления и переполнения

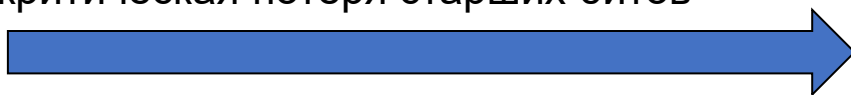
	Fractional Interpretation	Integer Interpretation
0.10	1/2	2
x 0.11	x 3/4	x 3
<hr/>		
. 010		
.010		
0.00		
<hr/>		
0.0110	3/8	6

Ошибка округления (truncation error, roundoff error) при перемножении дробных чисел – потеря младших битов



Ошибка переполнения (overflow error) при перемножении целых чисел и сложении дробных и целых чисел – критическая потеря старших битов

	Fractional Interpretation	Integer Interpretation
0.10	1/2	2
+ 0.11	+ 3/4	+ 3
<hr/>		
1.01	5/4 = -1/4	5 = -1



1.7. Форматы данных в системах ЦОС

Распространение ошибки при вычислениях

Умножение:

$$x: \text{---} \quad xy: \text{---} \quad z: \text{---} \quad x: \text{---} \quad y: \text{---}$$

Деление:

$$x: \text{---} \quad x/y: \text{---} \quad z: \text{---} \quad x: \text{---} \quad y: \text{---}$$

Сложение/вычитание:

$$z: \text{---} \quad x: \text{---} \quad y: \text{---} \quad z: \text{---} \quad \frac{x}{x: \text{---} \quad y: \text{---}} \quad x: \text{---} \quad \frac{y}{x: \text{---} \quad y: \text{---}} \quad y: \text{---}$$

Начальное значение ошибки ϵ равно ошибке квантования (АЦП) – половине веса младшего разряда DSP-процессора

1.7. Форматы данных в системах ЦОС

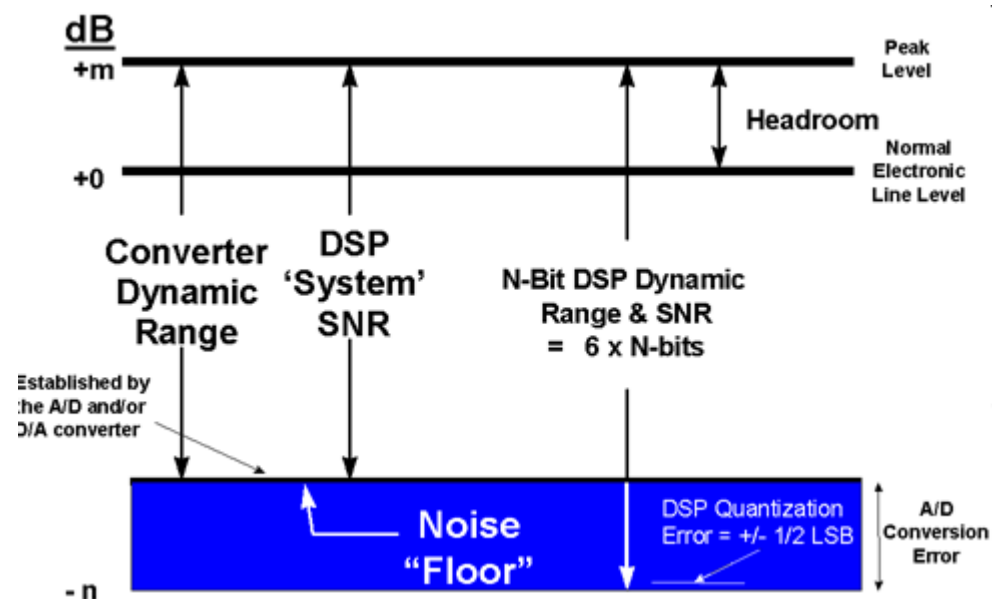
Динамический диапазон-1

Динамический диапазон – отношение максимального значения сигнала к ошибке квантования (или минимальному значению сигнала)

$$ДД(дБ) \approx 20 \lg \frac{\text{макс. значение}}{\text{мин. значение}}$$

Для цифровых систем:

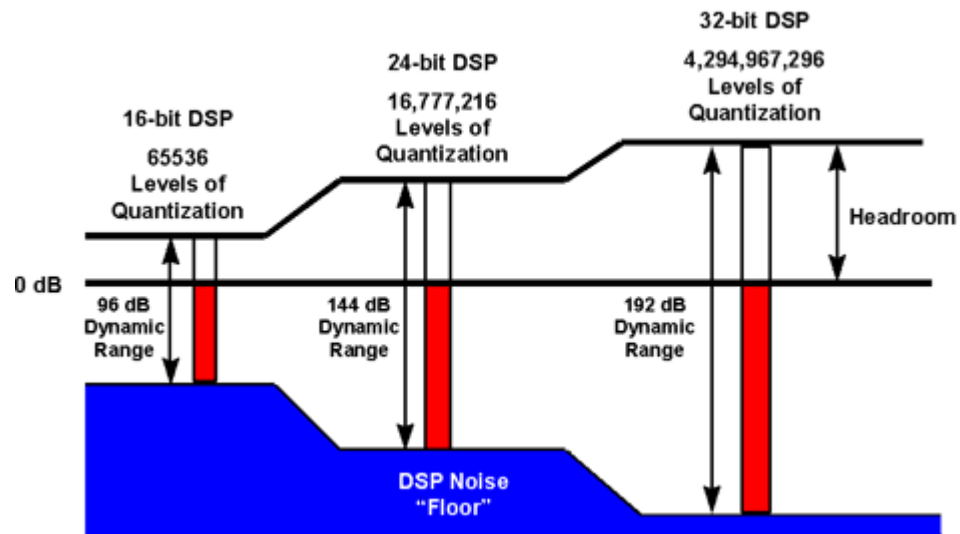
- увеличение разрядности на 1 бит приводит к увеличению ДД на 6 дБ (для ФЗ-данных);
- совпадает с отношением «сигнал/шум квантования»



1.7. Форматы данных в системах ЦОС

Динамический диапазон-2

Расширение полезного динамического диапазона при увеличении разрядности DSP-процессора



1.7. Форматы данных в системах ЦОС

Динамический диапазон-3

Способы увеличения динамического диапазона и повышения точности обработки данных в DSP-процессорах:

- увеличение разрядности процессоров (16→24 →32 бита)
- использование специального регистра-аккумулятора для хранения данных промежуточных вычислений при последовательных операциях умножения с накоплением или сложения

1.7. Форматы данных в системах ЦОС

Формат	Разрядность	Значащих битов	Вес младшего разряда	Диапазон значений (представления)	ДД, дБ
Целочисленный беззнаковый	16	16	1	$0 \leq C \leq (2^{16}-1)$ 0...65535	96,3
Целочисленный знаковый	16	15	1	$-2^{15} \leq C \leq (2^{15}-1)$ -32768...32767	90,3
Дробный беззнаковый	16	16	2^{-16} 0,000015	$0 \leq C \leq (1-2^{-16})$ 0,0...0,999985	96,3
Дробный знаковый	16	15	2^{-15} 0,000031	$-1 \leq C \leq (1-2^{-15})$ -1,0...0,999969	90,3
С плавающей точкой	32	смещ. порядок-8 мантисса-24, дробн.часть-23, неявная-1 $-126 \leq e \leq 127$	min (при $e = -126$, $m=1,00...0$) – $1,4 * 10^{-45}$ max (при $e = 127$, $m=1,00...0$) – $2 * 10^{31}$	(+/-) $1 * 2^{-126} \leq C \leq (2-2^{-23}) * 2^{127}$ $C>0$: $-3,4 * 10^{38}... -1,18 * 10^{-38}$ $C<0$: $1,18 * 10^{38}...3,4 * 10^{-38}$	1530

Архитектура и программирование DSP-процессоров

- Примеры приложений и базовые алгоритмы ЦОС.
Форматы данных
- **Обобщенная архитектура DSP-процессоров.**
Основные элементы архитектуры процессоров ADSP-21x60
- Обзор аппаратных средств ЦОС. Альтернативные средства ЦОС. Тенденции рынка DSP-процессоров
- Этапы и средства разработки ПО для DSP-процессоров
- Базовые подходы к программной реализации алгоритмов ЦОС на DSP

Хусаинов Наиль Шавкятович
к.т.н., доцент кафедры МОП ЭВМ ТРТУ
KhussainovNSh@mopevm.tsure.ru

2. Обобщенная архитектура DSP-процессоров. Основные элементы архитектуры процессоров ADSP-21x60

- 2.1. Обобщенная архитектура DSP-процессоров
- 2.2. Взаимодействие DSP-процессора с периферийными устройствами в системе ЦОС
- 2.3. Методы оценки производительности DSP-процессоров.
Критерии выбора процессоров
- 2.4. Архитектура SHARC ADSP-21060
- 2.5. Особенности архитектуры SHARC ADSP второго поколения

2.1. Обобщенная архитектура DSP-процессоров

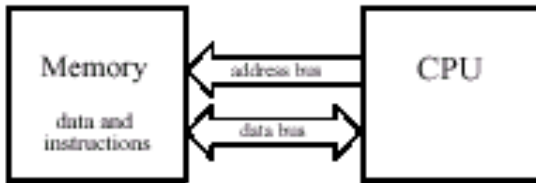
Принципиальные особенности архитектуры

- принцип «детерминированного выполнения»;
- Гарвардская архитектура;
- элементы RISC-архитектуры. Конвейеризация и сокращение длительности командного цикла;
- аппаратная реализация типовых операций ЦОС;
- специализированные команды ЦОС;
- расширенные коммуникационные возможности;

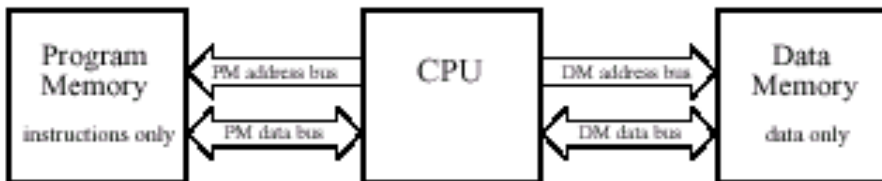
2.1. Обобщенная архитектура DSP-процессоров

Организация памяти

a. Von Neumann Architecture (*single memory*)



b. Harvard Architecture (*dual memory*)



c. Super Harvard Architecture (*dual memory, instruction cache, I/O controller*)

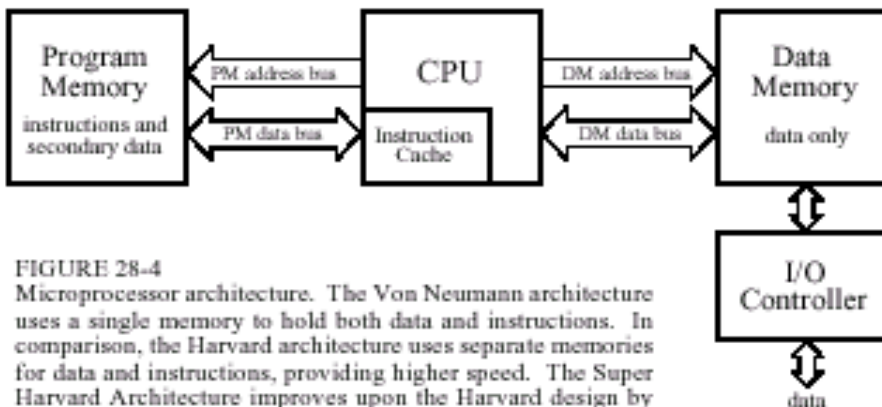
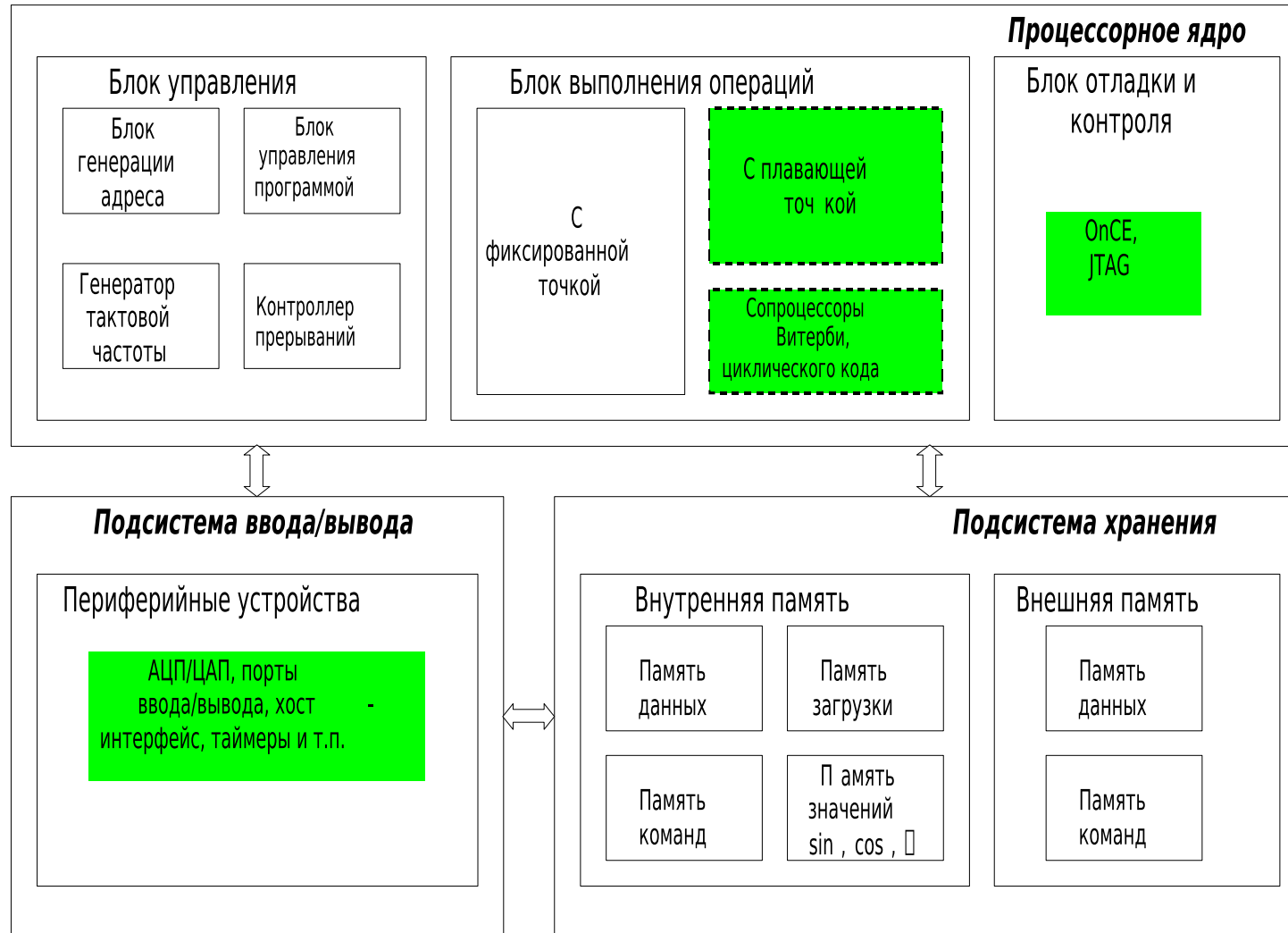


FIGURE 28-4
Microprocessor architecture. The Von Neumann architecture uses a single memory to hold both data and instructions. In comparison, the Harvard architecture uses separate memories for data and instructions, providing higher speed. The Super Harvard Architecture improves upon the Harvard design by adding an instruction cache and a dedicated I/O controller.

2.1. Обобщенная архитектура DSP-процессоров



2.1. Обобщенная архитектура DSP-процессоров

Дополнительные элементы архитектуры

1. Генераторы адресов данных для каждого пространства памяти (поддержка работы с несколькими буферами, кольцевые буферы, бит-реверсная адресация, пред/пост-модификация указателя);
2. Элементы RISC-архитектуры (конвейерный режим, сокращенное число способов адресации операндов, расширенный регистровый файл, «теневые» регистры, аппаратная поддержка циклов);
3. Элементы VLIW-архитектуры (одновременное выполнение операций различными блоками. Распараллеливание – на этапе программирования);
4. SIMD-обработка;
5. Встроенные средства поддержки отладки в реальном времени (JTAG-интерфейс)

2.1. ОБОБЩЕННАЯ АРХИТЕКТУРА DSP-ПРОЦЕССОРОВ

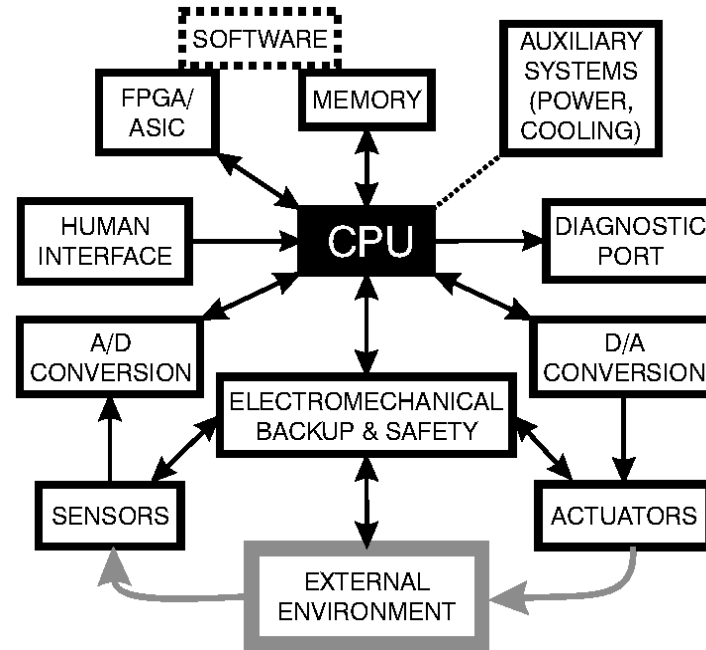
.Основные подходы к снижению энергопотребления:

- понижение уровня напряжения питания, программное управление делителем тактовой частоты, программное отключение отдельных вычислительных блоков;

- управление питанием путем перевода процессора в один из режимов:

- fully operational
- standby mode
- clock-off mode

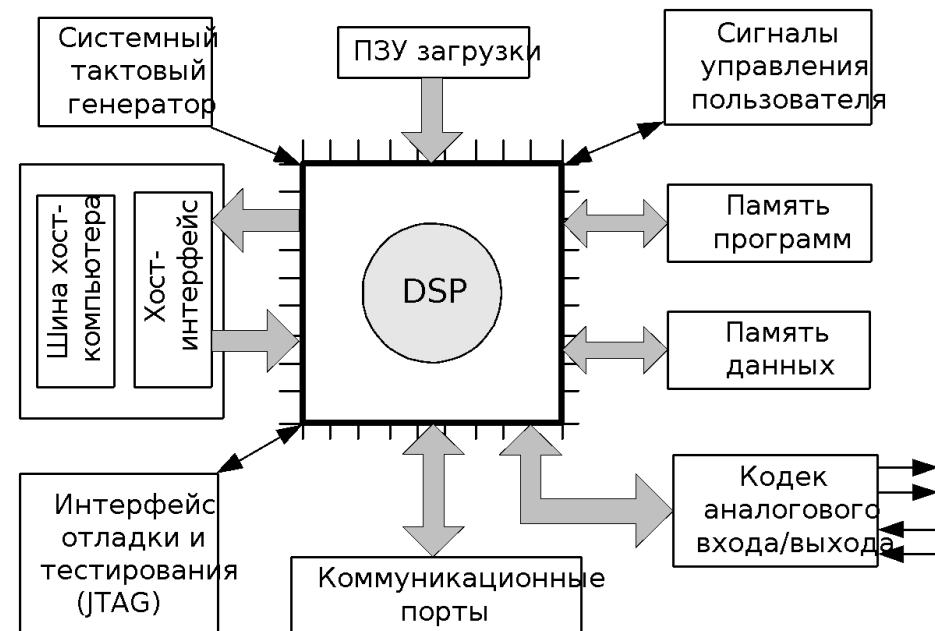
2.2. Процессор во встраиваемой системе



Требования:

- возможность поддержки реального времени (для конкретного приложения или класса приложений);
- интерфейсы с внешними устройствами;
- вес, габариты, энергопотребление, цена

2.2. DSP-процессор в системе ЦОС



Одновременно:

- синхронная обработка данных - ввод/вывод «потокowych» данных для обработки (с АЦП/ЦАП);
- асинхронная обработка данных – реакция на сигналы хост-ЭВМ или пользователя;
- взаимодействие с другими DSP в многопроцессорной системе;
- отладка в реальном времени

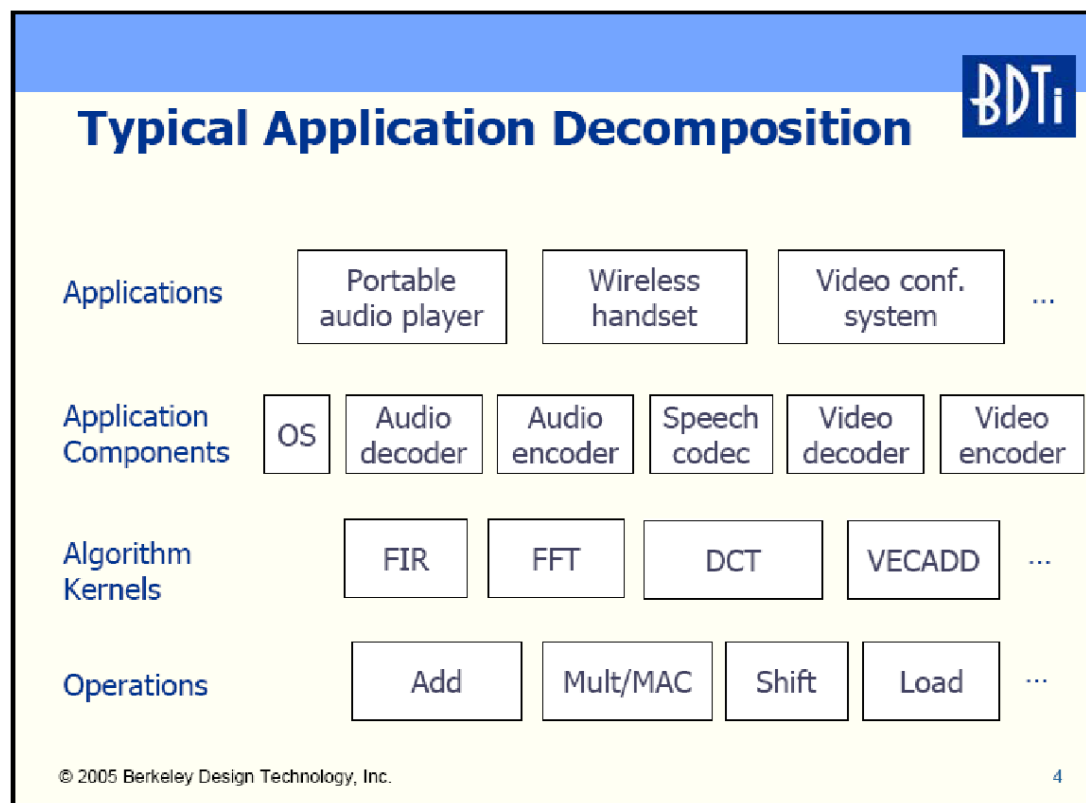
2.3. Оценка производительности DSP-процессоров

Подходы к оценке производительности

- тактовая частота/энергопотребление;
- количество вычислительных инструкций/операций в единицу времени (MIPS, MFLOPS, MOPS, MMACS);
- длительность выполнения типовых алгоритмов ЦОС: фильтрация, ДПФ, декодер Витерби и т.д. (Kernel Algorithm Benchmark);
 - оценки на *наборах* тестов (BDTImark2000 и др.);
 - оценки на типовых приложениях (Full Application Benchmark)

2.3. Оценка производительности DSP-процессоров

Связь между различными уровнями оценок производительности DSP-процессоров



2.3. Оценка производительности DSP-процессоров

Оценки производительности на наборах тестов BDTImark2000 и BDTImemmark2000 (pdf)

Функция	Описание функции	Область практического применения
КИХ-фильтр (FIR) в области действительных чисел	Фильтр с конечной импульсной характеристикой, выполняется над блоком данных	Обработка речи, например кодек G.728
КИХ-фильтр (FIR) в области комплексных чисел	То же, но работа с комплексными данными	Компенсация ошибок канала связи в модемах
БИХ-фильтр (IIR)	Фильтр с бесконечной импульсной характеристикой	Аудиообработка, фильтрация
Векторное произведение	Сумма поэлементных произведений элементов векторов	Свертка, корреляция, перемножение матриц, многомерная обработка сигналов
Сложение векторов	Поэлементное сложение элементов векторов, результат – вектор	Графика
Поиск максимального значения	Поиск положения максимального элемента в векторе	Контроль ошибок кодирования и передачи данных
Декодер Витерби	Декодирование блока битов, закодированных с использованием сверточного кодирования	Контроль ошибок кодирования и передачи данных
Управление	Последовательность управляющих инструкций (проверка условия, ветвление, работа со стеком, манипуляции с битами)	Практически все DSP-приложения
256- или 1024-точечное БПФ	Преобразование сигнала из временной области в частотную	Радары, сонары, MPEG-аудиокодирование, спектральный анализ
Распаковка битов	Распаковка данных переменной длины из битового потока	Аудио/видео декомпрессия, реализация протоколов связи

2.3. ОЦЕНКА ПРОИЗВОДИТЕЛЬНОСТИ DSP-ПРОЦЕССОРОВ

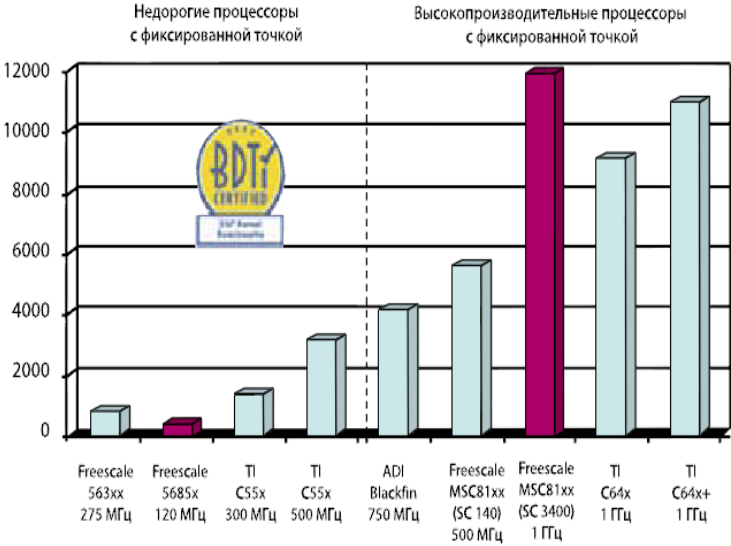


Рис. 1. Оценки BDTmark2000 для DSP с фиксированной точкой

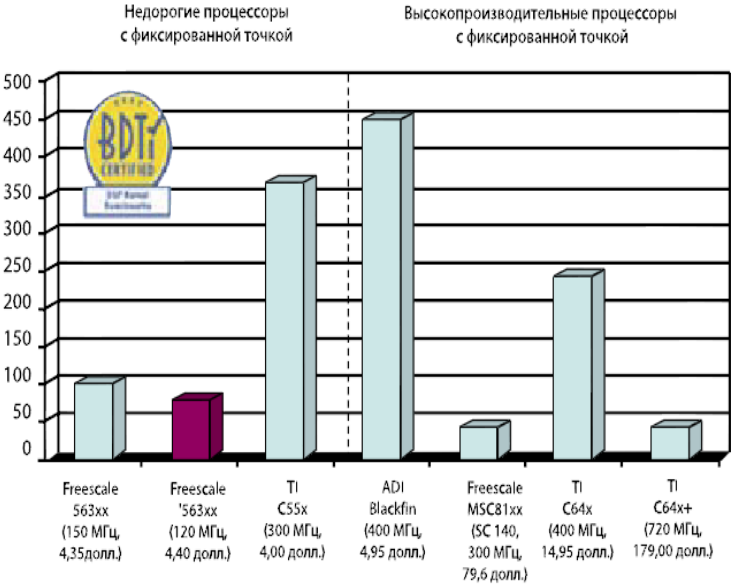


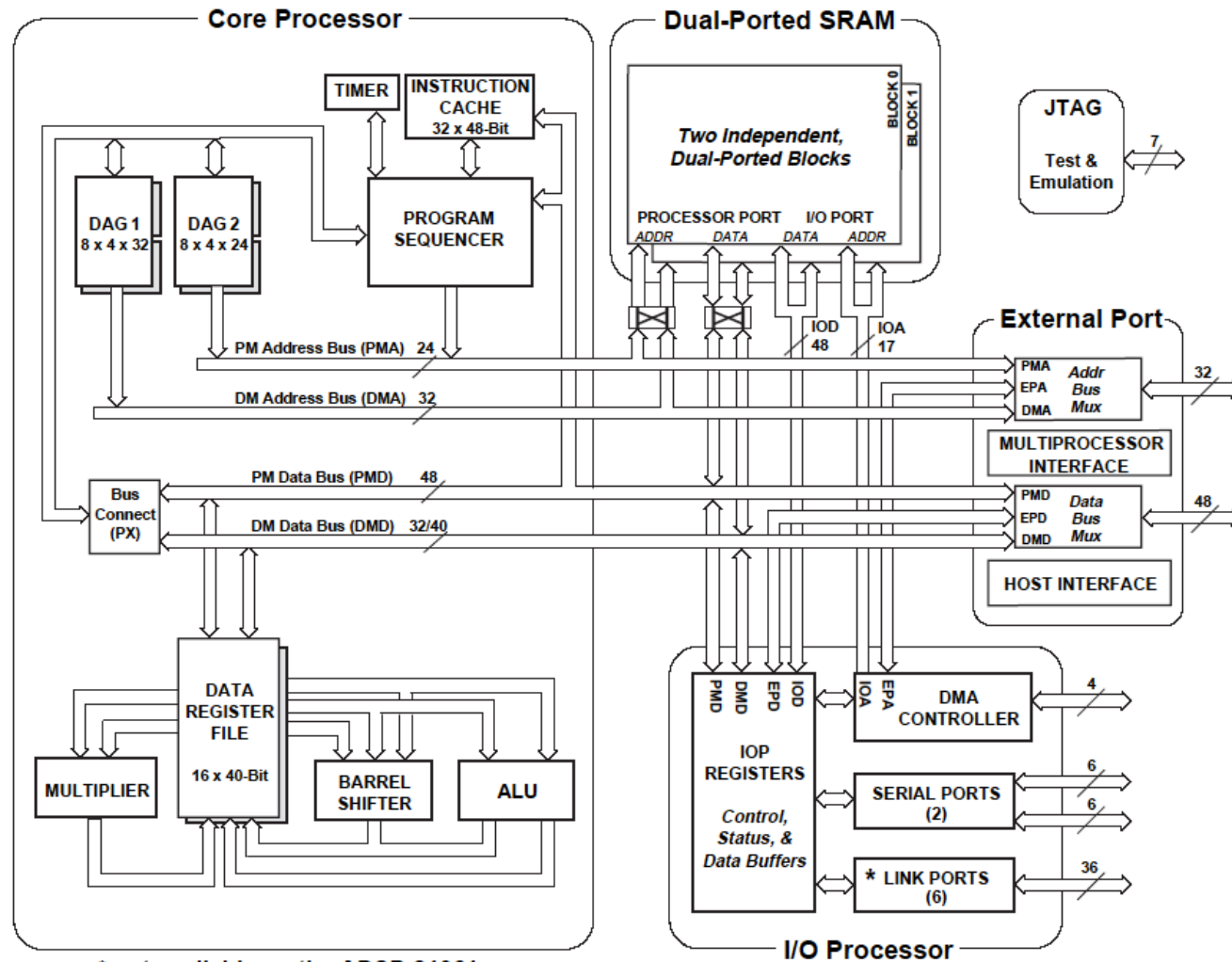
Рис. 2. Оценки BDTmark2000 для DSP с фиксированной точкой, приведённые для самых экономически

2.3. Оценка производительности DSP-процессоров

При выборе процессора следует учитывать:

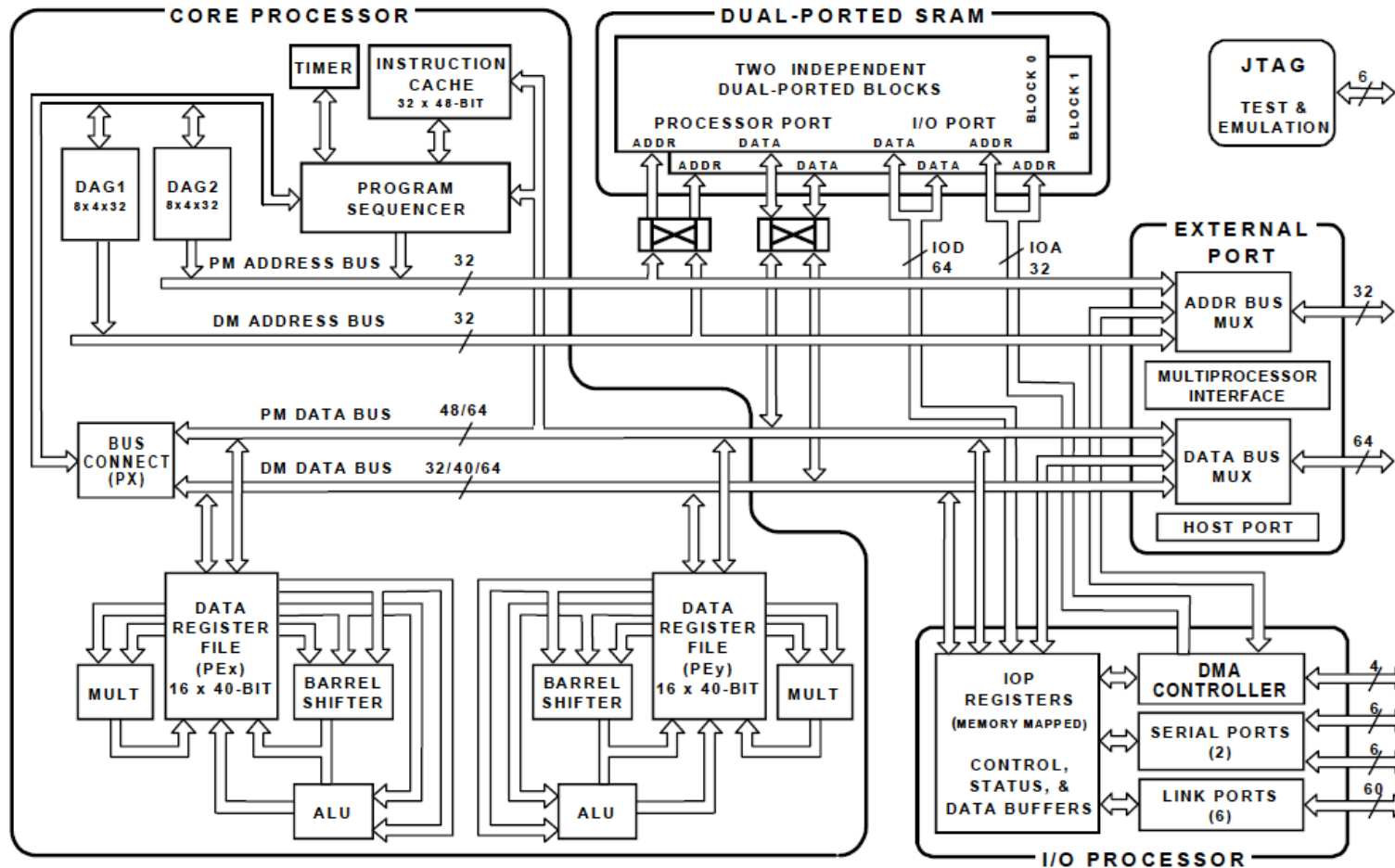
- все ПЗ-процессоры поддерживают обработку ФЗ-данных;
- в ПЗ-числах шкала квантования неравномерна в отличие от ФЗ-чисел;
- использование ПЗ-процессоров облегчает и ускоряет процесс разработки программного обеспечения, повышает его надежность;
- при использовании ПЗ-арифметики требуются дополнительные операции для преобразования ФЗ->ПЗ при вводе данных и ПЗ->ФЗ при их выводе;
- применение ЦСП с ФЗ-обработкой необходимо в системах, ориентированных на массовое коммерческое производство, когда даже незначительная разница в цене может способствовать успеху или неудаче изделия на рынке. Напротив, ПЗ-процессоры следует использовать там, где необходимо добиться более высокой производительности и точности даже за счет более высокой цены изделия;
- обычно системы с ФЗ-процессорами имеют более низкое энергопотребление, что особенно актуально, например, в сотовой телефонии;
- примеры применения ФЗ-процессоров – звуковые платы, мини-АТС (обработка голоса), сотовая телефония, системы управления на уровне микроконтроллеров и т.п.;
- примеры применения ПЗ-процессоров – устройства коммутации пакетов в коммуникационных сетях, устройства обработки медиаданных (телевидение, многоканальная обработка звука, спецэффекты и т.п.), многопроцессорные системы.

2.4. Семейство процессоров SHARC ADSP



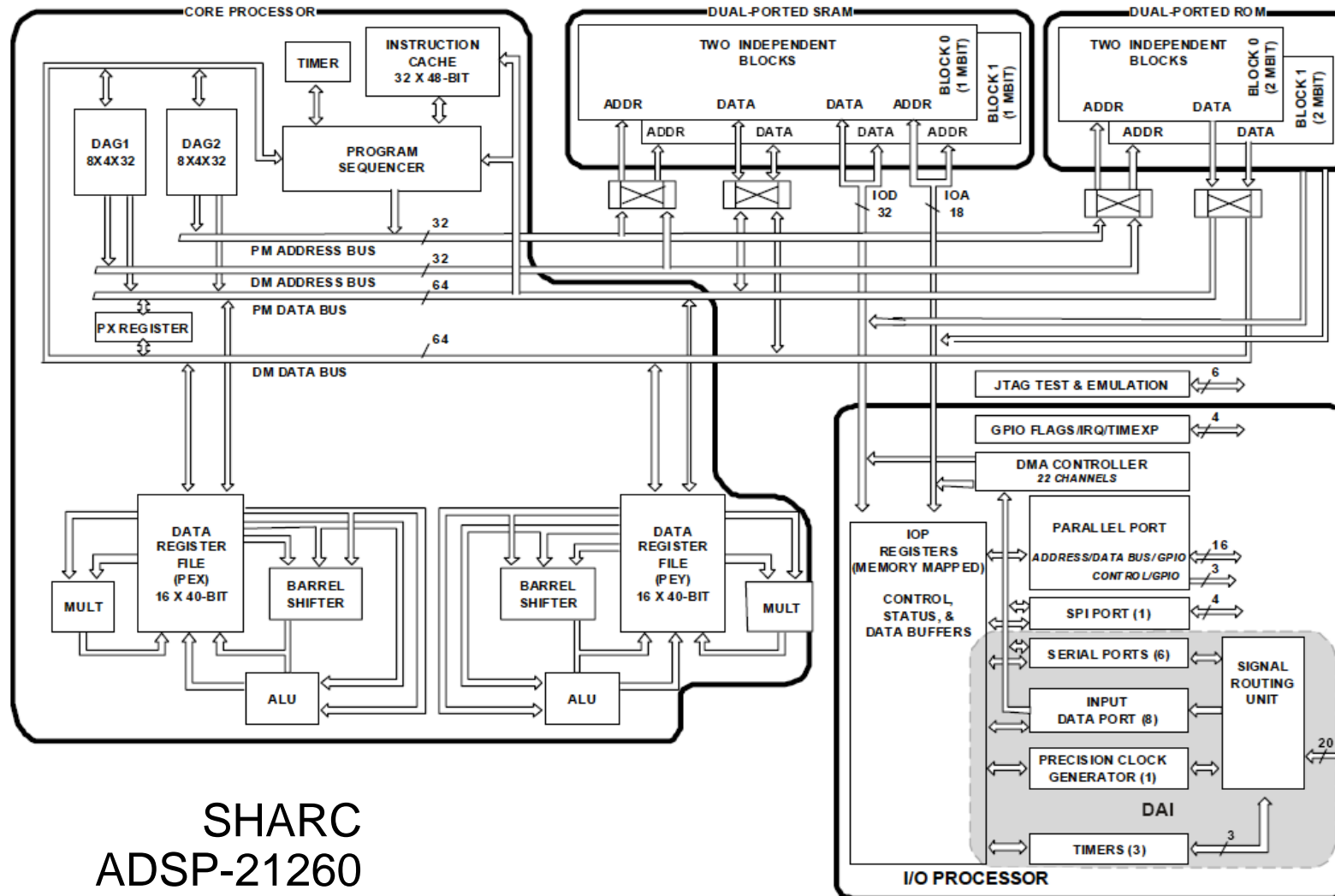
SHARC
ADSP-21060

2.5. Семейство процессоров SHARC ADSP



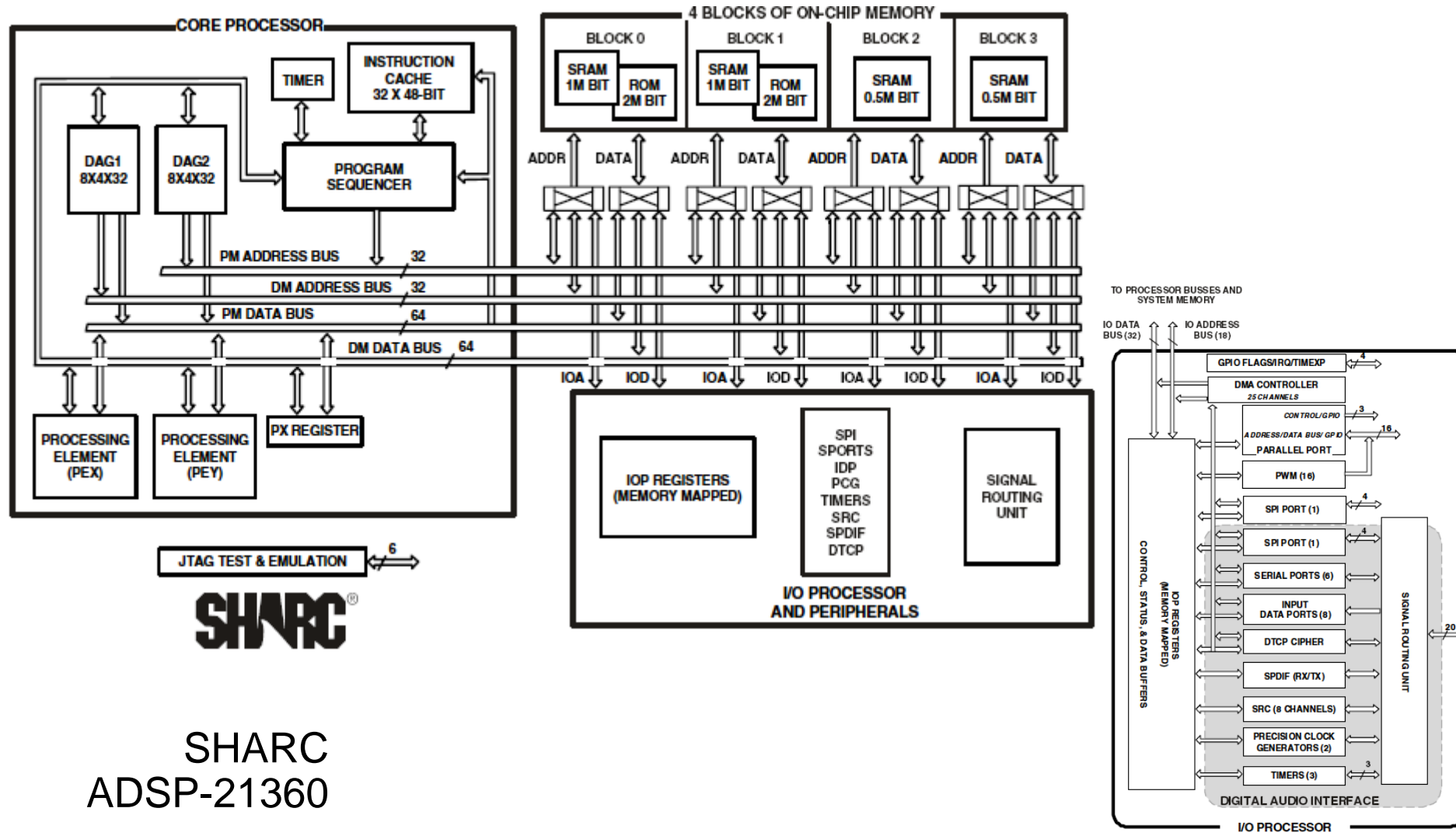
SHARC
ADSP-21160

2.5. Семейство процессоров SHARC ADSP

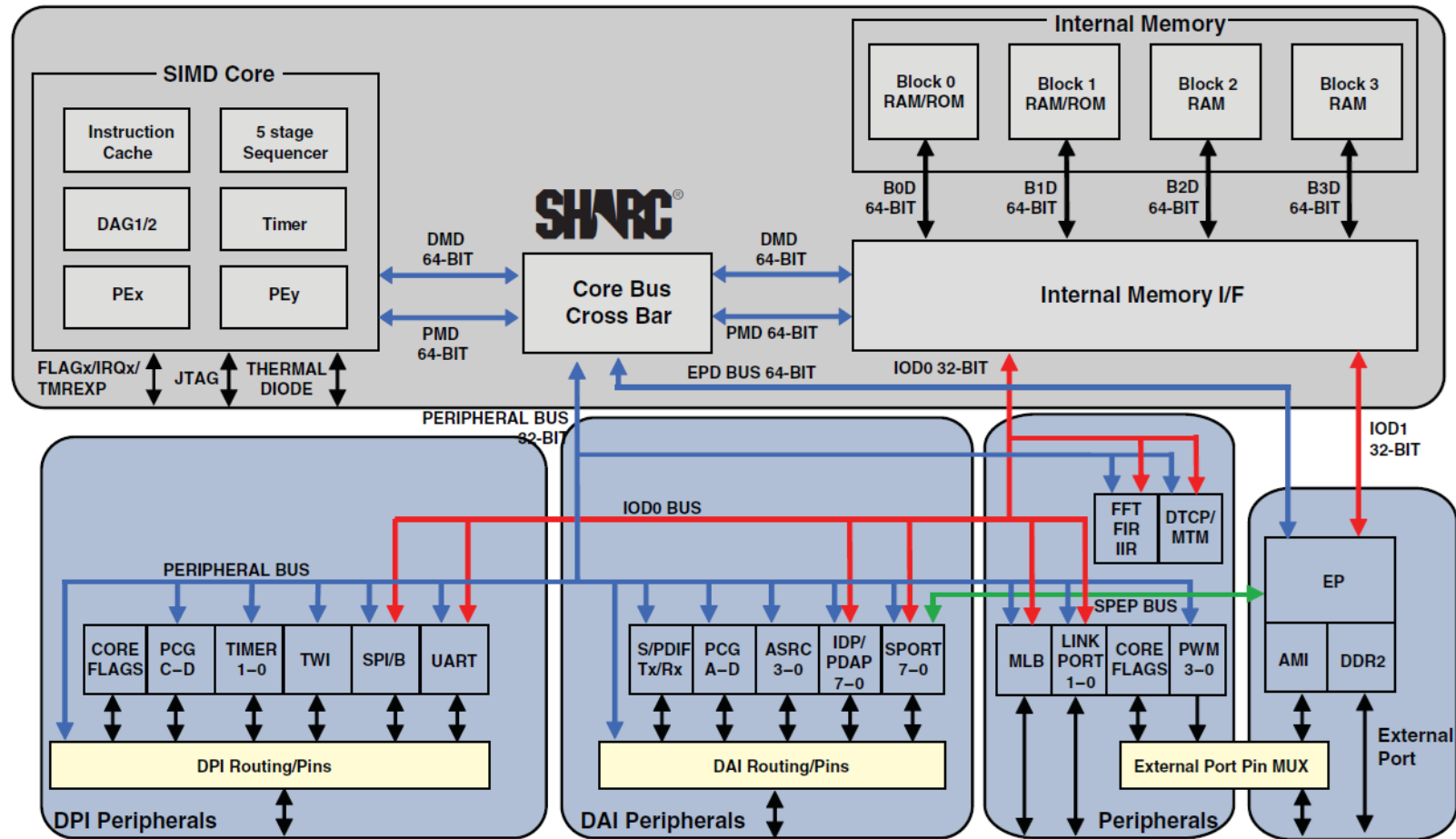


SHARC
ADSP-21260

2.5. Семейство процессоров SHARC ADSP



2.5. Семейство процессоров SHARC ADSP



SHARC
ADSP-21460

Архитектура и программирование DSP-процессоров

- Примеры приложений и базовые алгоритмы ЦОС.
Форматы данных
- Обобщенная архитектура DSP-процессоров. Основные элементы архитектуры процессоров ADSP-21х60
- **Обзор аппаратных средств ЦОС. Альтернативные средства ЦОС. Тенденции рынка DSP-процессоров**
- Этапы и средства разработки ПО для DSP-процессоров
- Базовые подходы к программной реализации алгоритмов ЦОС на DSP

Хусаинов Наиль Шавкятович
к.т.н., доцент кафедры МОП ЭВМ ТРТУ
KhussainovNSh@mopevm.tsure.ru

3. Обзор аппаратных средств ЦОС. Альтернативные средства ЦОС. Тенденции рынка DSP-процессоров

3.1. Классификация устройств ЦОС

3.2. Тенденции развития рынка устройств ЦОС

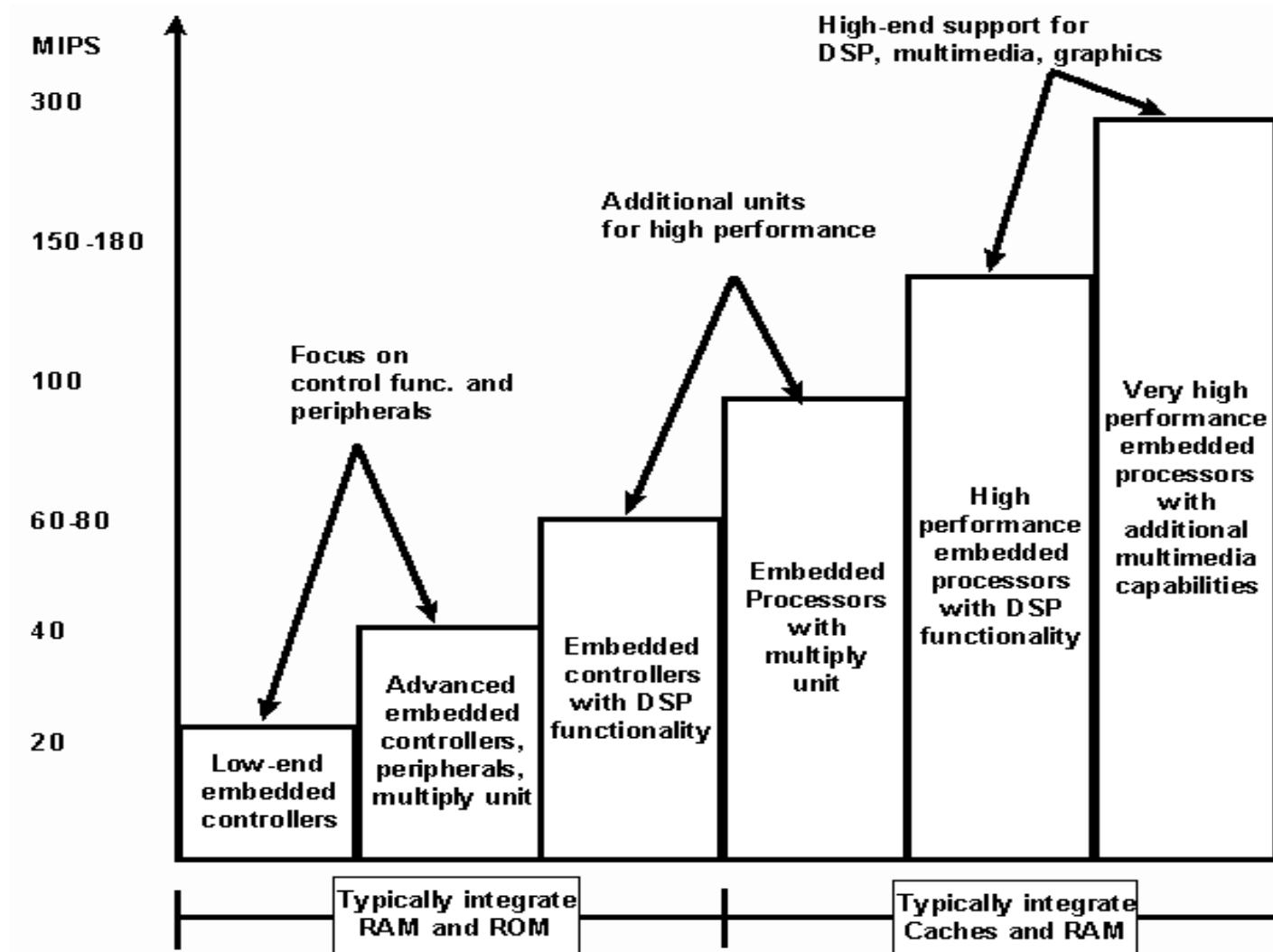
3.3. Основные разработчики DSP-процессоров. Линейки DSP-процессоров Texas Instruments и Analog Devices

ПРОЦЕССОРЫ ДЛЯ ВСТРАИВАЕМЫХ СИСТЕМ

- Условная классификация по типу архитектуры и решаемым задачам*:
 - специализированные процессоры (процессоры ЦОС, сетевые процессоры, процессоры/контроллеры для мобильных устройств)
 - процессоры «общего» назначения (традиционные процессоры, используемые во встраиваемых системах)

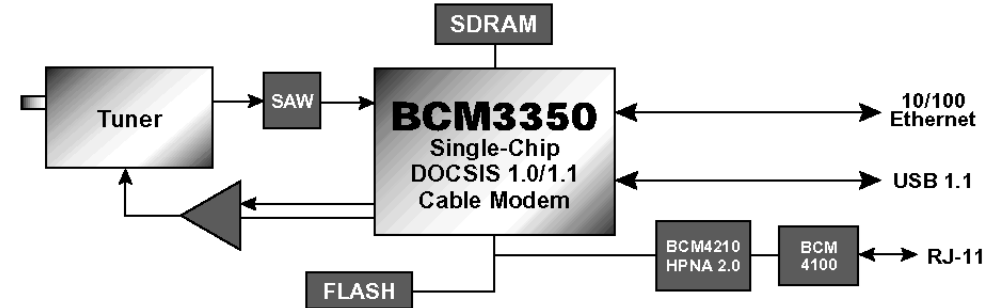
* Граница между процессорами условна

ПРОЦЕССОРЫ ДЛЯ ВСТРАИВАЕМЫХ СИСТЕМ

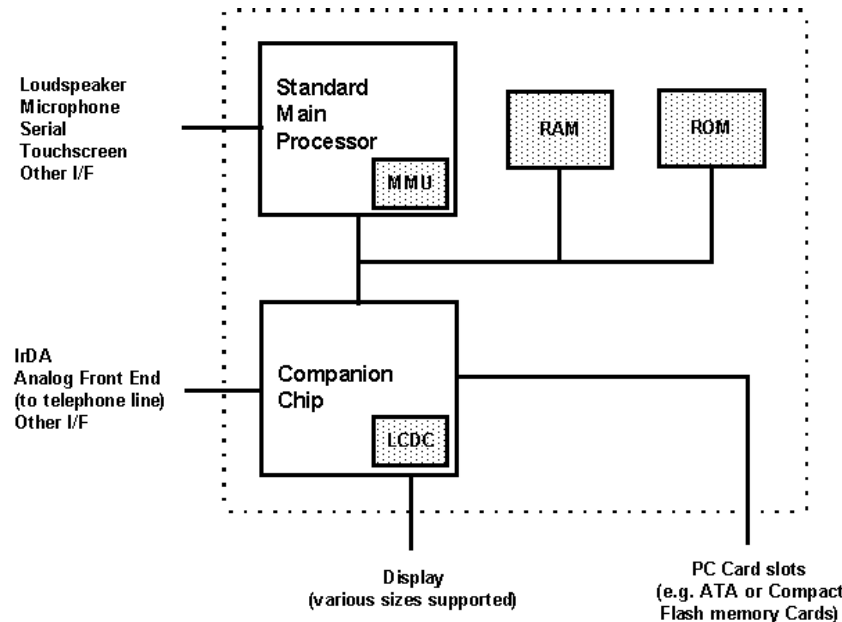


ПОДХОДЫ К ИНТЕГРАЦИИ ПЕРИФЕРИИ С ПРОЦЕССОРОМ

Базовое процессорное ядро, на основе которого создаются несколько моделей процессоров с различными наборами периферии (под различные типовые приложения)



Пример: Broadcom BCM3350 – готовое решение для кабельного модема



- «Типовой» процессор используется с процессором-компаньоном (чипсет), отвечающим за решение специфических задач

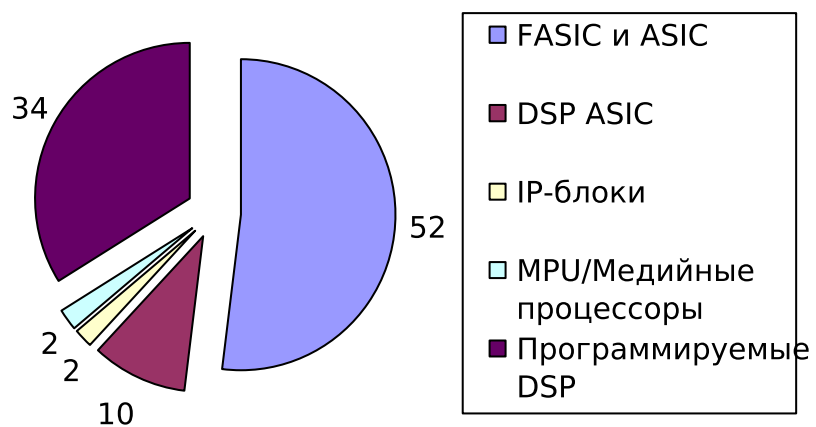
Пример архитектуры «наладонника» на базе чипсета с основным процессором (управление, вычисления), и «компаньоном» (взаимодействие с LCD, IR, Touchscreen)

3.1. Классификация устройств ЦОС

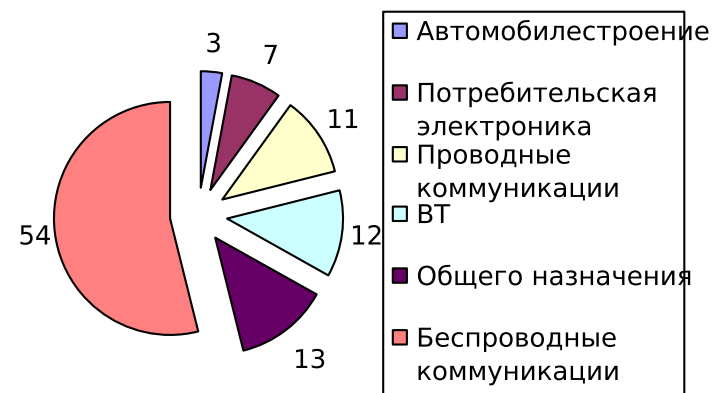
1. Специализированные БИС и СБИС (ASIC или FASIC)
2. Устройства на базе программируемых ПЛИС (FPGA)
3. IP-блоки
4. MPU/Media-процессоры
5. Микроконтроллеры (MCU)
6. DSP-процессоры
7. RISC-процессоры общего назначения с аппаратными модулями обработки сигналов (Intel: MMX, Motorola G4: AltiVec)

3.2. Тенденции развития рынка устройств ЦОС

Рост рынка устройств ЦОС – 20-30% в год



а) Устройства ЦОС



б) Программируемые DSP-процессоры

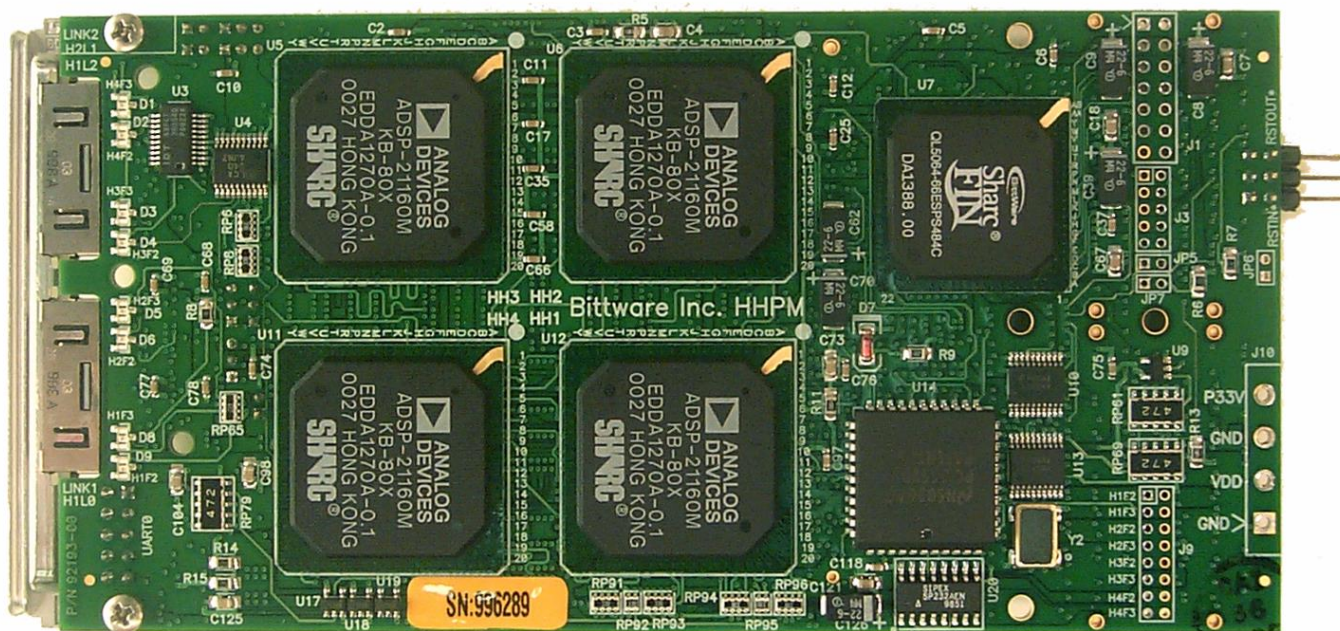
3.2. Тенденции развития рынка устройств ЦОС

Способы распространения DSP-процессоров

- IP-ядра (IP-core), включающие вычислительные блоки, генераторы адресов данных и программный секвенсор (обычно в виде схем)
- готовые процессоры
- готовые решения и платы с одним или несколькими DSP, памятью, портами расширения с возможностью подключения к ЭВМ (board level)

3.2. Тенденции развития рынка устройств ЦОС

Пример решения от Third Parties



Hummerhead 56U-VME (Bittware Inc.)

8 x SHARC ADSP-21160 80 МГц (2 кластера по 4 DSP), VME, 3840 MFLOPS, 512 Мб SDRAM, 4Мб FLASH, 16 линк-портов 80 Мбит/с, 2 последовательных порта по 40 Мбит/с

3.2. Тенденции развития рынка устройств ЦОС

- гибриды MCU/DSP (ADI Blackfin, TMS320C2000);
- медийные/сигнальные расширения системы команд (MMX, SSE,...);
- обеспечение наилучшего соотношения между производительностью, энергопотреблением и стоимостью – пока у DSP-процессоров;
- развитие средств разработки и портирования ПО для DSP-процессоров, в том числе различных семейств

3.3. Основные разработчики DSP-процессоров

<i>Компании-лидеры Company Name</i>		<i>Доля рынка DSP (2006)</i>
1	Texas Instruments	54,3%
2	Freescale Semiconductor	14,1%
3	Analog Devices	8,0%
4	Philips Semiconductors	7,5%
5	Agere Systems	7,3%
6	Toshiba	4,9%
7	DSP Group	2,2%
8	NEC Electronics	0,6%
9	Fujitsu	0,4%
10	Intersil	0,3%
	Other Companies	0,5%

Архитектура и программирование DSP-процессоров

- Примеры приложений и базовые алгоритмы ЦОС.
Форматы данных
- Обобщенная архитектура DSP-процессоров. Основные элементы архитектуры процессоров ADSP-21х60
- Обзор аппаратных средств ЦОС. Альтернативные средства ЦОС. Тенденции рынка DSP-процессоров
- **Этапы и средства разработки ПО для DSP-процессоров**
- Базовые подходы к программной реализации алгоритмов ЦОС на DSP

Хусаинов Наиль Шавкятович
к.т.н., доцент кафедры МОП ЭВМ ТРТУ
KhussainovNSh@mopevm.tsure.ru

4. Этапы и средства разработки ПО для DSP-процессоров



- 4.1. Этапы проектирования системы ЦОС на базе DSP-процессора
- 4.2. Выбор средств разработки ПО для DSP-процессоров
- 4.3. Типовой набор средств разработки ПО
- 4.4. Средства отладки и тестирования ПО
- 4.5. Среда разработки VisualDSP++

4.1. Этапы проектирования системы ЦОС на базе DSP-процессора



Этапы
проектирован
ия системы
ЦОС



4.2. Выбор средств разработки ПО для DSP-процессоров

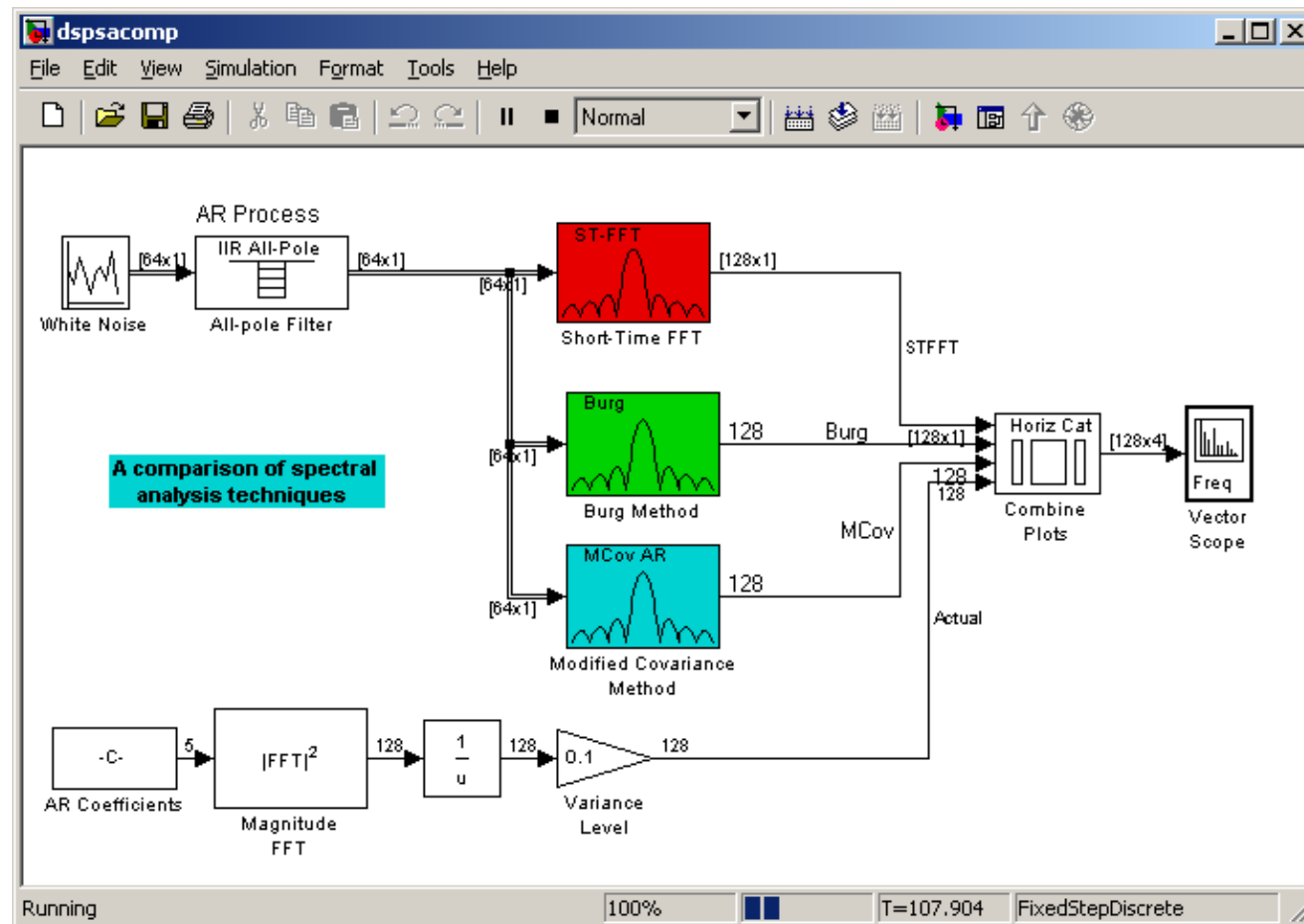
Распространенные средства реализации алгоритмов (логики) обработки

Средство разработки	«+»	«-»
Matlab (LabView, HyperSignal)	простота, наглядность, наличие инструментов для моделирования процессов обработки в масштабе времени близком к реальному (Simulink), средств генерации и отладки кода для DSP-процессоров (DSP-Developer на целевой архитектуре;	высокая цена, трудность низкоуровневой отладки, ресурсоемкость, недостаточная гибкость, низкая эффективность кода
языки высокого уровня (C/C++);	значительная платформенно-независимость, «читаемость» кода, сокращение сроков разработки ПО, наличие оптимизирующих компиляторов	обычно не удается получить предельную производительность и минимальный расход памяти
языки низкого уровня (ASM)	максимальная эффективность, минимальные требования к DSP-процессору – самое дешевое hardware	непереносимость кода, длительный срок разработки и отладки и др.



4.2. Выбор средств разработки ПО для DSP-процессоров

Рабочий экран Matlab Simulink



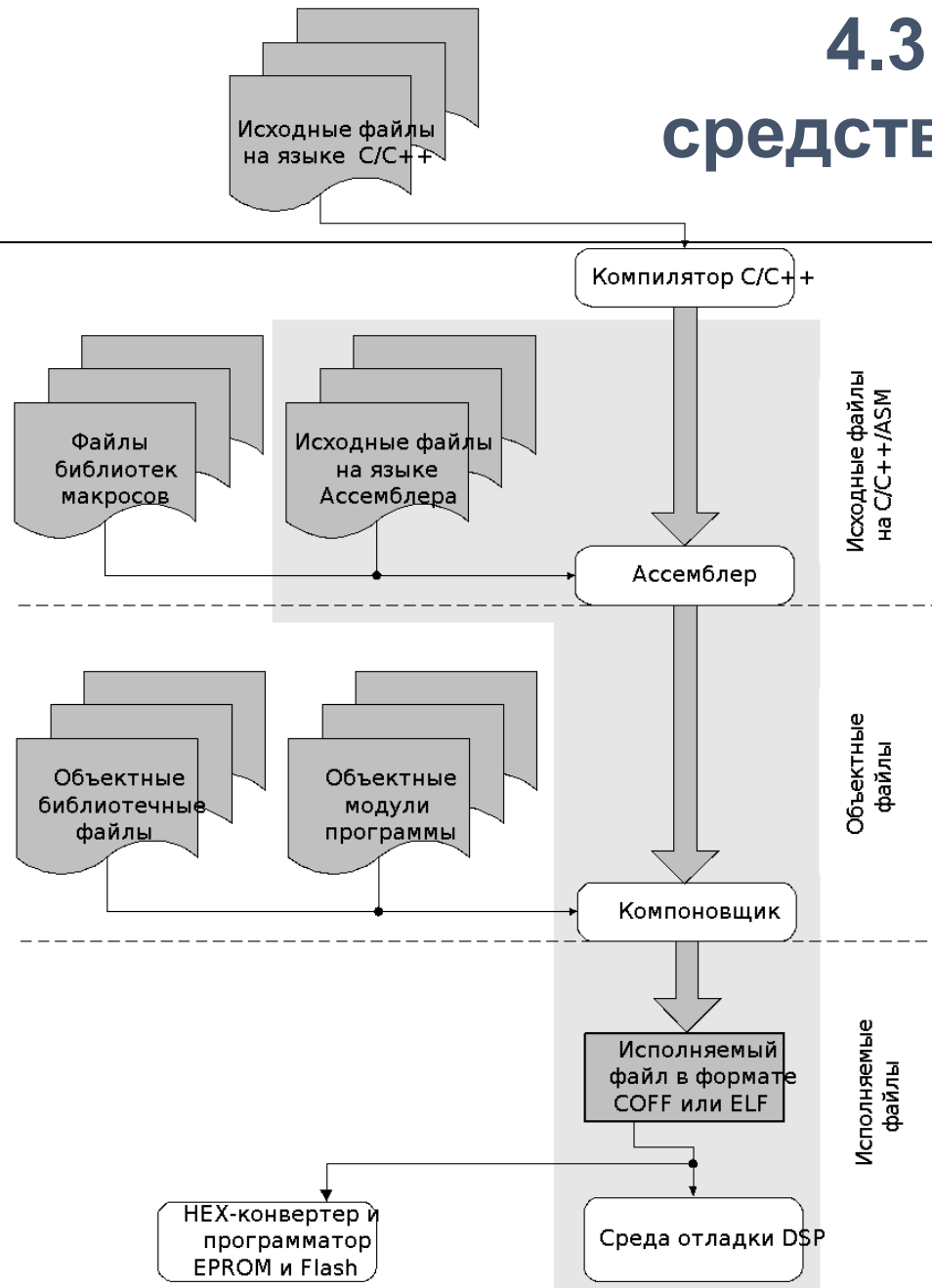
4.2. Выбор средств разработки ПО для DSP-процессоров

Отличия в производительности программного кода на Ассемблере и С

Текст на Си	Неоптимизированный код на Ассемблере (время выполнения $4*LEN + 4$)	Оптимизированный код на Ассемблере (время выполнения $((LEN-2)+4)+4$)
00x #define LEN 20 00x float dm x[LEN]; 00x float pm y[LEN]; 00x float result; 00x 001 main() 002 003 { 004 int n; 005 float s; 006 for (n=0;n<LEN;n++) 007 s += x[n]*y[n]; 008 result = s 009 }	001 i12 = _y; 002 i4 = _x; 003 004 lcntr = 20, do (pc,4) until lce; 005 f2 = dm(i4,m6); 006 f4 = pm(i12,m14); 007 f8 = f2*f4; 008 f12 = f8 + f12; 009 010 dm(_result) = f12;	001 i12 = _y; 002 i4 = _x; 003 004 f2 = dm(i4,m6), f4 = pm(i12,m14) 005 f8 = f2*f4, f2 = dm(i4,m6), f4 = pm(i12,m14); 006 007 lcntr = 18, do (pc,1) until lce; 008 f12 = f8 + f12, f8 = f2*f4, f2 = dm(i4,m6), f4 = pm(k12,m14); 009 010 f12 = f8 + f12, f8 = f2*f4; 011 f12 = f8 + f12; 012 013 f12 = f8 + f12;



4.3. Типовой набор средств разработки ПО



- текстовый редактор;
- компилятор;
- ассемблер;
- компоновщик (Linker);
- архиватор (Archiver);
- загрузчик (Loader);
- сплиттер (Hex-конвертер);
- программатор Flash-памяти

IDDE



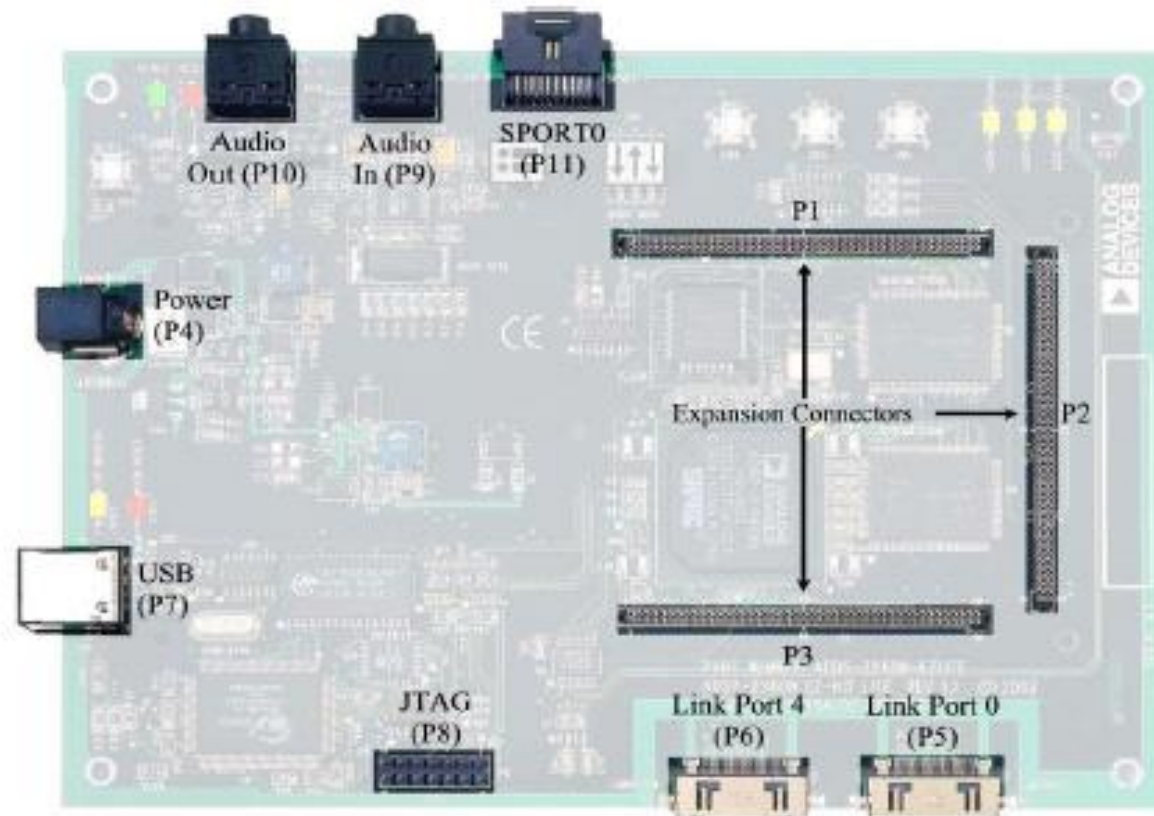
4.4. Средства отладки и тестирования ПО

- симулятор;
- типовая оценочная плата (EZLAB, EZKIT-Lite);
- сканирование DSP-процессора в целевой архитектуре через JTAG-интерфейс в режиме реального времени



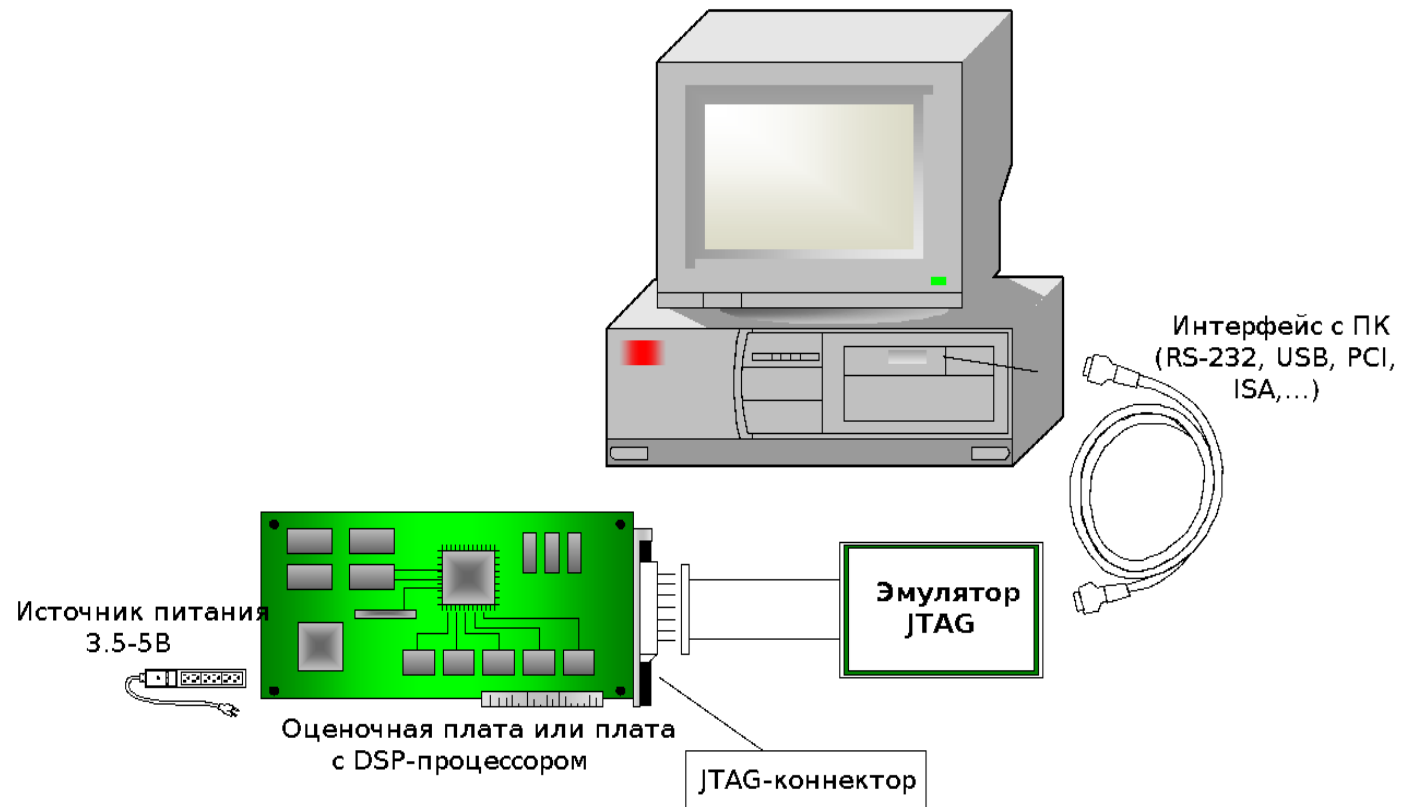
4.4. Средства отладки и тестирования ПО

Типовая оценочная плата EZKIT-Lite



4.4. Средства отладки и тестирования ПО

Схема подключения JTAG-эмулятора



4.4. Средства отладки и тестирования ПО

Преимущества и недостатки способов отладки

Способ отладки	«+»	«-»
Программный симулятор (начальный этап)	Простота, минимальная стоимость, возможность оценить логическую правильность алгоритма	Низкая производительность, особенно при больших объемах кода, трудности в имитации взаимодействия с внешними устройствами
Оценочная плата	Исполнение программы в реальном масштабе времени. Оценка правильности взаимодействия процессора с типовой периферией (кодеками и внешней памятью). Приемлемая стоимость. Идеально для обучения	Невозможность отладки в реальном времени (только в моменты останова). Ограничения в использовании внешних устройств
JTAG-эмулятор (конечный этап)	Максимальная эффективность. Отладка на целевой архитектуре.	Высокая стоимость интерфейса эмулятора



4.5. Среда разработки VisualDSP++

1. Создание проекта
3. Выбор сессии
3. Описание целевой архитектуры (файл описания линкера)
4. Оконный интерфейс в режиме отладки программы на ASM
5. Создание и отладка проекта с VDK



Архитектура и программирование DSP-процессоров

- Примеры приложений и базовые алгоритмы ЦОС.
Форматы данных
- Обобщенная архитектура DSP-процессоров. Основные элементы архитектуры процессоров ADSP-21х60
- Обзор аппаратных средств ЦОС. Альтернативные средства ЦОС. Тенденции рынка DSP-процессоров
- Этапы и средства разработки ПО для DSP-процессоров
- **Базовые подходы к программной реализации алгоритмов ЦОС на DSP**

Хусаинов Наиль Шавкятович
к.т.н., доцент кафедры МОП ЭВМ ТРТУ
KhussainovNSh@mopevm.tsure.ru

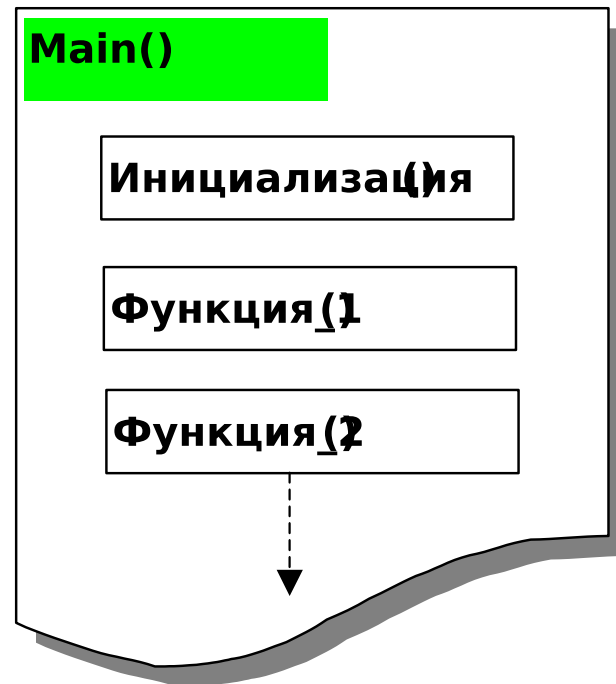
5. Базовые подходы к программной реализации алгоритмов ЦОС на DSP



- 5.1. Принципы разработки ПО для систем реального времени
- 5.2. Способы организации ввода-вывода данных
- 5.3. Организация буферов и доступа к данным при обработке сигналов

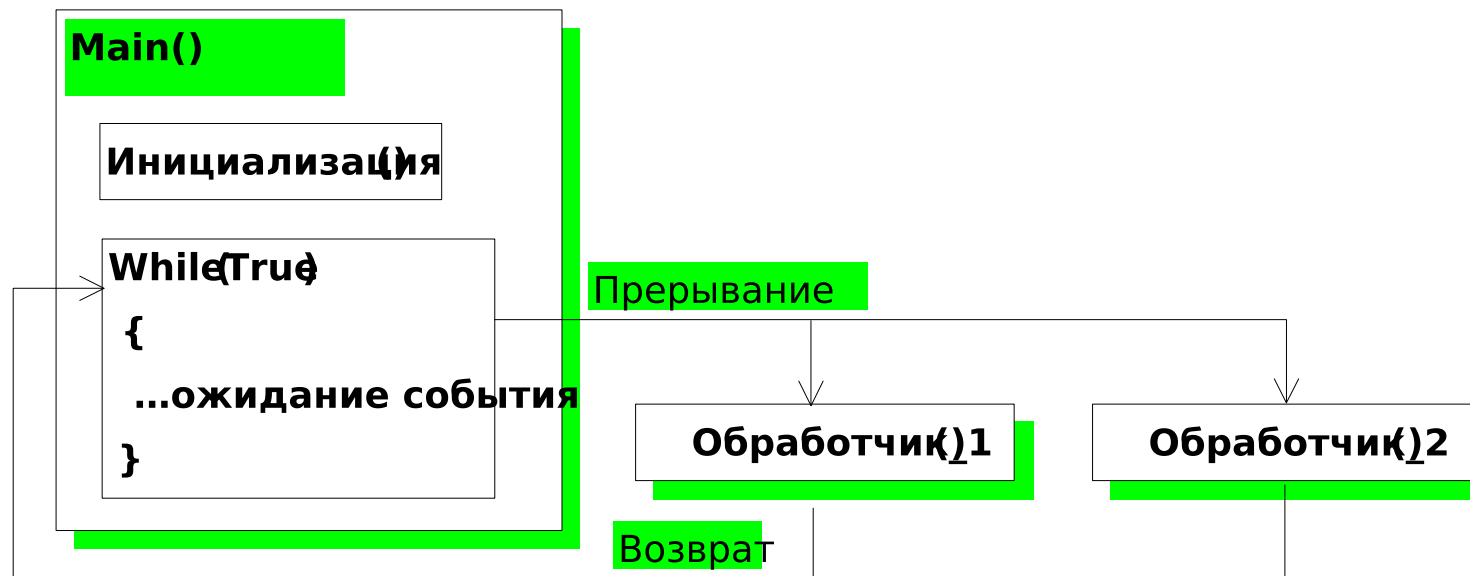
5.1. Принципы разработки ПО для систем реального времени

Традиционный структурный подход к разработке программ



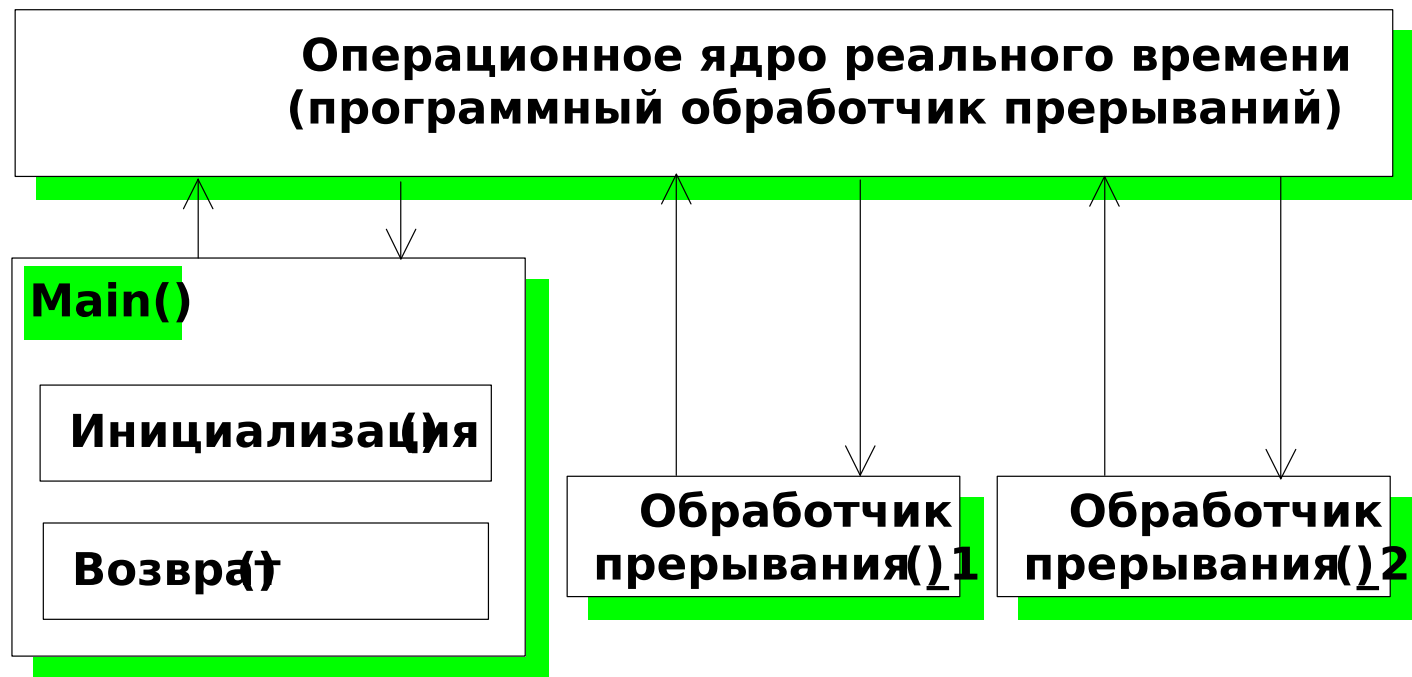
5.1. Принципы разработки ПО для систем реального времени

Принцип функционирования приложения реального времени



5.1. Принципы разработки ПО для систем реального времени

Принцип функционирования программного ядра реального времени



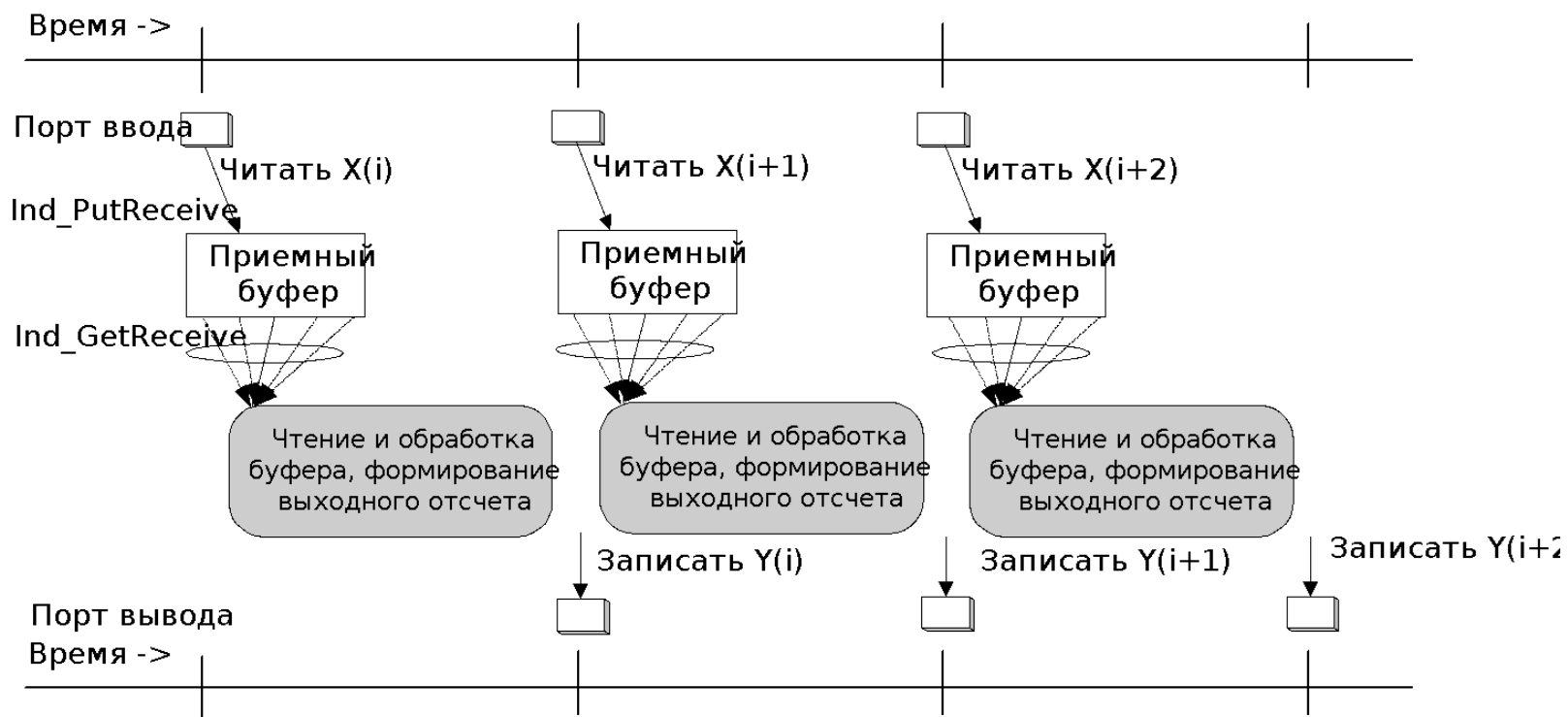
5.2. Способы организации ввода-вывода данных

- с последовательным опросом портов ввода/вывода
- по прерываниям от портов ввода/вывода
- с использованием DMA-контроллера

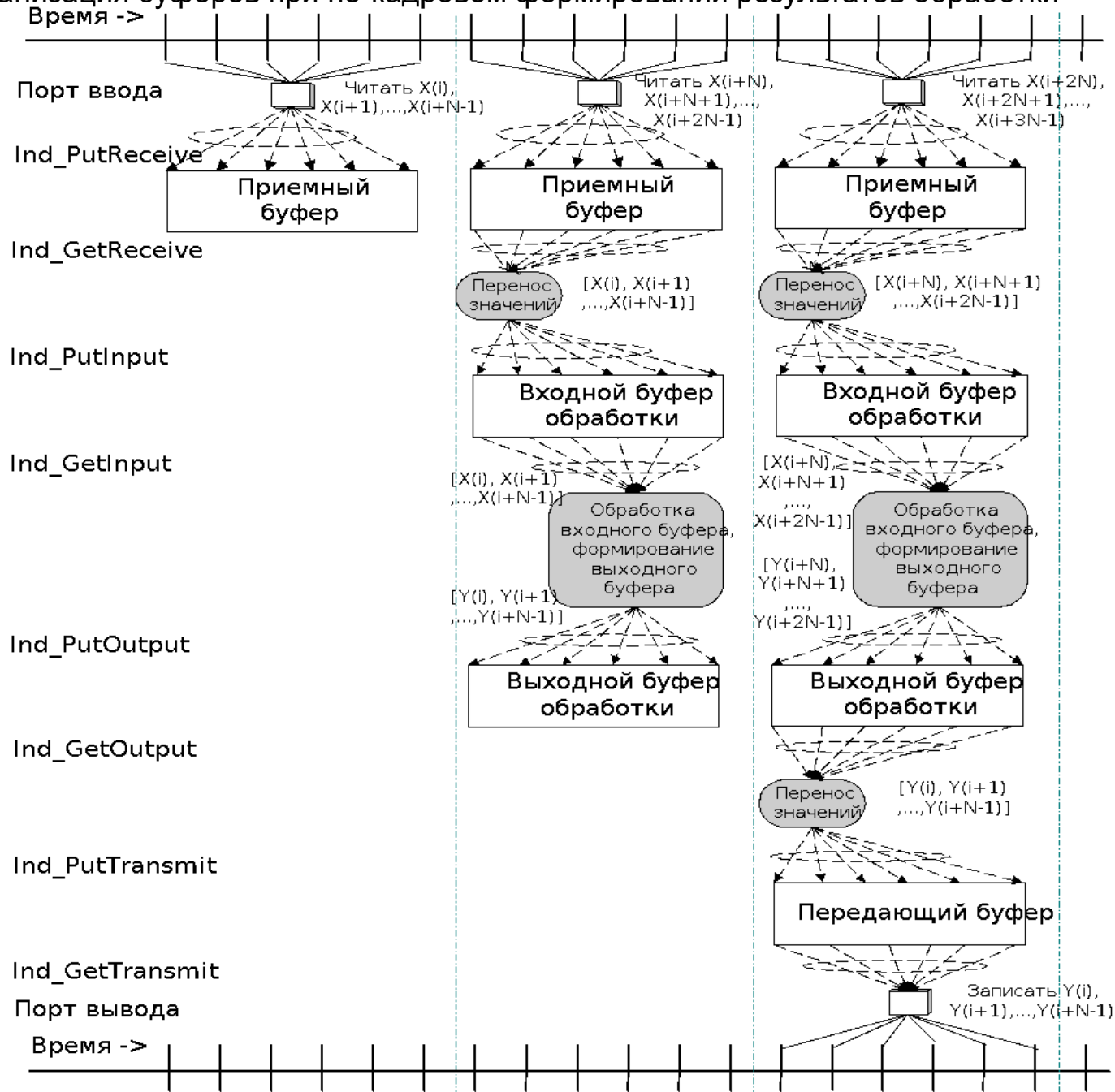


5.3. Организация буферов и доступа к данным при обработке сигналов

Организация буферов при поэлементном формировании результатов обработки



Организация буферов при по-кадровом формировании результатов обработки



Организация буферов при по-кадровом формировании результатов обработки (вариант 2)

