

Новгородский государственный университет имени Ярослава Мудрого  
Научно-исследовательская лаборатория цифровой обработки сигналов

# АНАЛИЗ И СИНТЕЗ ЦИФРОВЫХ УСТРОЙСТВ

---

ИЛЛЮСТРАТИВНЫЙ МАТЕРИАЛ К КОНСПЕКТУ ЛЕКЦИЙ

Реганов Владислав Михайлович

ауд. 4301

Великий Новгород 2014

?????

1. А.Б.Сергиенко. Цифровая обработка сигналов. – СПб.: Питер, 2003 г.
2. Лайонс Р. Цифровая обработка сигналов. пер. с англ. под ред. Бритова А.А. – М.: Бином, 2006.
3. Айфичер Э.С, Джервис Б.У. Цифровая обработка сигналов: практический подход, 2-е издание .Пер. с англ. – М.: Издательский дом «Вильямс», 2004
4. Рабинер Л., Гоулд Б.. Теория и применение цифровой обработки сигналов. /пер. с англ. – Мир, 1978
5. Блейхут Р. Быстрые алгоритмы цифровой обработки сигналов, М., Мир, 1989
6. С.Л. Марпл-мл. Цифровой спектральный анализ и его приложения

# Темы

---

1. Спектральный анализ
  - БПФ
  - Гребенка фильтров
  - Непараметрические методы оценивания СПМ
  - Полифазное БПФ
  - Другие методы
2. Формирование цифровых случайных сигналов
3. Построение цифровых приемных устройств
  - Цифровой гетеродин
  - Преобразователь Гильберта
  - СИС фильтры
  - Полуполосные (диадные) фильтры
  - Полифазные фильтры
4. Цифровое формирование сигнала на несущей частоте
  - Прямой цифровой синтез
5. Основы вейвлет-преобразования
6. Метод CORDIC

# Методы цифрового спектрального анализа

---

## Основные приложения:

- радиолокация, радионавигация, радиоастрономия;
- гидроакустика, гидролокация;
- системы распознавания речи;
- сжатие полосы речевых сигналов;
- вибрационный анализ.

Спектральный анализ – это измерение, которое дает точные или приближенные значения  $Z$  - преобразования дискретного сигнала в заданном множестве точек  $Z$  - плоскости.

Различают “мгновенный” спектр и оценку спектральной плотности мощности.

## Разновидности спектрального анализа:

- вычисление “мгновенного” спектра с использованием окон;
- оценивание СПМ классическими методами;
- оценивание СПМ параметрическими методами;
- оценивание блочных данных;
- рекурсивное оценивание;
- многомерный спектральный анализ.

# Алгоритмы БПФ

Быстрое преобразование Фурье (БПФ) – метод вычисления ДПФ

$\{x(n)\}$ ,  $0 \leq n \leq N-1$  – комплексный сигнал.

$$\text{ДПФ: } X(k) = \sum_{n=0}^{N-1} x(n) W_N^{-jk} =$$

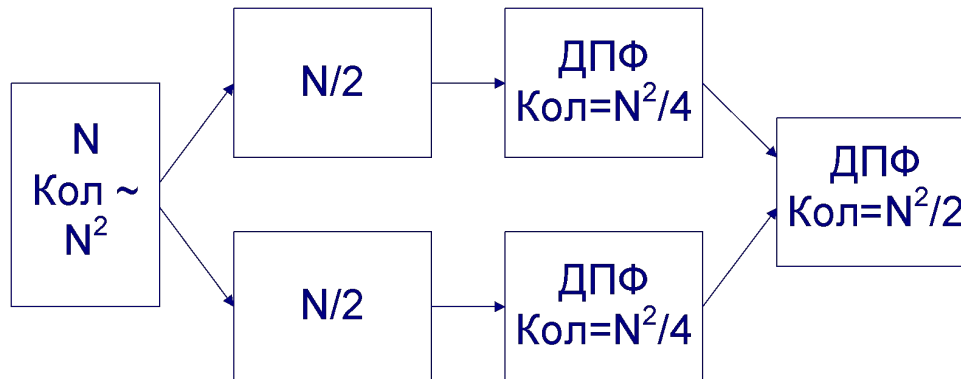
$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot W^{nk} \quad \text{где } W = e^{-j\frac{2\pi}{N}} \text{ - множитель вращения}$$

$\{W^{nk}\}$  периодична по  $n$  и  $k$  с периодом  $N$ :  
 $W^{(n+mN)(k+lN)} = (W_N)^{nk}$ , где  $m, l = 0, \pm 1, \pm 2, \dots$ ,  
 $W_N$  – множитель вращения с периодом  $N$ .

Количество операций для ДПФ размерности  $N$ :

$(N-1)^2$  – комплексных умножений,  $N(N-1)$  – комплексных сложений.

Основная идея БПФ – разбиение длинной последовательности на короткие.



# Алгоритм БПФ с прореживанием по времени

Пусть  $N$  – степень 2.

Разобьем  $\{x(n)\}$  на  $\{x1(n)\}$  – четные отсчеты,  $\{x2(n)\}$  – нечетные отсчеты.

$$x1(n) = x(2n), x2(n) = x(2n+1), \text{ для } n = 0, 1, \dots, \frac{N}{2} - 1$$

$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot W_N^{nk} = \sum_{n=0}^{\frac{N}{2}-1} x(2n) \cdot W_N^{2nk} + \sum_{n=0}^{\frac{N}{2}-1} x(2n+1) \cdot W_N^{(2n+1)k}$$

Поскольку  $W_N^2 = [e^{-j\frac{2\pi}{N}}]^2 = e^{-j\frac{2\pi}{N/2}} = W_{N/2}$  то  $X(k) = \sum_{n=0}^{\frac{N}{2}-1} x_1(n) \cdot W_{N/2}^{nk} + W_N^k \cdot \sum_{n=0}^{\frac{N}{2}-1} x_2(n) \cdot W_{N/2}^{nk}$

$$X(k) = X_1(k) + W_N^k \cdot X_2(k)$$

Тогда  $X(k) = X_1(k) + W_N^k \cdot X_2(k)$

Вычисление  $X_1(k)$  и  $X_2(k)$  –  $2N^2/4$  МАС + объединение  $X_1(k)$  и  $X_2(k)$  –  $N$  МАС

Всего  $N^2/2 + N \rightarrow N^2/2$  при больших  $N$

# Алгоритм БПФ с прореживанием по времени

Доопределение  $X(k)$  для  $k \geq N/2$  на основании периодичности  $N/2$  точечных ДПФ:

$$X(k) = \begin{cases} X_1(k) + W_N^k \cdot X_2(k), & 0 \leq k \leq \frac{N}{2} - 1 \\ X_1(k - \frac{N}{2}) + W_N^k \cdot X_2(k - \frac{N}{2}), & \frac{N}{2} \leq k \leq N - 1 \end{cases}$$

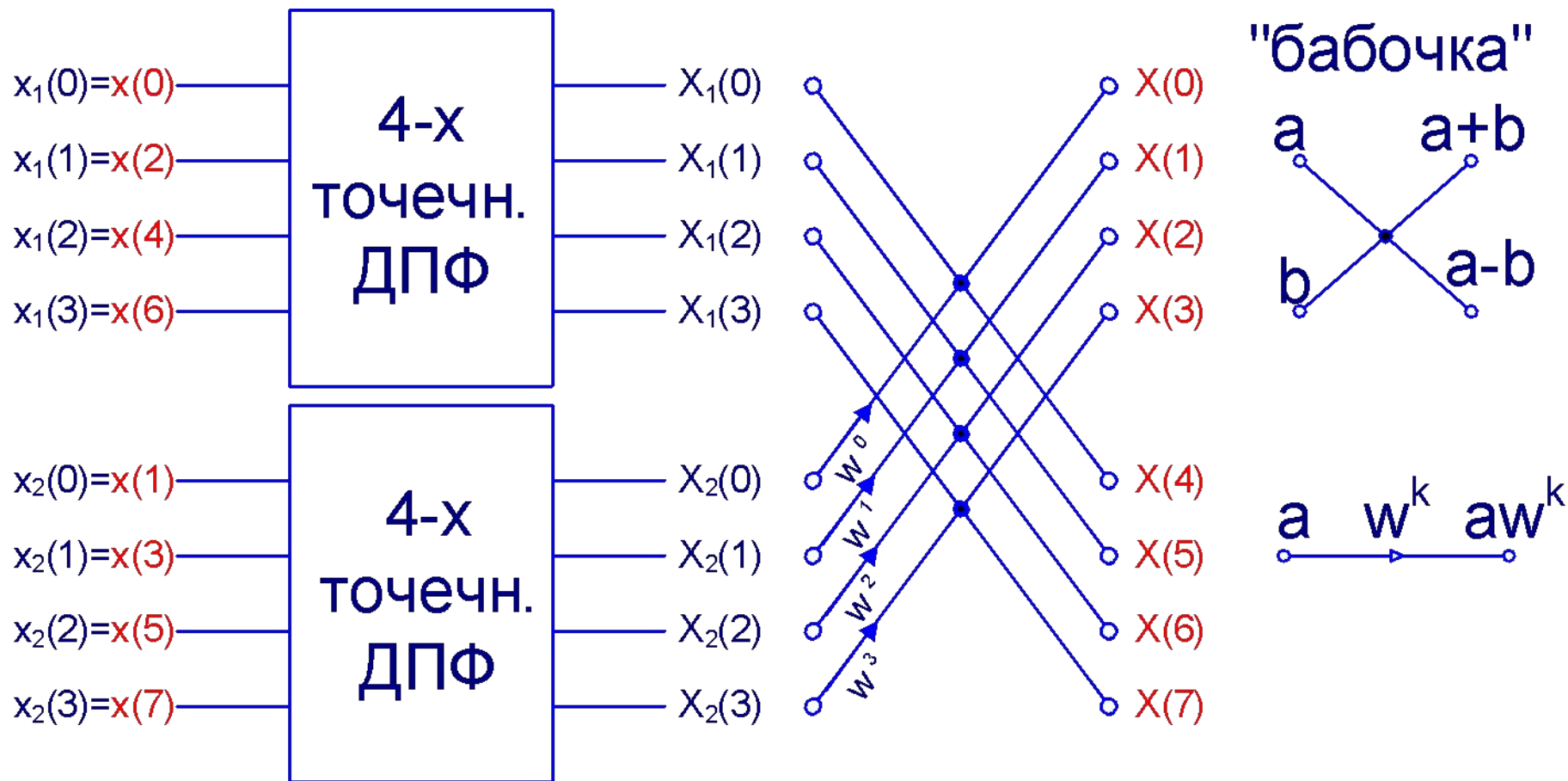
Из-за  $W_N^k$  - период  $X(k)$  не равен периоду  $X_1(k)$ .

Т.к.  $W_N^k = -W_N^{k - \frac{N}{2}}$  то  $X(k) = \begin{cases} X_1(k) + W_N^k \cdot X_2(k), & 0 \leq k \leq \frac{N}{2} - 1 \\ X_1(k - \frac{N}{2}) - W_N^{k - \frac{N}{2}} \cdot X_2(k - \frac{N}{2}), & \frac{N}{2} \leq k \leq N - 1 \end{cases}$

# Пример алгоритма БПФ размерности 8 по основанию 2 с прореживанием по времени

Разложение ДПФ размерности 8 на два ДПФ размерности 4.

## Этап 3



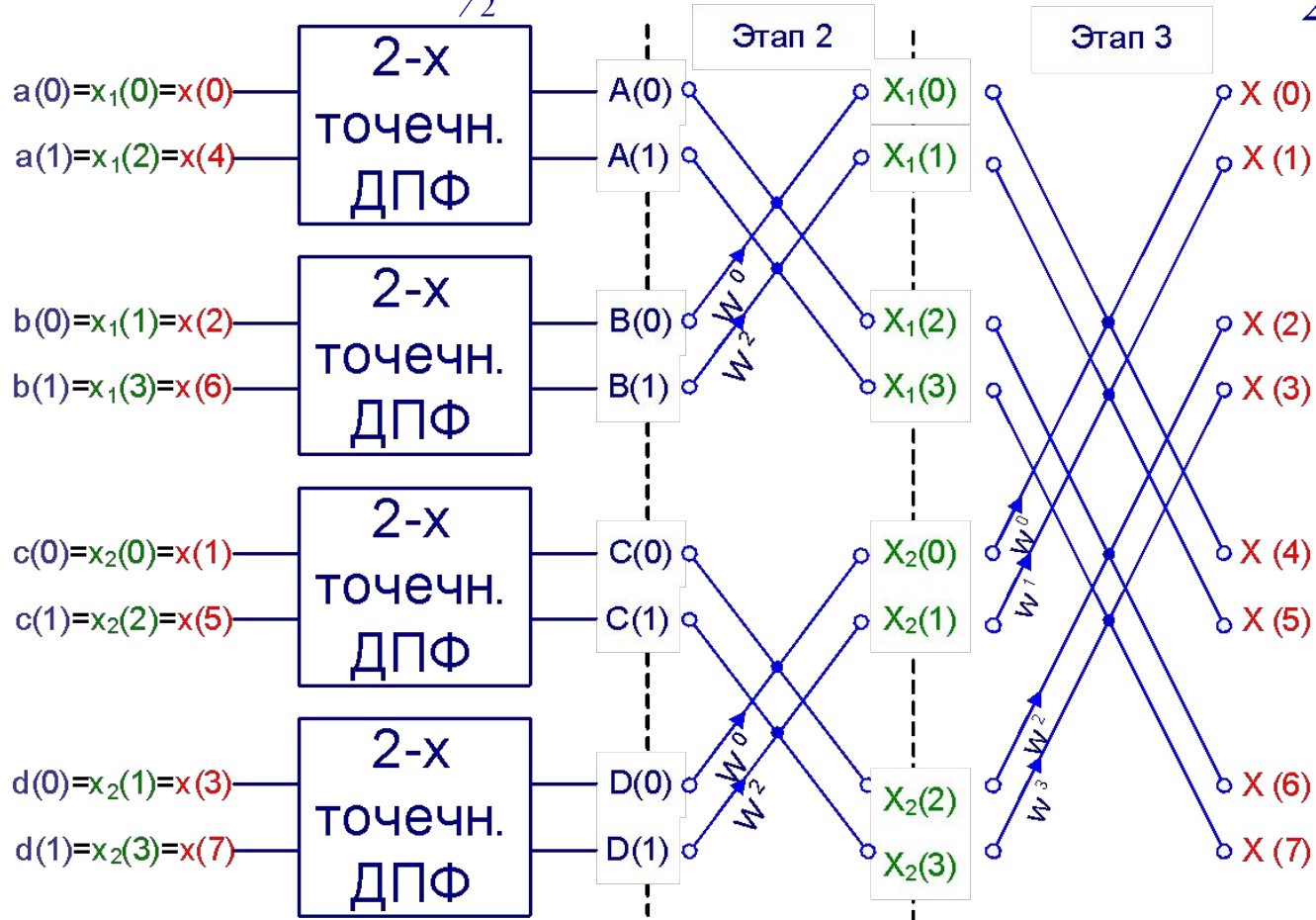


# Пример алгоритма БПФ размерности 8 по основанию 2 с прореживанием по времени

## Этап 2

$$X_1(k) = A(k) + W_{N/2}^k \cdot B(k) = A(k) + W_N^{2k} \cdot B(k), \quad 0 \leq k \leq \frac{N}{2} - 1$$

$$X_2(k) = C(k) + W_{N/2}^k \cdot D(k) = C(k) + W_N^{2k} \cdot D(k), \quad 0 \leq k \leq \frac{N}{2} - 1$$

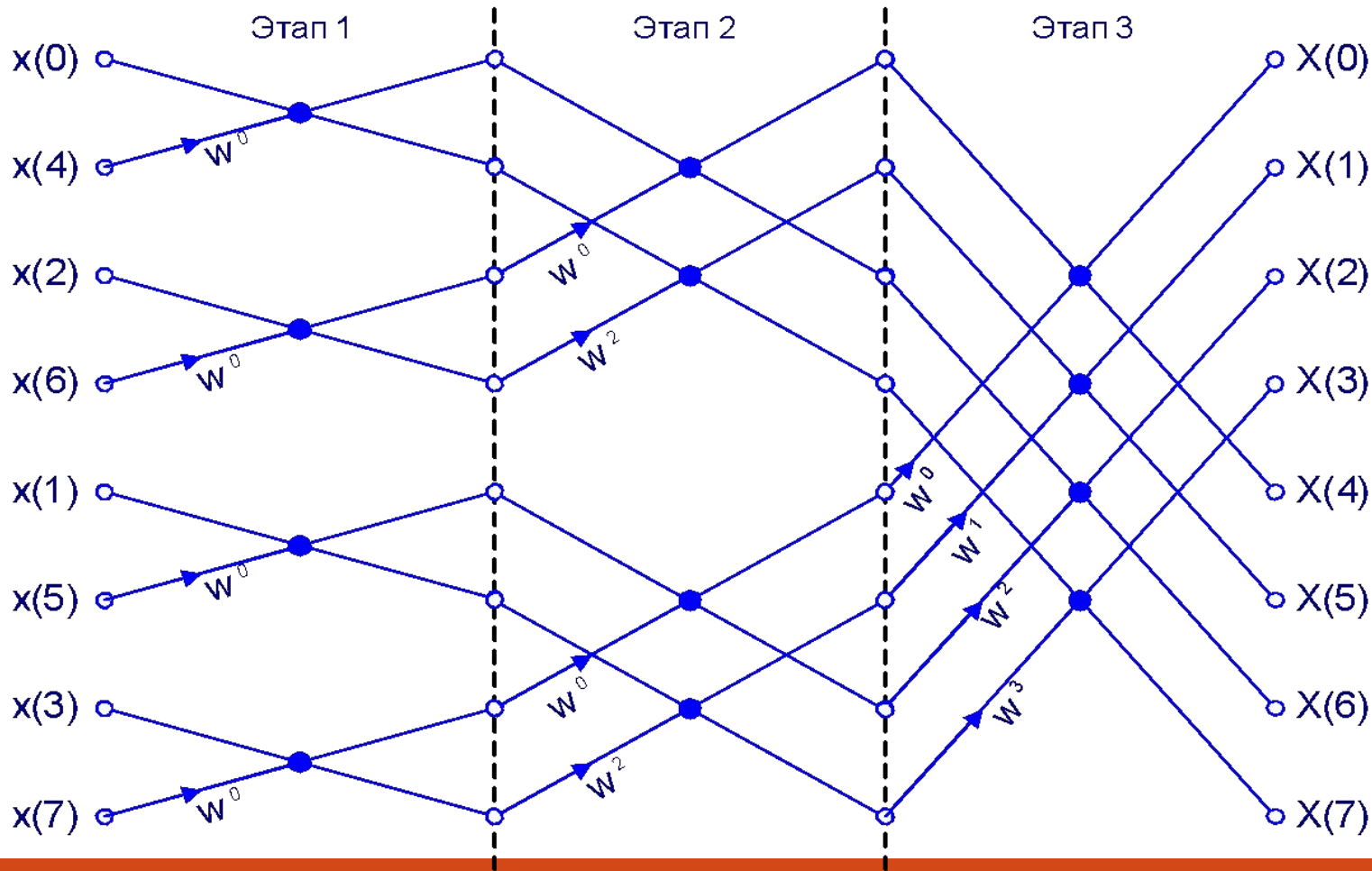


# Направленный граф алгоритма БПФ размерности N = 8 по основанию 2 с прореживанием по времени и с замещением (алгоритм Кули-Тьюки).

Этап 1

$$F(0) = f(0) + f_1 \cdot W_N^0 = f(0) + f(1);$$

$$F(1) = f(0) + f_1 \cdot W_N^{N/2} = f(0) - f(1);$$

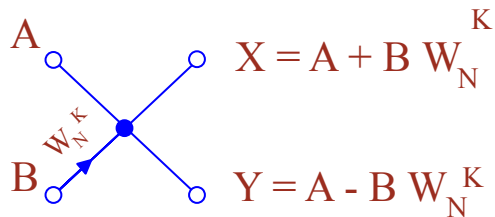


# Свойства алгоритма БПФ по основанию 2

1. Алгоритм состоит из этапов. На каждом этапе происходит изменение размерности БПФ вдвое по сравнению с предыдущим.

$$K_{\text{эт}} = \log_2 N$$

2. На каждом этапе необходимо выполнить  $N/2$  операций "бабочка".



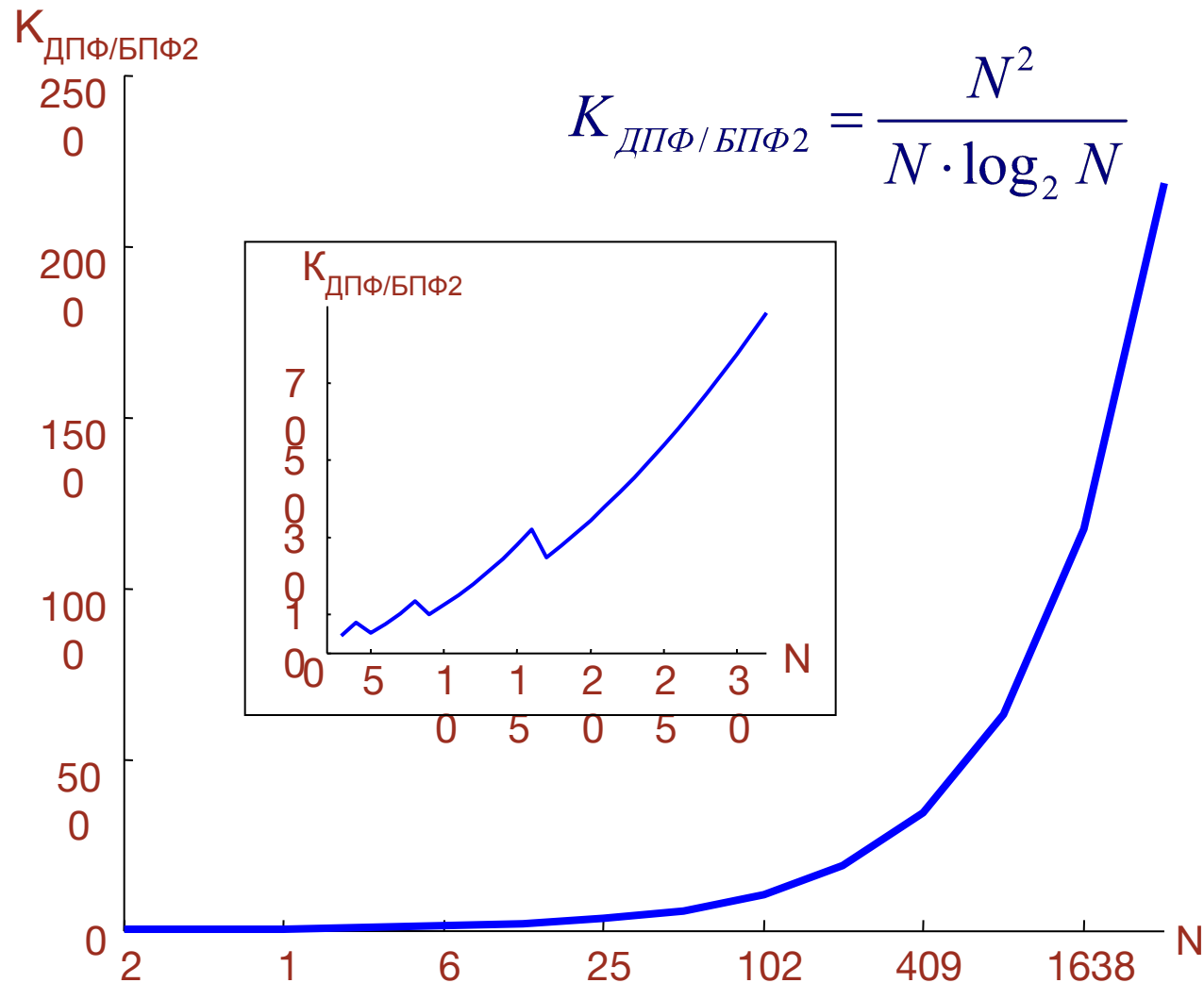
2 операции комплексного сложения и  
1 операция комплексного умножения

3. Общее число базовых операций "бабочка":

$$K_{\text{оп}} = \frac{N}{2} \cdot \log_2 N$$

4. Для вычисления базовой операции достаточно иметь одну дополнительную ячейку для хранения произведения. Остальные результаты размещаются в освободившиеся ячейки. Это алгоритм с замещением.

# Сравнение вычислительных затрат



Выигрыш в количестве операций алгоритма БПФ<sub>2</sub> по сравнению с ДПФ в зависимости от размерности N

# Перестановка данных и двоичная инверсия

Для алгоритма по основанию 2 и прореживанием по времени закон чередования входных отсчетов описывается двоично-инверсным порядком.

Пример:  $N = 8 \Rightarrow L = \log_2 8 = 3$

№ п/п	Двоичное представление	Двоичная инверсия	Двоич.-инверс. номер
0	000	000	0
1	001	100	4
2	010	010	2
3	011	110	6
4	100	001	1
5	101	101	5
6	110	011	3
7	111	111	7

## Способы получения поворачивающих множителей

1. Табличный – требует много памяти, но имеет наибольшее быстродействие
2. Последовательный – не требует много памяти, но имеет низкое быстродейст.
3. Рекуррентный  $W_N^K = W_N^{K-L} \cdot W_N^L$  с изменением шага от этапа к этапу и с начальным условием  $W_N^0 = 1$  на каждом этапе

# Алгоритм БПФ с прореживанием по частоте

Входная последовательность разбивается на 2 половины:

$$x_1(n) = x(n), \quad n = 0, 1, 2, \dots, \frac{N}{2} - 1$$

$$x_2(n) = x\left(n + \frac{N}{2}\right), \quad n = 0, 1, 2, \dots, \frac{N}{2} - 1$$

Тогда N-точечное ДПФ последовательности  $\{x(n)\}$ :

$$X(k) = \sum_{n=0}^{\frac{N}{2}-1} x(n) \cdot W_N^{nk} + \sum_{n=\frac{N}{2}}^{N-1} x(n) \cdot W_N^{nk} = \sum_{n=0}^{\frac{N}{2}-1} x_1(n) \cdot W_N^{nk} + \sum_{n=0}^{\frac{N}{2}-1} x_2(n) \cdot W_N^{(n+\frac{N}{2})k}$$

Т.к.  $W_N^{\frac{N}{2}k} = e^{-j\pi k}$  то  $X(k) = \sum_{n=0}^{\frac{N}{2}-1} [x_1(n) + e^{-j\pi k} \cdot x_2(n)] \cdot W_N^{nk}$

# Алгоритм БПФ с прореживанием по частоте

Поскольку  $W_N^{\frac{N}{2}k} = e^{-j\pi k} = \pm 1$  то  $X(k)$  для четных и нечетных  $k$ :

$$X(2k) = \sum_{n=0}^{\frac{N}{2}-1} [x_1(n) + x_2(n)] \cdot W_N^{2nk} = \sum_{n=0}^{\frac{N}{2}-1} [x_1(n) + x_2(n)] \cdot W_{N/2}^{nk}$$

$f(n)$

$$X(2k+1) = \sum_{n=0}^{\frac{N}{2}-1} [x_1(n) - x_2(n)] \cdot W_N^{n(2k+1)} = \sum_{n=0}^{\frac{N}{2}-1} \{ [x_1(n) - x_2(n)] \cdot W_N^n \} \cdot W_{N/2}^{nk}$$

$g(n)$

$X(2k)$  получаются из  $N/2$ -точечных ДПФ последовательности:

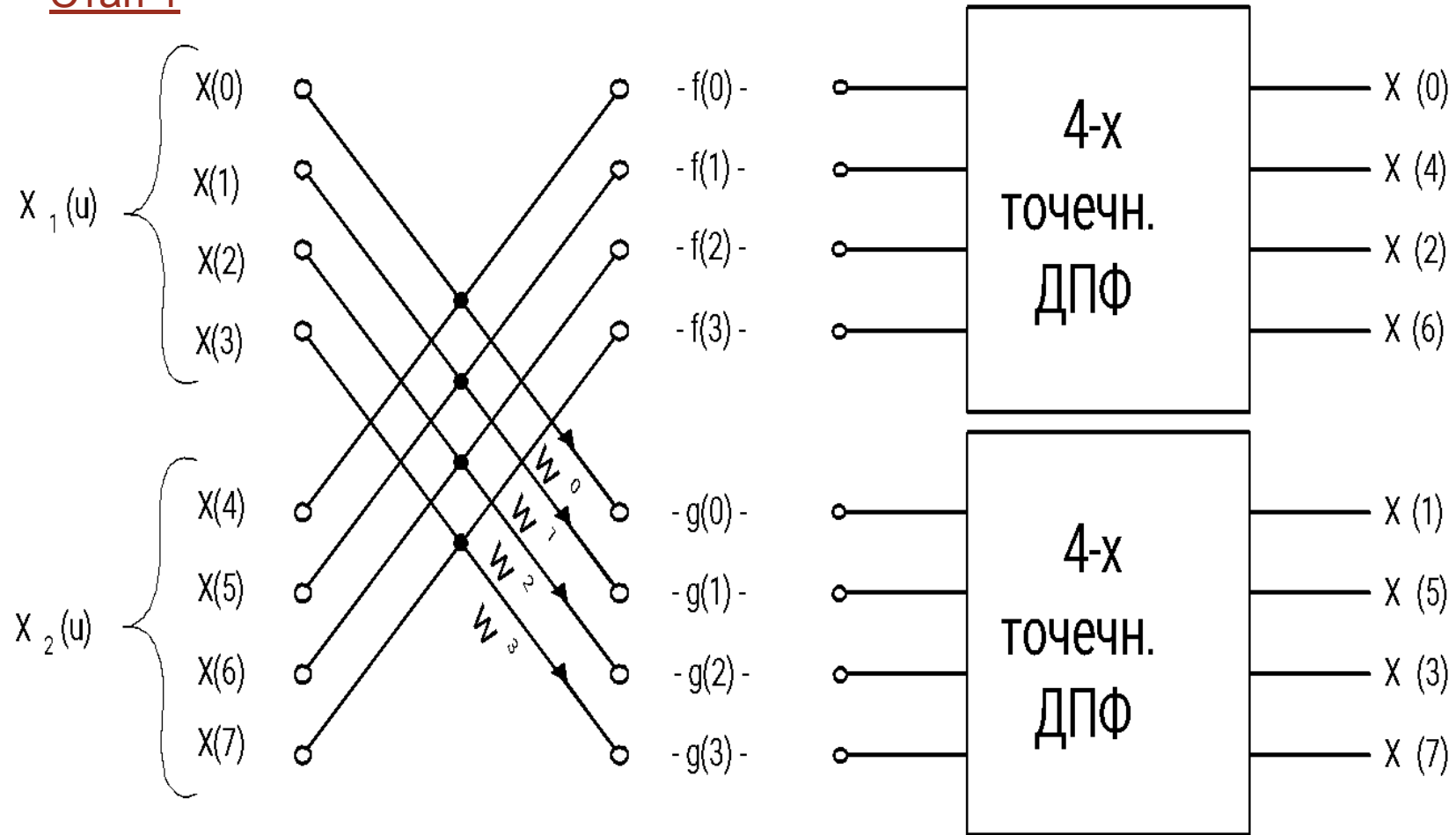
$$f(n) = x_1(n) + x_2(n); \quad n = 0, 1, 2, \dots, N/2 - 1$$

$X(2k+1)$  получаются из  $N/2$ -точечных ДПФ последовательности:

$$g(n) = [x_1(n) - x_2(n)] W_N^n; \quad n = 0, 1, 2, \dots, N/2 - 1$$

# Пример построения алгоритма БПФ размерности 8 с прореживанием по частоте

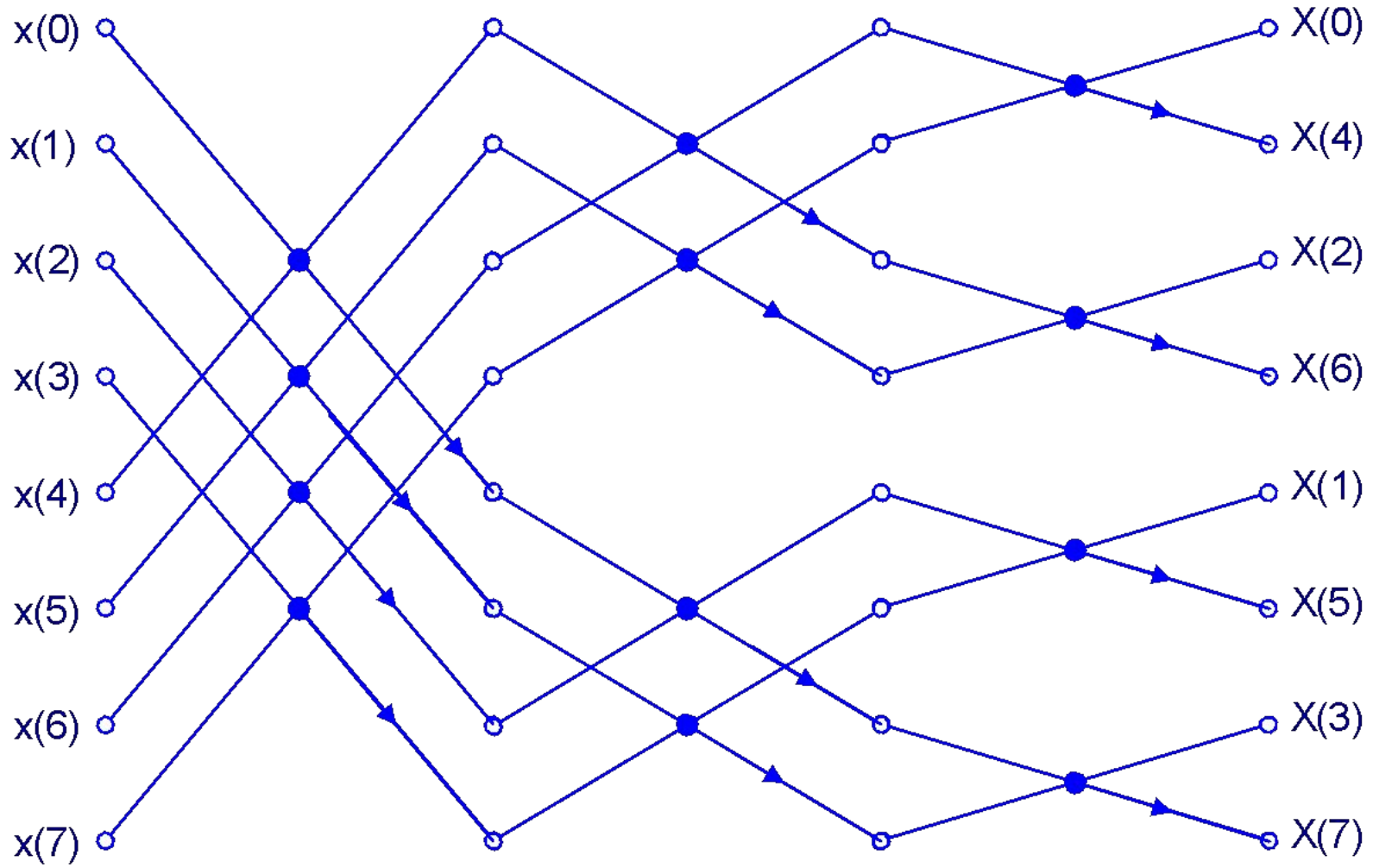
## Этап 1





# Алгоритмы БПФ по основанию 2

Направленный граф алгоритма БПФ по основанию 2 с прореживанием по частоте.



# Различия алгоритмов БПФ с прореживанием по времени и по частоте

## по времени

### 1. Порядок следования входных отсчетов:

двоично-инверсный

## по частоте

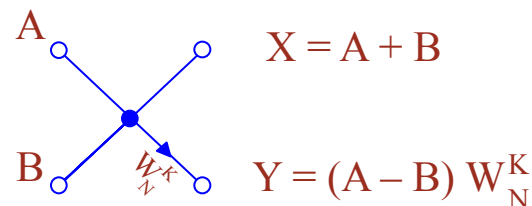
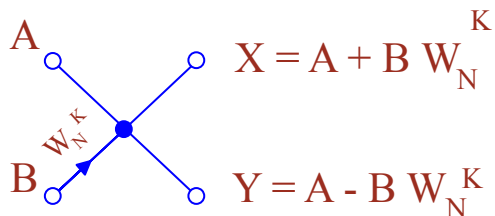
прямой

### 2. Порядок следования выходных отсчетов:

прямой

двоично-инверсный

### 3. Базовая операция “бабочка”



# Вычисление обратного ДПФ по алгоритму прямого

Обратное ДПФ  $\{x(n)\}$  для последовательности  $\{X(k)\}$ ,  $k=0,1,\dots,N-1$

$$x(n) = \frac{1}{N} \cdot \sum_{k=0}^{N-1} X(k) \cdot W^{-nk} \quad \text{- обратное ДПФ}$$

$$N \cdot x^*(n) = \sum_{k=0}^{N-1} X^*(k) \cdot W^{nk} \quad \text{* - знак комплексного сопряжения}$$

ДПФ послед-ти  $X^*(k)$

Тогда:

$$x(n) = \frac{1}{N} \cdot \left[ \sum_{k=0}^{N-1} X^*(k) \cdot W^{nk} \right]^*$$

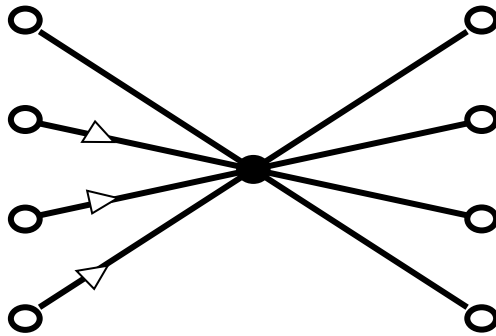
Т.о. можно использовать алгоритмы БПФ для вычисления ДПФ и ОДПФ

# Алгоритмы БПФ по основанию 4

По аналогии с основанием 2 можно построить алгоритмы БПФ по основанию 4.

ДПФ размерности 4 не требует операций комплексного умножения, так

как умножение на  $W = \exp\left(j \frac{2\pi}{4}\right)$  выполняется перестановкой реальной и мнимой компонент



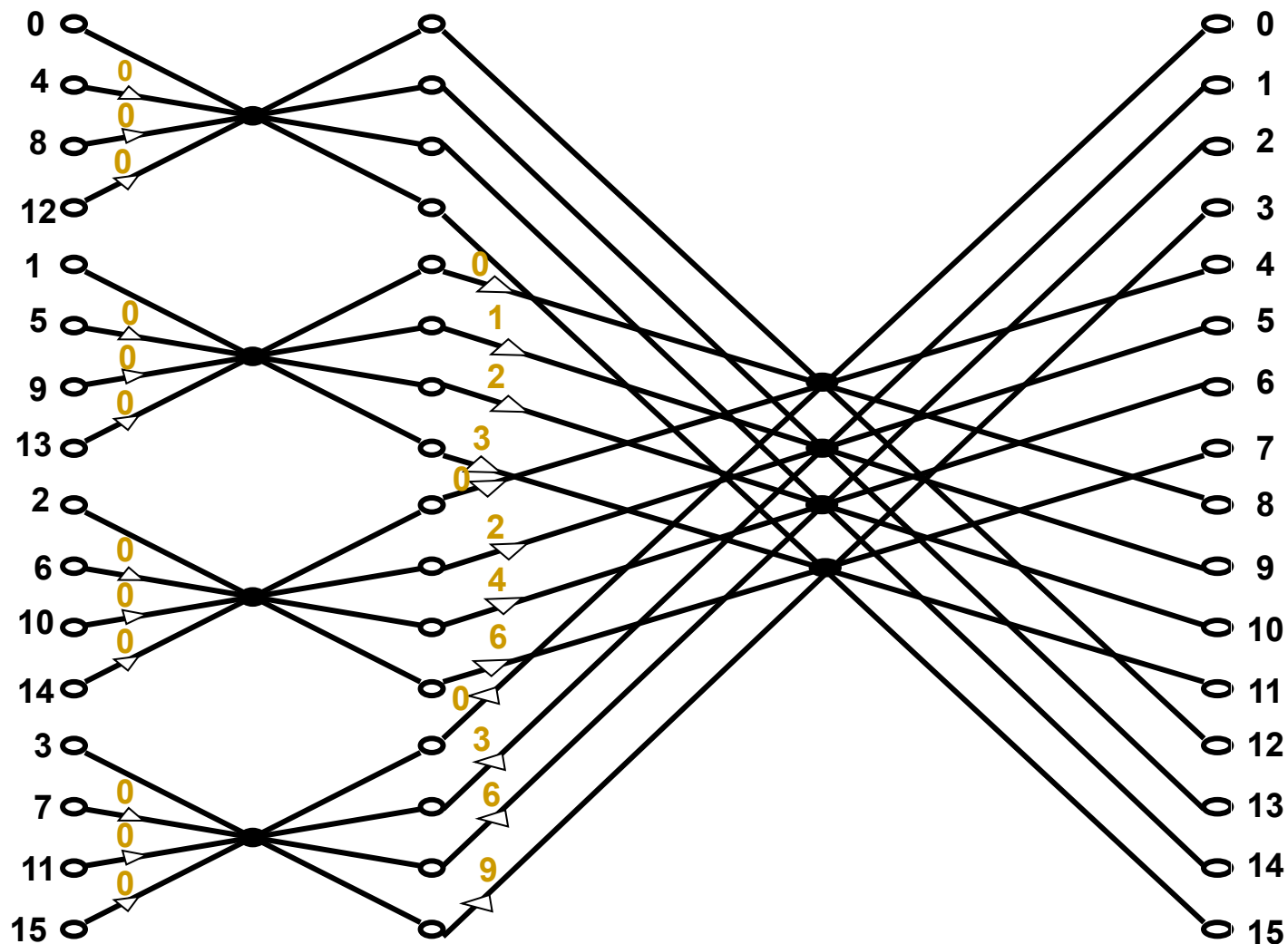
**3 комплексных  
умножения**

**12 комплексных  
сложений**

Операция «бабочка» по основанию 4 с прореживанием по времени

Выигрыш по количеству операций комплексного умножения по сравнению с алгоритмом БПФ по основанию 2 около 25%.

# Алгоритм БПФ по основанию 4 размерности 16



# Принцип построения алгоритма БПФ с произвольным основанием

Если  $N$  – составное число, то одномерный массив отсчетов можно записать в виде матрицы размерности  $N=M \times L$ .

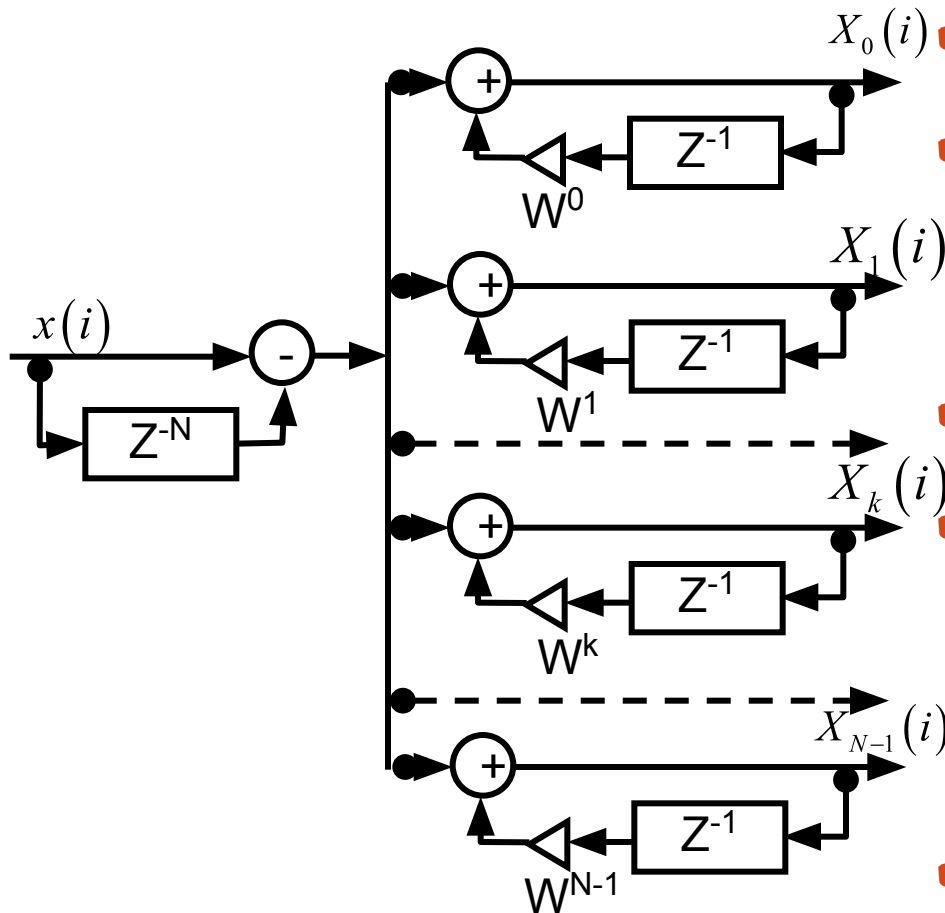
## Алгоритм вычисления ДПФ размерности $N$ :

- ❖ Преобразовать одномерный массив в матрицу (заполнение по строкам!)
- ❖ Вычислить ДПФ каждого столбца  
$$X_c(k, l) = \sum_{m=0}^{M-1} x(m, l) \cdot W_M^{mk}, \quad k = 0..M-1, \quad l = 0..L-1$$
- ❖ Умножить элементы матрицы  
$$\tilde{X}_c(k, l) = X_c(k, l) \cdot W_N^{kl}, \quad k = 0..M-1, \quad l = 0..L-1$$
- ❖ Вычислить ДПФ каждой строки  
$$X_r(k, i) = \sum_{l=0}^{L-1} \tilde{X}_c(k, l) \cdot W_L^{li}, \quad k = 0..M-1, \quad i = 0..L-1$$
- ❖ Преобразовать матрицу в одномерный массив (считывание по строкам!).

*Если размерность строки или столбца - составное число, разбиение можно повторить.*

Для произвольных составных  $N$  наиболее быстрый алгоритм со смешанным основанием – АВПФ (алгоритм Винограда преобразования Фурье).

# Сравнение БПФ и гребенки фильтров.



## Гребенка фильтров:

- ✓ выдает N спектральных отсчетов в каждый момент времени;
- ✓ Требуется N операций умножения-накопления на 1 отсчет сигнала.

## БПФ без перекрытия:

- ✓ Выдает N спектральных отсчетов через N отсчетов сигнала;
- ✓ Требуется  $\frac{1}{2} \cdot \log_2 N$  операций умножения-накопления на 1 отсчет сигнала.

## БПФ с перекрытием:

- ✓ Выдает N отсчетов через  $\frac{N}{K}$  отсчетов сигнала;
- ✓ Требуется в K раз больше операций

Анализатор спектра в виде гребенки фильтров

# Использование «окон» при спектральном анализе

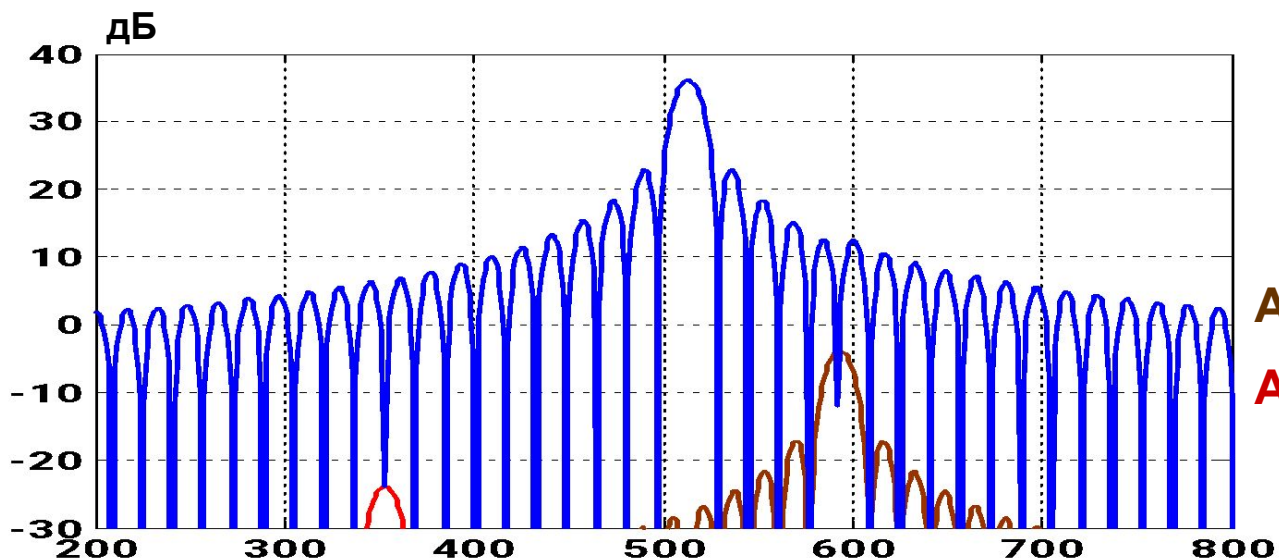
Импульсная характеристика  
одного из гребенки фильтров:

$$h(n) = \begin{cases} \exp\left(j \cdot \frac{2\pi k}{N} \cdot n\right), & 0 \leq n \leq N-1 \\ 0, & \text{остальные } n \end{cases}$$

Частотная характеристика  
(без фазового множителя):

$$H(e^{j\omega}) \approx \frac{\sin\left(\frac{N\omega}{2}\right)}{\sin\left(\frac{\omega}{2} - \frac{\pi k}{N}\right)} \quad k=0, 1..N-1$$

Проблема: маскировка слабых спектральных компонент сильными из-за высоких боковых лепестков АЧХ фильтра.



Амплитуды  
сигналов:

**A1 – 1 (0 дБ)**

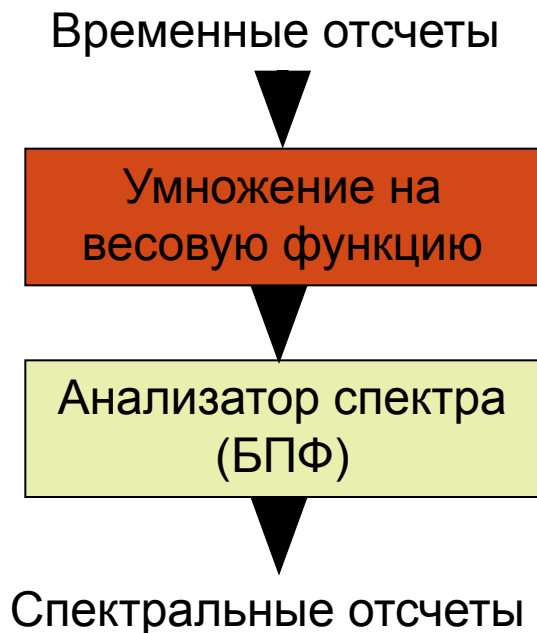
**A2 – 0.01 (-40 дБ)**

**A3 – 0.001 (-60 дБ)**



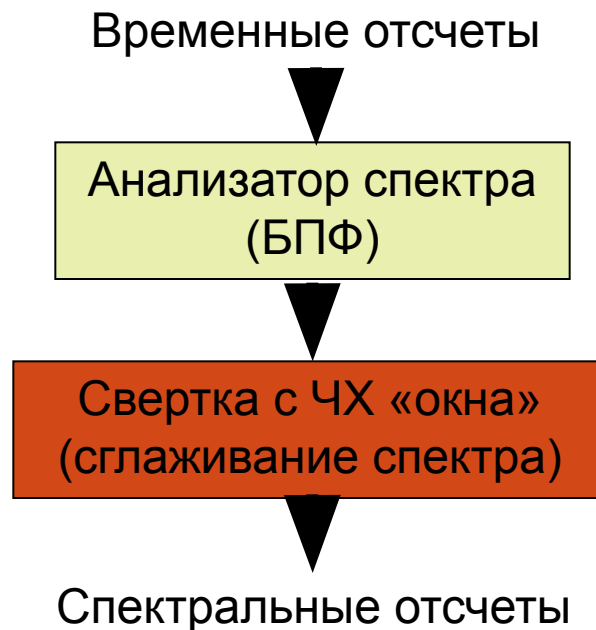
# Использование «окон» при спектральном анализе

Во временной области – умножение сигнала на весовую функцию «окна».



1 умножение на отсчет для всех видов окон

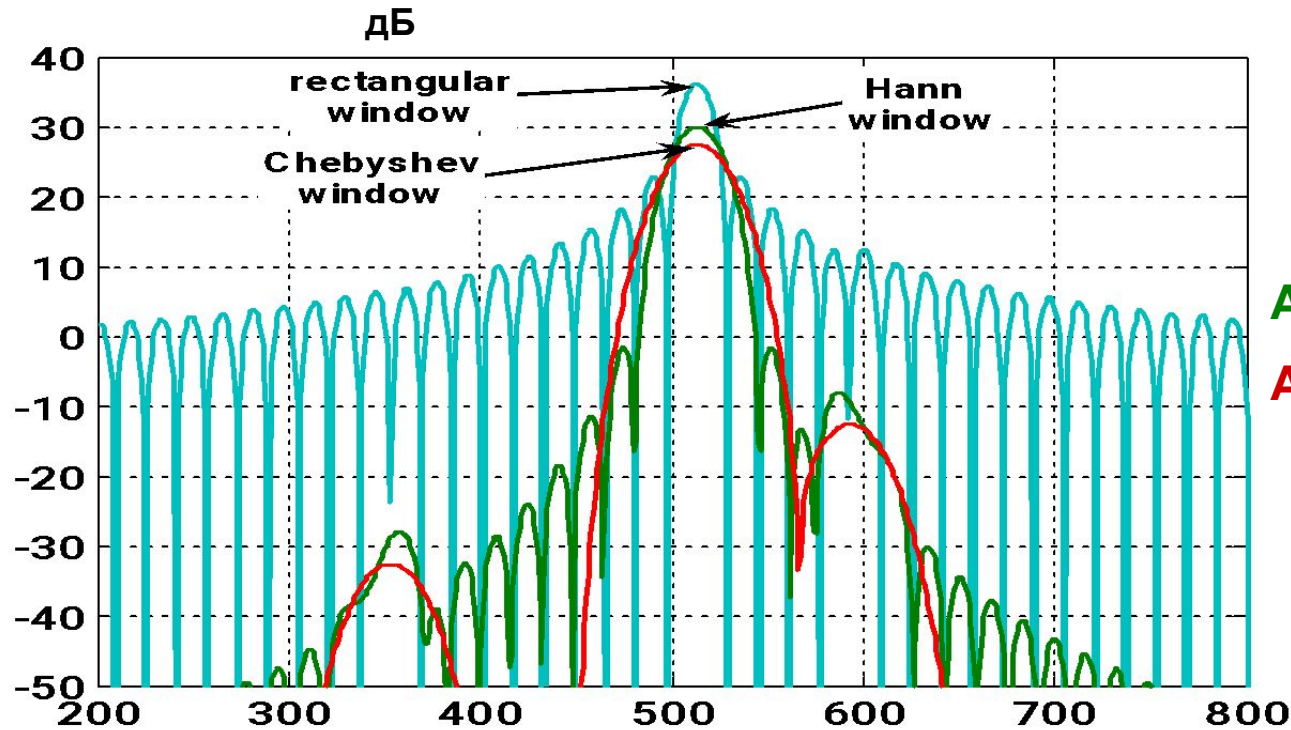
В спектральной области – свертка спектра сигнала с частотной характеристикой «окна».



Для окна Ханна порядок фильтра -3  
(окно Хэмминга – без умножений).

Для окна Блэкмана порядок фильтра - 5.

# Использование «окон» при спектральном анализе



Стратегия выбора «окна» по одному из параметров:

- по скорости спада БЛ – при большой разнице амплитуд и частот;
- по максимальному уровню БЛ – при разных амплитудах и неизвестных (распределенных в большом диапазоне) частотах;
- по ширине основного лепестка АЧХ – при сопоставимых амплитудах и близко расположенных частотах.

# Классические методы спектрального оценивания

**Задача:** получить оценку спектральной плотности мощности сигнала с минимальной среднеквадратической ошибкой по зашумленной реализации конечной длительности.

**Основные характеристики:**

## ✓ Диапазон анализируемых частот

Определяется частотой дискретизации  $F_s$ :

- от 0 до  $\frac{1}{2} F_s$  для действительных сигналов;
- от  $-\frac{1}{2} F_s$  до  $+\frac{1}{2} F_s$  для комплексных сигналов.

## ✓ Разрешающая способность по частоте

Определяется эффективной шириной главного лепестка ЧХ окна  $B_e$ :

$$B_e = \frac{\sum_{k=0}^{N-1} |W(k)|}{\max_{k \in [0, N-1]} (|W(k)|)}$$

## ✓ Достоверность

Определяется относительной среднеквадратической ошибкой  $Q$  оценки СПМ

$$Q(k) = \frac{\langle |S(k) - \bar{S}(k)|^2 \rangle}{\langle |S(k)|^2 \rangle}, \quad k = 0, 1, \dots, N-1$$

# Классические методы спектрального оценивания

*Особенность оценки СПМ при наличии шума:*

При увеличении размерности БПФ ошибка оценки СПМ не уменьшается, так как определяется спектральной плотностью шума.

Для ее снижения необходимо усреднение спектральных оценок.

*При ограниченной длине реализации случайного процесса:*

- ✓ Повышение достоверности оценки приводит к ухудшению разрешающей способности;
- ✓ Повышение разрешающей способности приводит к потере достоверности оценки.

$$B_e \cdot T_e \approx 1$$

*Если влияние шума пренебрежимо мало, то*

$T_e$  - эффективная длительность реализации.

*Если необходимо усреднение оценок СПМ для повышения достоверности*

$$B_S = \frac{\left[ \sum_{k=0}^{N-1} W(k) \right]}{\sum_{k=0}^{N-1} [W(k)]^2}$$

- статистическая ширина

$$Q \cdot T_e \cdot B_S \approx 1$$

полосы

«ОКНА»

# Периодограммный метод оценки СПМ

*Последовательность операций:*

1. Реализация процесса длиной  $L$  отсчетов разбивается на  $M$  сегментов размером  $N$  отсчетов каждый

$$x_m(n), \quad n = 0, 1, \dots, N-1, \quad m = 1, 2, \dots, M.$$

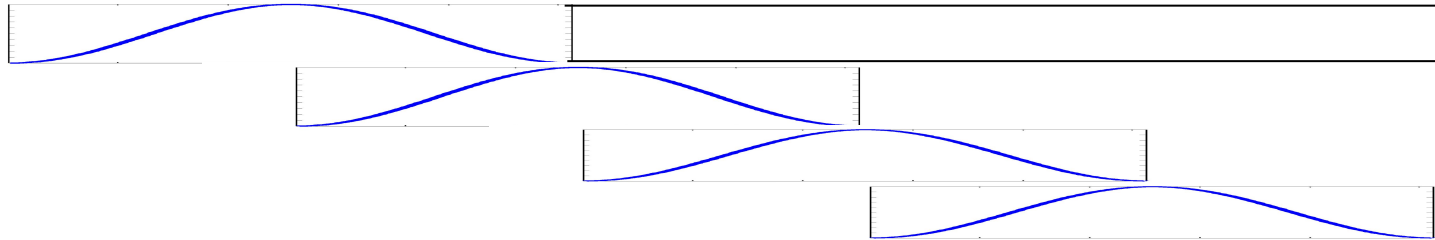
2. Вычисляется БПФ от каждого сегмента

$$X_m(k), \quad k = 0, 1, \dots, N-1, \quad m = 1, 2, \dots, M.$$

3. Усредняется оценка СПМ

$$P_X(k) = \frac{1}{M} \cdot \sum_{m=1}^M X_m(k) \cdot X_m^*(k), \quad k = 0, 1, \dots, N-1$$

Для снижения потерь из-за взвешивания функцией «окна» применяется перекрытие сегментов



*Увеличение длины сегмента соответствует улучшению разрешающей способности и снижению достоверности (возрастанию ошибки), и наоборот.*

# Коррелограммный метод оценки СПМ

Основан на дискретном аналоге теоремы Винера-Хинчина.

*Последовательность операций:*

1. Вычислить АКФ реализации процесса в диапазоне  $[0, N-1]$  дискретных задержек:

$$R_X(m) = \sum_{n=0}^{L-1} x(n) \cdot x^*(n-m), \quad m = 0, 1..N-1.$$

2. Вычислить ДПФ размерности  $N$  от АКФ с использованием «окна»:

$$P_X(k) = \sum_{m=0}^{N-1} R_X(m) \cdot \exp\left(-j \frac{2\pi}{N} \cdot k \cdot m\right), \quad k = 0, 1..N-1.$$

*Увеличение диапазона задержек АКФ соответствует улучшению разрешающей способности, и снижению достоверности*

*(возрастанию ошибки), и наоборот.*

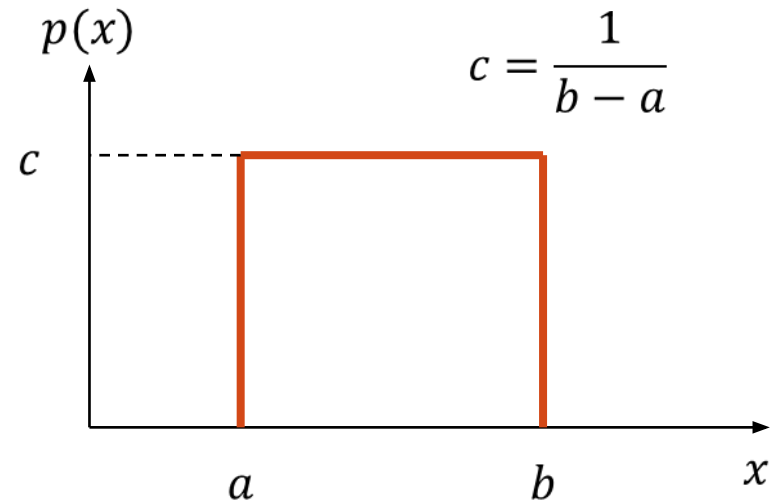
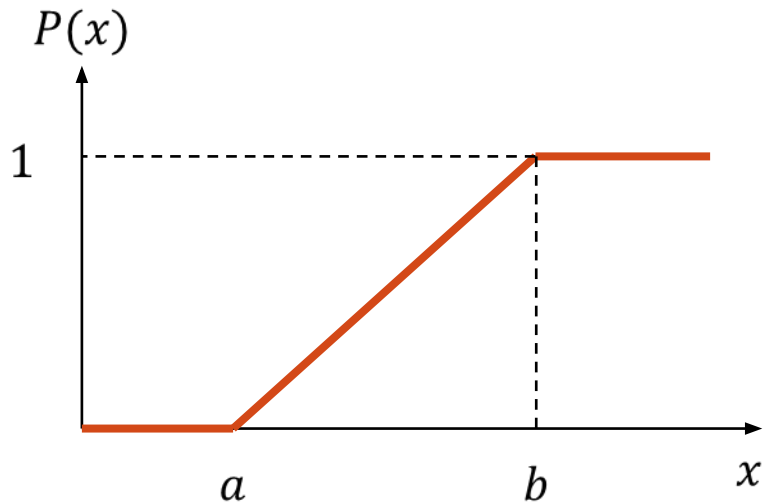
**Рекомендуется:** начинать оценку СПМ с высокой достоверности, продвигаясь в направлении более высокого разрешения по частоте.

# Формирование случайных сигналов (шумов)

## Формирование случайного сигнала с равномерным распределением

Пусть  $X$  – случайный процесс, а  $x$  – его любое допустимое значение, тогда функция распределения вероятностей  $P(x)$  определяет вероятность того, что  $X \leq x$

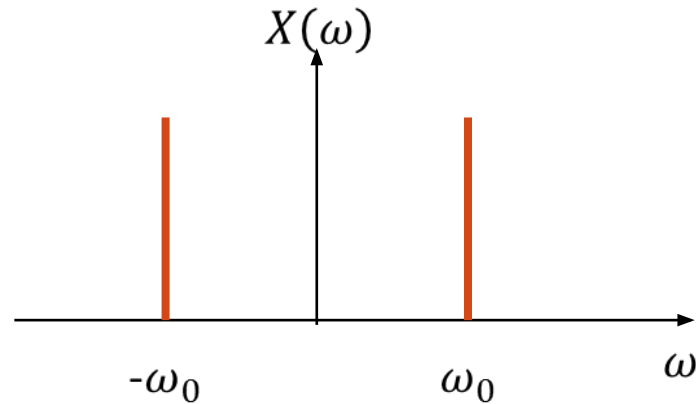
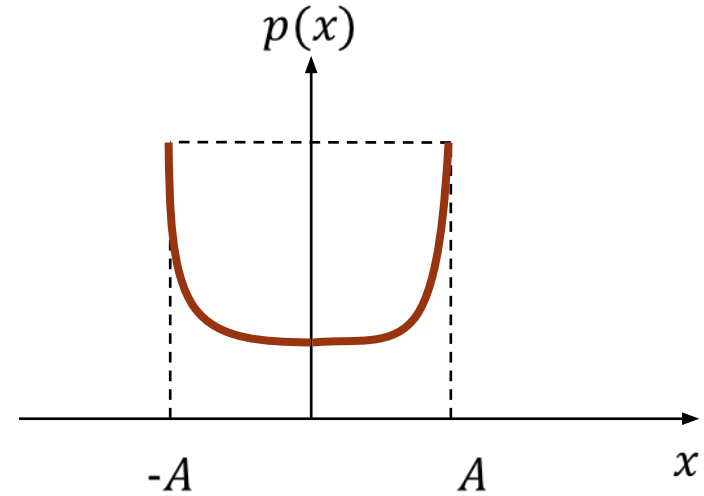
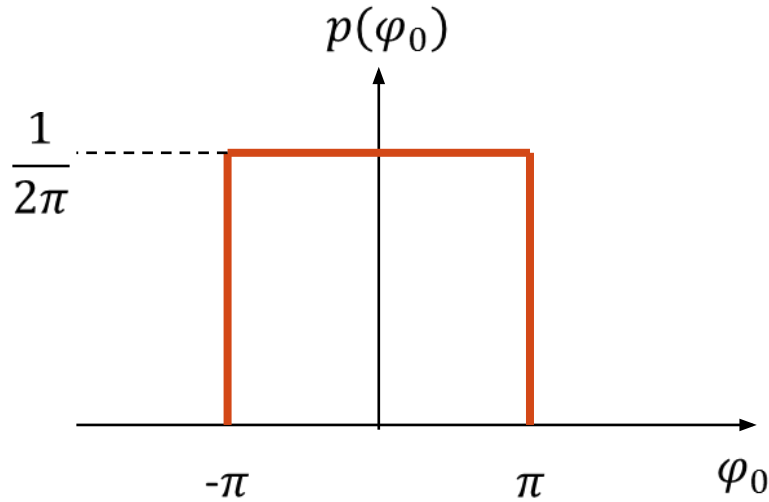
$p(x) = \frac{dP(x)}{d(x)}$  – плотность вероятности случайного процесса  $X$



# Формирование случайного сигнала с равномерным распределением

Гармонический сигнал со случайной начальной фазой

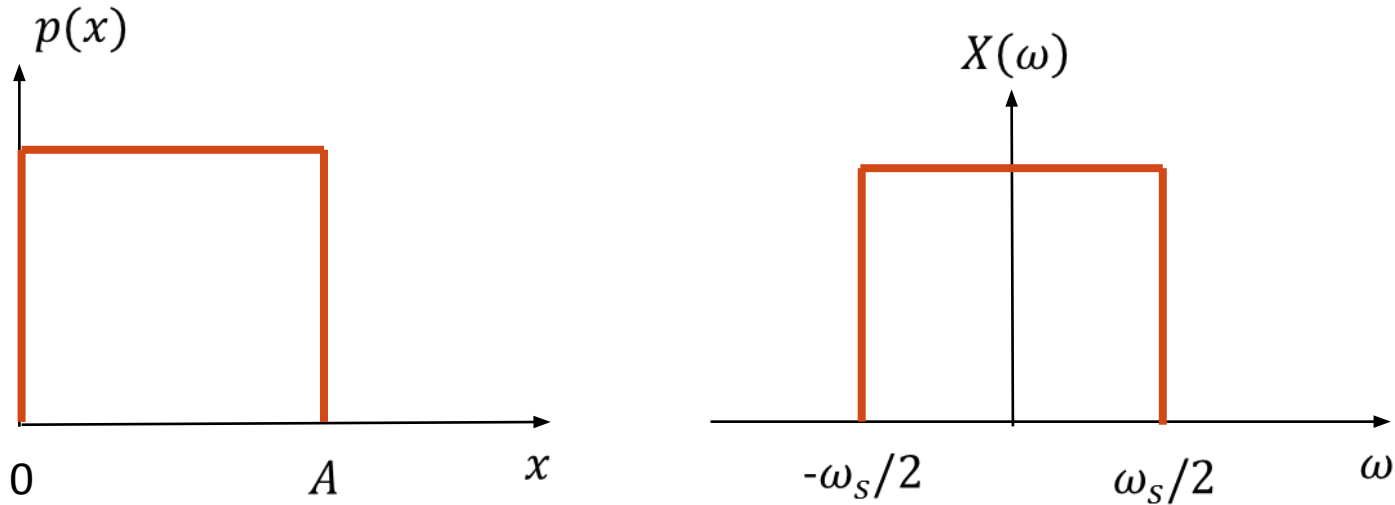
$$x(n) = A \sin(\omega_0 n + \varphi_0)$$





# Формирование случайного сигнала с равномерным распределением

Случайный дискретный сигнал с равномерным распределением амплитуд



**Шум, но не белый!!!,  
т.к. распределение амплитуд не по нормальному (гауссовому) закону**

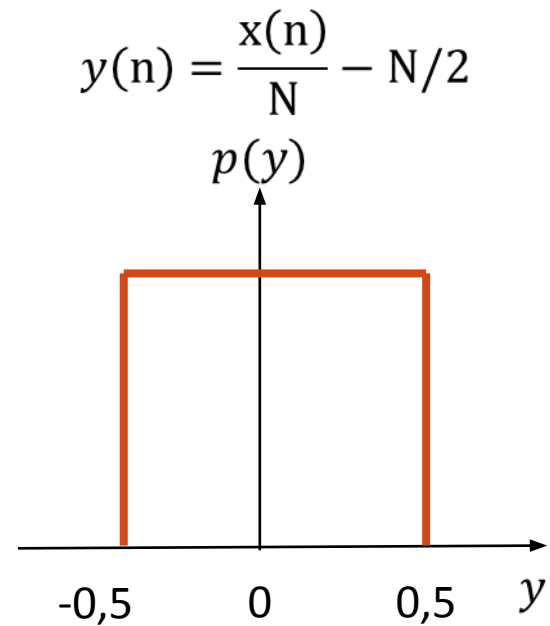
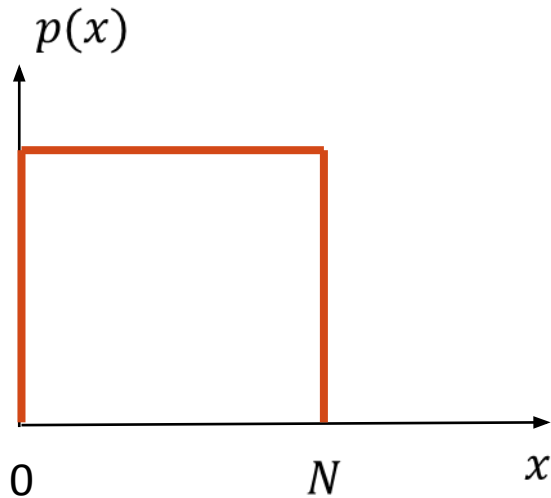
# Формирование случайного сигнала с равномерным распределением

$$x(n) = x(n - 1) \cdot K \cdot \text{mod}(N), \quad n = 1, 2, \dots; \quad x(0) \neq 0$$

$K$  – специально подобранная константа;

$K$  – чаще всего большое простое число.

Удобно задавать  $N = 2^M - 1$



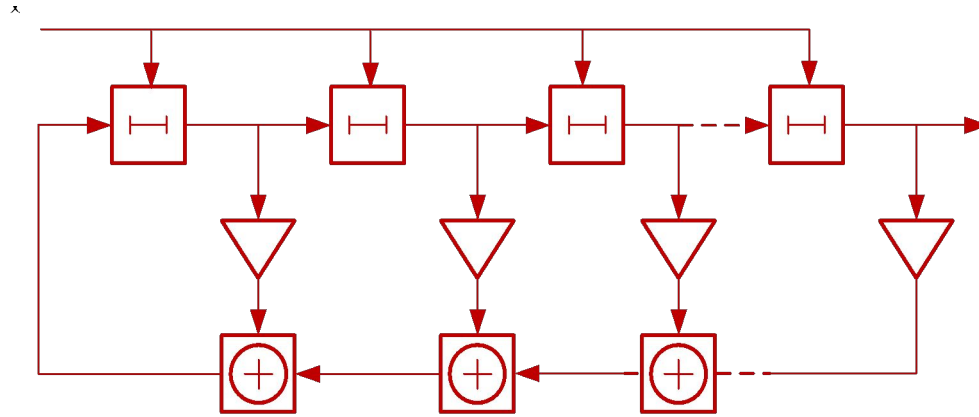
# Формирование случайного сигнала с равномерным распределением

**M-последовательность** или последовательность максимальной длины (Maximum length sequence, MLS) — псевдослучайная двоичная последовательность, порожденная регистром сдвига с линейной обратной связью и имеющая максимальный период.

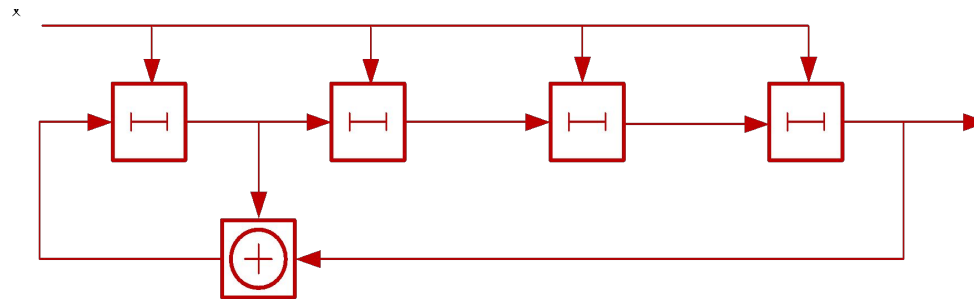
- период последовательности  $N = 2^M - 1$ ;
- количество “1”, на длине N на единицу больше, чем количества “0”;
- любая комбинация из M символов на длине N имеет не более одного повторения (кроме нулей);
- сумма по модулю 2 любой M-последовательности с её произвольным циклическим сдвигом также является M-последовательностью;
- УБЛ периодической АКФ равен  $-\frac{1}{N}$
- УБЛ АКФ близок к  $\frac{1}{\sqrt{N}}$

# Формирование случайного сигнала с равномерным распределением

## Обобщенная схема генератора двоичной M-последовательности

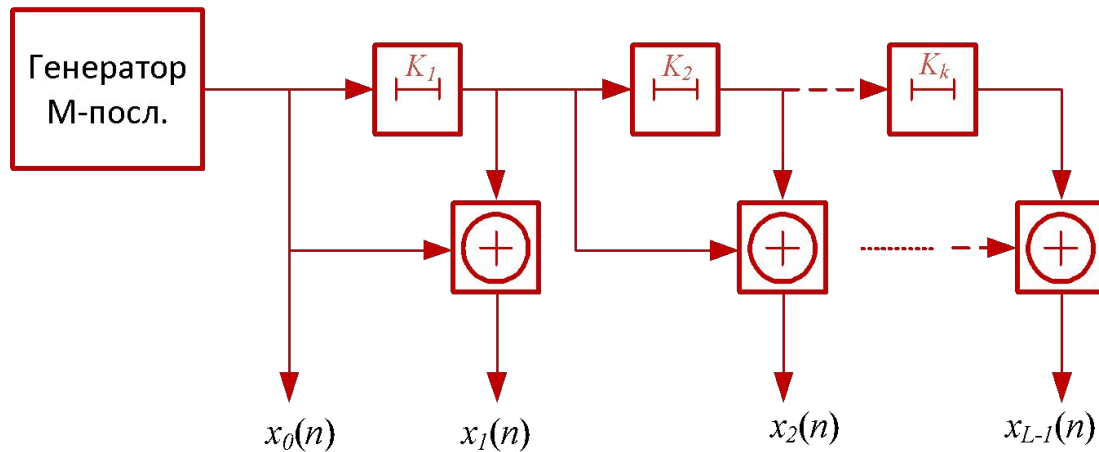


## Схема генератора с полиномом $z^4+z+1$



# Формирование случайного сигнала с равномерным распределением

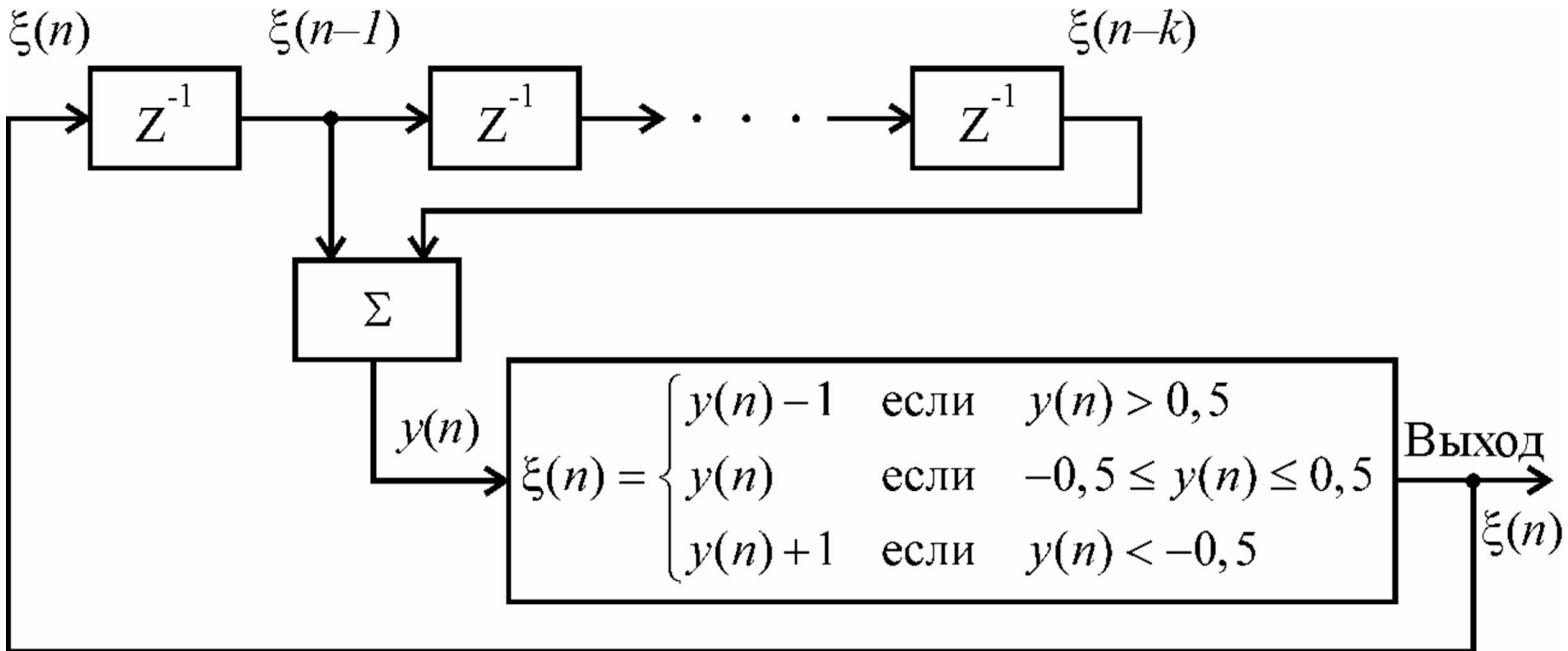
## Формирование многоразрядного псевдослучайного сигнала



# Формирование случайного сигнала с равномерным распределением

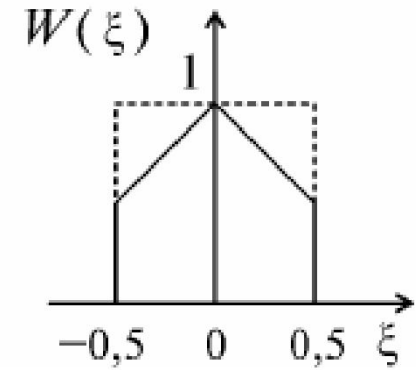
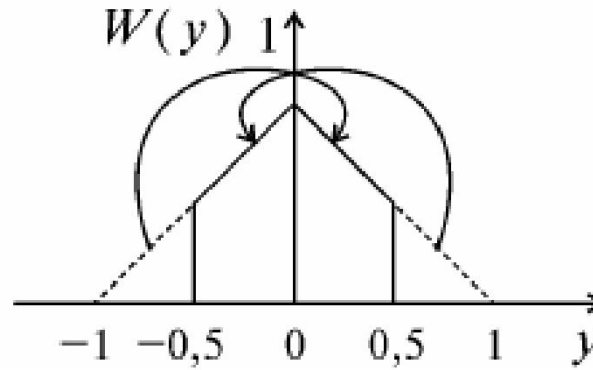
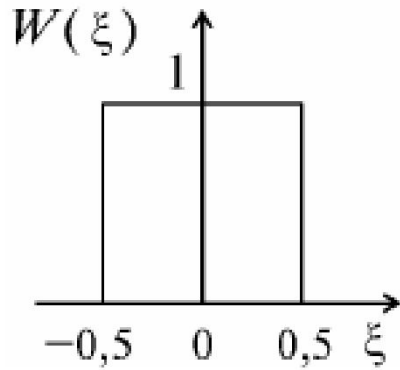
## Формирование многоразрядного псевдослучайного сигнала

$$\xi(n) = [\xi(n-1) + \xi(n-k)] (\text{mod } 0,5)$$



# Формирование случайного сигнала с равномерным распределением

## Формирование многоразрядного псевдослучайного сигнала



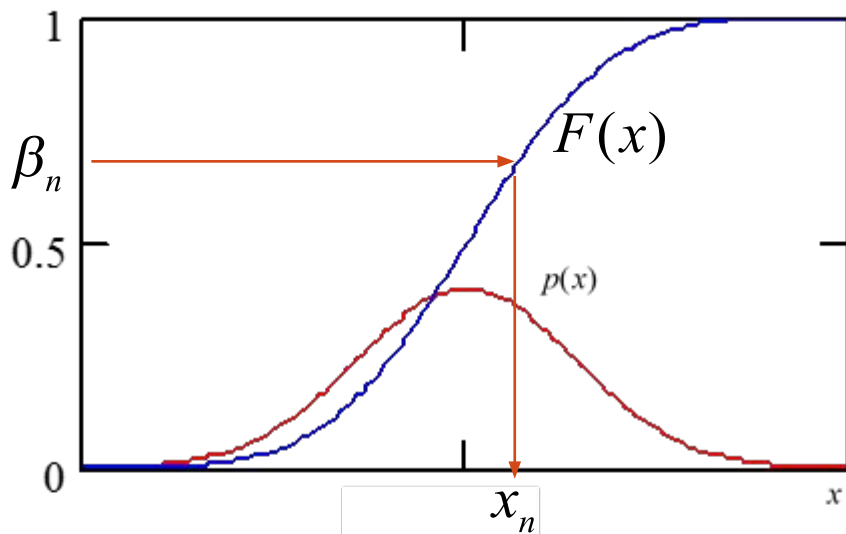
# Формирование случайного сигнала с произвольным распределением

## Метод обратной функции

Для любой  $X$  с плотностью распределения вероятности

$$p(x)$$

$F(x) = \int_{-\infty}^x p(x)dx$  - распределена по равномерному закону на интервале  $[0 \dots 1]$



1. Формируется  $\beta_n$  с равномерным распределением;

2. Решается уравнение

$$\beta_n = \int_{-\infty}^{x_n} p(x)dx$$

$$\beta_n = F(x_n)$$



# Формирование случайного сигнала с произвольным распределением

## Формирование сигнала с распределением Рэля

$$p(x) = \begin{cases} 0, & x \leq 0 \\ \frac{x^2}{\sigma^2} \exp\left(-\frac{x^2}{2\sigma^2}\right), & x > 0 \end{cases}$$

$$F(x) = 1 - \exp\left(-\frac{x^2}{2\sigma^2}\right)$$

$$x_n = \sigma \sqrt{-2 \ln(\beta_n)}$$

# Формирование случайного сигнала с нормальным распределением

## Формирование сигнала суммированием

$$x_n = \sum_{i=0}^{n-1} \beta_i \quad n \rightarrow \infty$$

На практике  $n > 8$

Если

$$x(n) = \sqrt{\frac{12}{n}} \sum_{i=0}^{n-1} (\beta_i - 0.5)$$

то  $\mu = 0$   
 $\sigma = 1$

# Формирование случайного сигнала с нормальным распределением

## Формирование методом полярных координат

На основе распределения Рэлея:

Если  $\beta_1, \beta_2$  - с равномерным распределением  $[0 \dots 1]$ , то:

$$\rho = \sigma \sqrt{-2 \ln(\beta_1)} \quad \varphi = 2\pi\beta_2$$

По определению закона Рэлея, проекции  $(\rho, \varphi)$  на координатные вектора будут распределены по нормальному закону  $\mu = 0$  и  $\sigma$  СКО

Таким образом:

$$x(n) = \sqrt{-2 \ln(\beta_1)} \cdot \sin(2\pi\beta_2)$$

$$y(n) = \sqrt{-2 \ln(\beta_1)} \cdot \cos(2\pi\beta_2)$$

# АЛГОРИТМЫ CORDIC

---

**CORDIC** (от англ. **CO**ordinate **R**otation **D**igital **C**omputer – цифровой вычислитель вращения координат) – итерационный метод сведения прямых вычислений сложных функций к выполнению операций сложения и сдвига.

Применяется в микроконтроллерах и ПЛИС, когда операции умножения являются ресурсоемкими

1624 г. Генри Бриггс использовал для составления таблиц логарифмов.

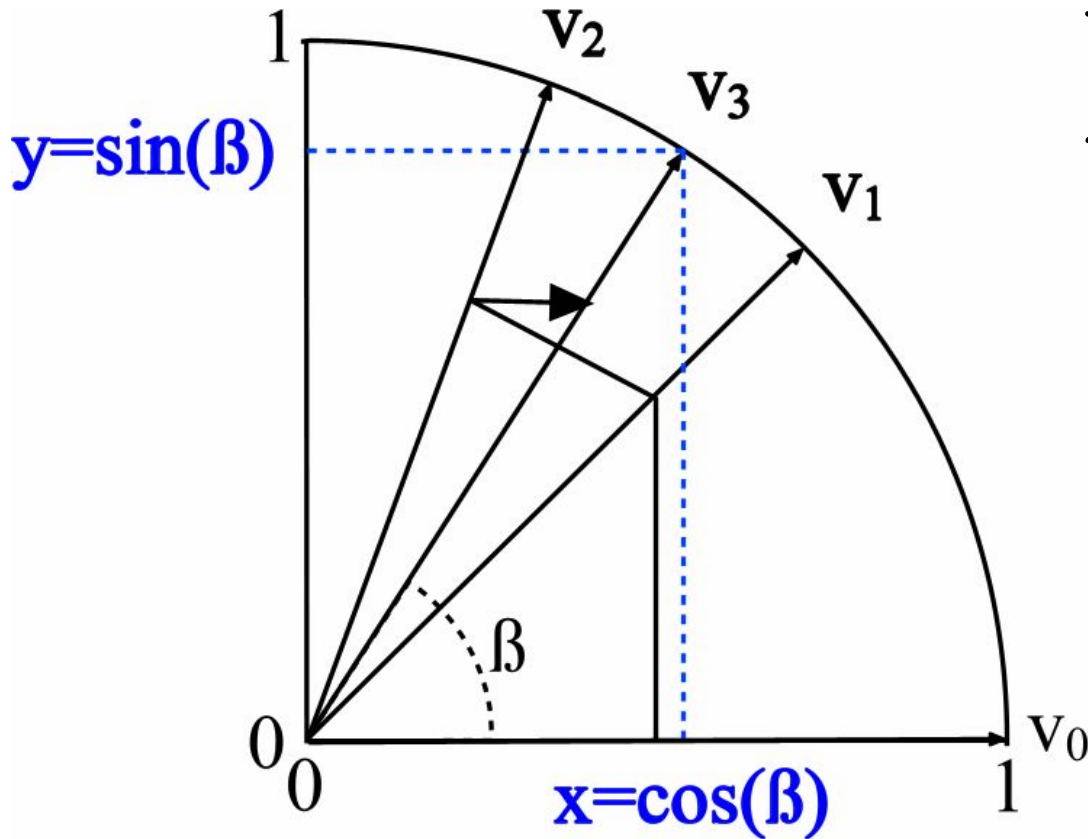
1956 Джек Волдер применил для вычисления тригонометрических функций

1975 В.Д. Байков систематизировал алгоритмы

В.Д. Байков, В.Б. Смоллов Специализированные процессоры: итерационные алгоритмы и структуры

# АЛГОРИТМЫ CORDIC

Основная идея – последовательное (итерационное) приближение аргумента (вектора) к нулю (единице)



$$x = x_0 \cos(\beta) - y_0 \sin(\beta)$$

$$y = x_0 \sin(\beta) - y_0 \cos(\beta)$$

или:

$$x = \cos(\beta)(x_0 - y_0 \operatorname{tg}(\beta))$$

$$x = \cos(\beta)(y_0 + x_0 \operatorname{tg}(\beta))$$

ПОЛОЖИВ:

$$\operatorname{tg}(\beta) = \pm 2^{-m}$$

# АЛГОРИТМЫ CORDIC

Таким образом, если произвольный угол представить в виде суммы углов:

$$\beta_m = \pm \arctan(2^{-m})$$

$$x_{m+1} = K_m (x_m - \sigma_m y_m 2^{-m})$$

$$y_{m+1} = K_m (y_m + \sigma_m x_m 2^{-m})$$

$$K_m = \frac{1}{\sqrt{1 + 2^{-2m}}}$$

$$\sigma_m \in \{-1, +1\}$$

$$x_{m+1} = x_m - \sigma_m y_m 2^{-m}$$

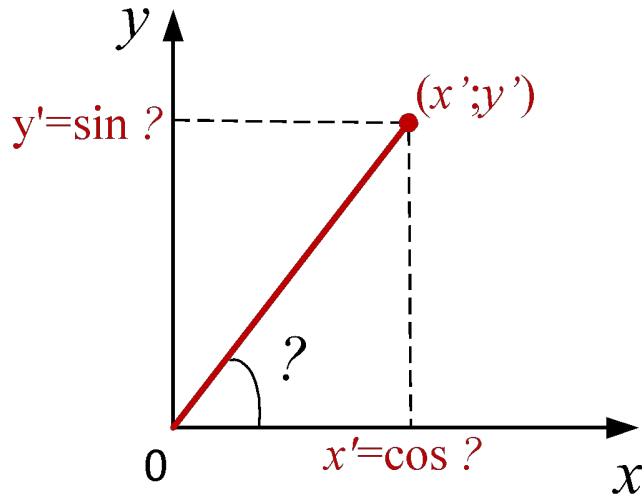
$$y_{m+1} = y_m + \sigma_m x_m 2^{-m}$$

$$K(n) = \prod_{m=0}^{n-1} K_m = \prod_{m=0}^{n-1} 1/\sqrt{1 + 2^{-2m}}$$

$$\sigma_m = \text{sign}(\beta_m)$$

# АЛГОРИТМЫ CORDIC

Пример:  $\cos \beta = ?$   $\sin \beta = ?$



Против часовой стрелки:

$$x_{n+1} = x_n \cos \beta_n - y_n \sin \beta_n$$

$$y_{n+1} = y_n \cos \beta_n + x_n \sin \beta_n$$

По часовой стрелке:

$$x_{n+1} = x_n \cos \beta_n + y_n \sin \beta_n$$

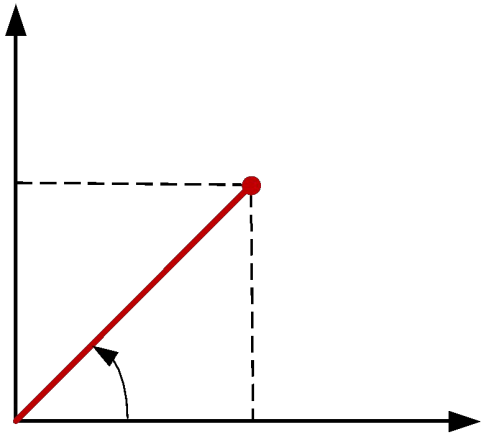
$$y_{n+1} = y_n \cos \beta_n - x_n \sin \beta_n$$

$$\beta_{n+1} = \beta_n / 2 \quad \beta_0 = 45'$$

$$x_0 = 1 \quad y_0 = 0$$

# АЛГОРИТМЫ CORDIC

Пример:  $\beta = 37^\circ$ ;  $\cos 37^\circ = ?$ ;  $\sin 37^\circ = ?$



Против часовой стрелки:

$$x_{n+1} = x_n \cos \beta_n - y_n \sin \beta_n$$

$$y_{n+1} = y_n \cos \beta_n + x_n \sin \beta_n$$

По часовой стрелке:

$$x_{n+1} = x_n \cos \beta_n + y_n \sin \beta_n$$

$$y_{n+1} = y_n \cos \beta_n - x_n \sin \beta_n$$

$$n = 1 \quad \beta_0 = 45^\circ \quad \beta_2 = 0^\circ + 45^\circ = 45^\circ$$

$0 < 37^\circ \Rightarrow$  против часовой  
стрелки

$$x_1 = 1 \cos 45^\circ - 0 \sin 45^\circ = 0,7$$

$$y_1 = 0 \cos 45^\circ + 1 \sin 45^\circ = 0,7$$

$$\beta_{n+1} = \beta_n / 2 \quad \beta_0 = 45^\circ$$

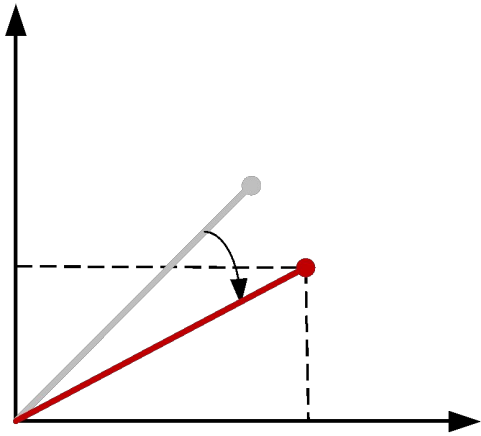
$$x_0 = 1 \quad y_0 = 0$$



# АЛГОРИТМЫ CORDIC

Пример:  $\beta = 37^\circ$ ;  $\cos 37^\circ = ?$ ;  $\sin 37^\circ = ?$

✕



$$n = 2 \quad \beta_1 = 22,5^\circ \quad \beta_2 = 45^\circ - 22,5^\circ = 22,5^\circ$$

$45^\circ > 37^\circ \Rightarrow$  по часовой стрелке

$$x_2 = 0,7 \cos 22,5^\circ + 0,7 \sin 22,5^\circ = 0,9$$

$$y_2 = 0,7 \cos 22,5^\circ - 0,7 \sin 22,5^\circ = 0,4$$

Против часовой стрелки:

$$x_{n+1} = x_n \cos \beta_n - y_n \sin \beta_n$$

$$y_{n+1} = y_n \cos \beta_n + x_n \sin \beta_n$$

По часовой стрелке:

$$x_{n+1} = x_n \cos \beta_n + y_n \sin \beta_n$$

$$y_{n+1} = y_n \cos \beta_n - x_n \sin \beta_n$$

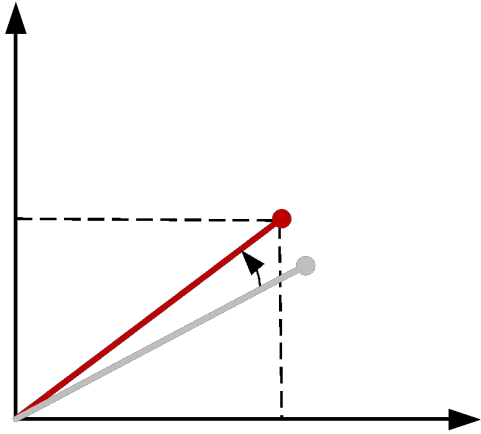
$$\beta_{n+1} = \beta_n / 2 \quad \beta_0 = 45'$$

$$x_0 = 1 \quad y_0 = 0$$

# АЛГОРИТМЫ CORDIC

Пример:  $\beta = 37^\circ$ ;  $\cos 37^\circ = ?$ ;  $\sin 37^\circ = ?$

✕



Против часовой стрелки:

$$x_{n+1} = x_n \cos \beta_n - y_n \sin \beta_n$$

$$y_{n+1} = y_n \cos \beta_n + x_n \sin \beta_n$$

По часовой стрелке:

$$x_{n+1} = x_n \cos \beta_n + y_n \sin \beta_n$$

$$y_{n+1} = y_n \cos \beta_n - x_n \sin \beta_n$$

$$n = 3 \quad \beta_2 = 11,3^\circ \quad \beta_3 = 22,5^\circ + 11,3^\circ = 33,8^\circ$$

$22,5^\circ < 37^\circ \Rightarrow$  против часовой стрелки

$$x_3 = 0,9 \cos 11,3^\circ - 0,4 \sin 11,3^\circ = 0,8$$

$$y_3 = 0,4 \cos 11,3^\circ + 0,9 \sin 11,3^\circ = 0,5$$

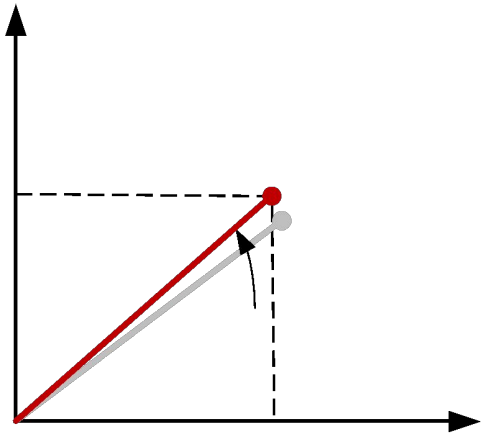
$$\beta_{n+1} = \beta_n / 2 \quad \beta_0 = 45'$$

$$x_0 = 1 \quad y_0 = 0$$

# АЛГОРИТМЫ CORDIC

Пример:  $\beta = 37^\circ$ ;  $\cos 37^\circ = ?$ ;  $\sin 37^\circ = ?$

✕



Против часовой стрелки:

$$x_{n+1} = x_n \cos \beta_n - y_n \sin \beta_n$$

$$y_{n+1} = y_n \cos \beta_n + x_n \sin \beta_n$$

По часовой стрелке:

$$x_{n+1} = x_n \cos \beta_n + y_n \sin \beta_n$$

$$y_{n+1} = y_n \cos \beta_n - x_n \sin \beta_n$$

$$n = 4 \quad \beta_3 = 5,6^\circ \quad \beta_4 = 33,8^\circ + 5,6^\circ = 39,4^\circ$$

$33,8^\circ < 37^\circ \Rightarrow$  против часовой стрелки

$$x_4 = 0,8 \cos 39,4^\circ - 0,5 \sin 39,4^\circ = 0,8$$

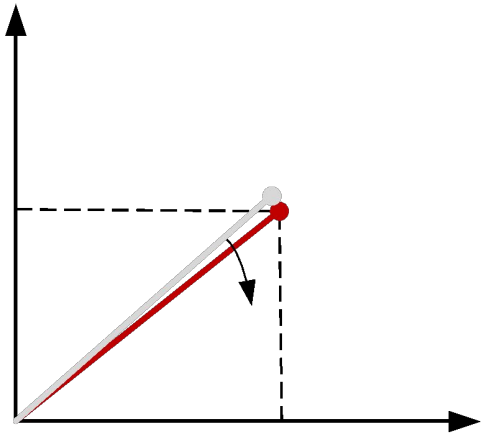
$$y_4 = 0,5 \cos 39,4^\circ + 0,8 \sin 39,4^\circ = 0,6$$

$$\beta_{n+1} = \beta_n / 2 \quad \beta_0 = 45'$$

$$x_0 = 1 \quad y_0 = 0$$

# АЛГОРИТМЫ CORDIC

Пример:  $\beta = 37^\circ$ ;  $\cos 37^\circ = ?$ ;  $\sin 37^\circ = ?$



Против часовой стрелки:

$$x_{n+1} = x_n \cos \beta_n - y_n \sin \beta_n$$

$$y_{n+1} = y_n \cos \beta_n + x_n \sin \beta_n$$

По часовой стрелке:

$$x_{n+1} = x_n \cos \beta_n + y_n \sin \beta_n$$

$$y_{n+1} = y_n \cos \beta_n - x_n \sin \beta_n$$

$$n = 5 \quad \beta_3 = 2,8^\circ \quad \beta_4 = 39,4^\circ - 2,8^\circ = 36,6^\circ$$

$33,8^\circ > 37^\circ \Rightarrow$  по часовой стрелке

$$x_5 = 0,8 \cos 2,8^\circ - 0,6 \sin 2,8^\circ = 0,803$$

$$y_5 = 0,6 \cos 2,8^\circ + 0,8 \sin 2,8^\circ = 0,596$$

... и так далее

$$\beta_{n+1} = \beta_n / 2 \quad \beta_0 = 45'$$

$$x_0 = 1 \quad y_0 = 0$$

# АЛГОРИТМЫ CORDIC

---

- значения функции  $\sin$  и  $\cos$  возможно вычислить заранее и хранить в памяти
- в данном примере для работы алгоритма требуются 2 операции умножения  
Преобразовав к виду:

Против часовой стрелки:

$$x_{n+1} = x_n \cos \beta_n - y_n \sin \beta_n \frac{\cos \beta_n}{\cos \beta_n} = \cos \beta_n (x_n - y_n \operatorname{tg} \beta_n)$$

$$y_{n+1} = \cos \beta_n (y_n + x_n \operatorname{tg} \beta_n)$$

По часовой стрелке:

$$x_{n+1} = \cos \beta_n (x_n + y_n \operatorname{tg} \beta_n)$$

$$y_{n+1} = \cos \beta_n (y_n - x_n \operatorname{tg} \beta_n)$$

потребуется 1 операция умножения на итерацию, а на последней необходимо будет скорректировать результат, умножив на постоянный множитель, пропорциональный числу итераций  
(**коэффициент деформации**)

# АЛГОРИТМЫ CORDIC

Если значение угла  $\beta$  выбирать не  $45^\circ, 45^\circ/2, 45^\circ/4, 45^\circ/8 \dots$ , а  $45^\circ, 26.565^\circ, 14.036^\circ, 7.125^\circ, 3.576^\circ, \dots$ , т.е. таким, чтобы  $\text{tg } \beta_n = 0,5; 0,25; 0,125, \dots$

$$\text{tg } \beta_n = 2^{-n} \Rightarrow \beta_n = \text{arctg}(2^{-n})$$

Тогда:

$$\begin{aligned} x_{n+1} &= x_n - \sigma_n y_n 2^{-n} \\ y_{n+1} &= y_n + \sigma_n x_n 2^{-n} \end{aligned}$$

$\sigma_m \in \{-1, +1\}$      $\sigma_m = \text{sign}(\beta_m)$     Таблица углов  $\beta_m$  заранее рассчитана и хранится в памяти

$$K_N = \prod_{m=0}^{N-1} K_n = \prod_{n=0}^{N-1} 1/\sqrt{1+2^{-2n}}$$

- коэффициент деформации, на который умножается результат; зависит только от числа итераций, поэтому может быть заранее рассчитан.

$$K_{N \rightarrow \infty} \approx 1.647$$

# АЛГОРИТМЫ CORDIC

Если значение угла  $\beta$  выбирать не  $45^\circ, 45^\circ/2, 45^\circ/4, 45^\circ/8 \dots$ , а  $45^\circ, 26.565^\circ, 14.036^\circ, 7.125^\circ, 3.576^\circ, \dots$ , т.е. таким, чтобы  $\text{tg } \beta_n = 0,5; 0,25; 0,125, \dots$

$$\text{tg } \beta_n = 2^{-n} \Rightarrow \boxed{\beta_n = \text{arctg}(2^{-n})}$$

Тогда:

$$\boxed{\begin{aligned} x_{n+1} &= x_m - \sigma_n y_n 2^{-n} \\ y_{n+1} &= y_m + \sigma_n x_n 2^{-n} \end{aligned}}$$

$$\sigma_m \in \{-1, +1\} \quad \sigma_m = \text{sign}(\beta_m)$$

$$K_N = \prod_{m=0}^{N-1} K_n = \prod_{n=0}^{N-1} 1/\sqrt{1+2^{-2n}}$$

- коэффициент деформации, на который умножается результат; зависит только от числа итераций, поэтому может быть заранее рассчитан.

# АЛГОРИТМЫ CORDIC

---

Пример реализации алгоритма:

AngTable = 45, 26.565, 14.036, 7.125, 3.576, 1.790, 0.895, 0.448

SumAngle = 0

LoopNum = 0

Если  $Y > 0$ :

$X_{\text{new}} = X + (Y \gg \text{LoopNum})$

$Y_{\text{new}} = Y - (X \gg \text{LoopNum})$

$\text{SumAngle} = \text{SumAngle} + \text{AngTable}[\text{LoopNum}]$

$\text{LoopNum} = \text{LoopNum} + 1$

Если  $Y < 0$ :

$X_{\text{new}} = X - (Y \gg \text{LoopNum})$

$Y_{\text{new}} = Y + (X \gg \text{LoopNum})$

$\text{SumAngle} = \text{SumAngle} - \text{AngTable}[\text{LoopNum}]$

$\text{LoopNum} = \text{LoopNum} + 1$

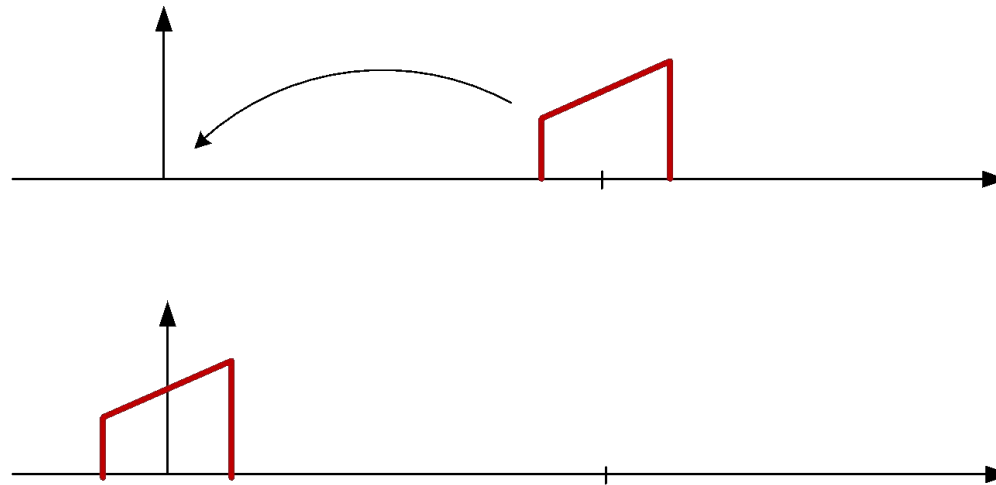


# Построение цифровых приёмных устройств

## Цифровое гетеродинирование

Назначение: перенос спектра сигнала с одной частоты на другую.

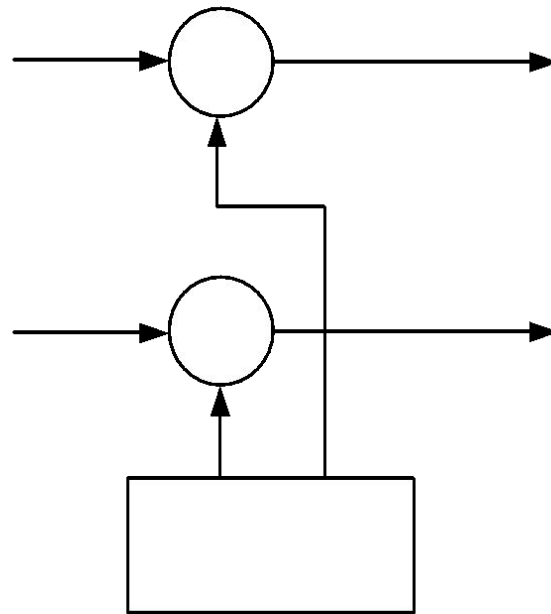
Применение: в цифровых приемниках прямого преобразования



# Цифровое гетеродинирование

Основан на свойстве преобразования Фурье: переносу спектра сигнала  $\omega_0$  на соответствующее умножение на комплексный гармонический сигнал этой частоты  $e^{j\omega_0}$

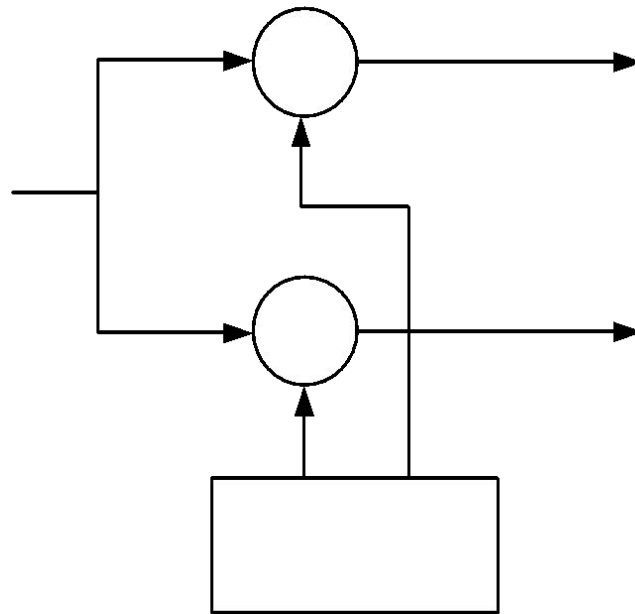
$$S(\omega - \omega_0) \Leftrightarrow e^{j\omega_0} s(n)$$



# Цифровое гетеродинирование

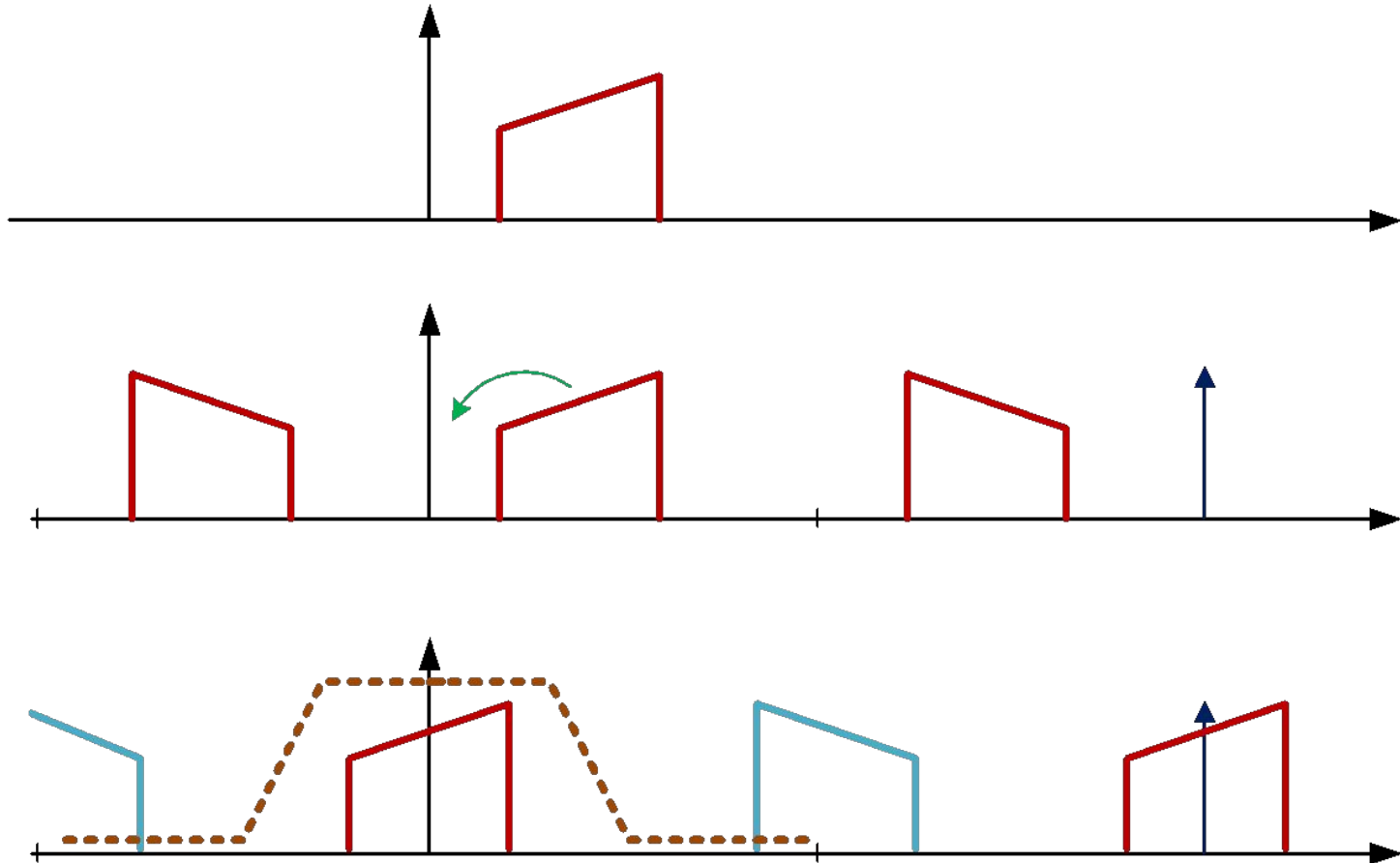
Основан на свойстве преобразования Фурье: переносу спектра сигнала  $\omega_0$  на соответствующее умножение на комплексный гармонический сигнал этой частоты  $e^{j\omega_0}$

$$S(\omega - \omega_0) \Leftrightarrow e^{j\omega_0} s(n)$$



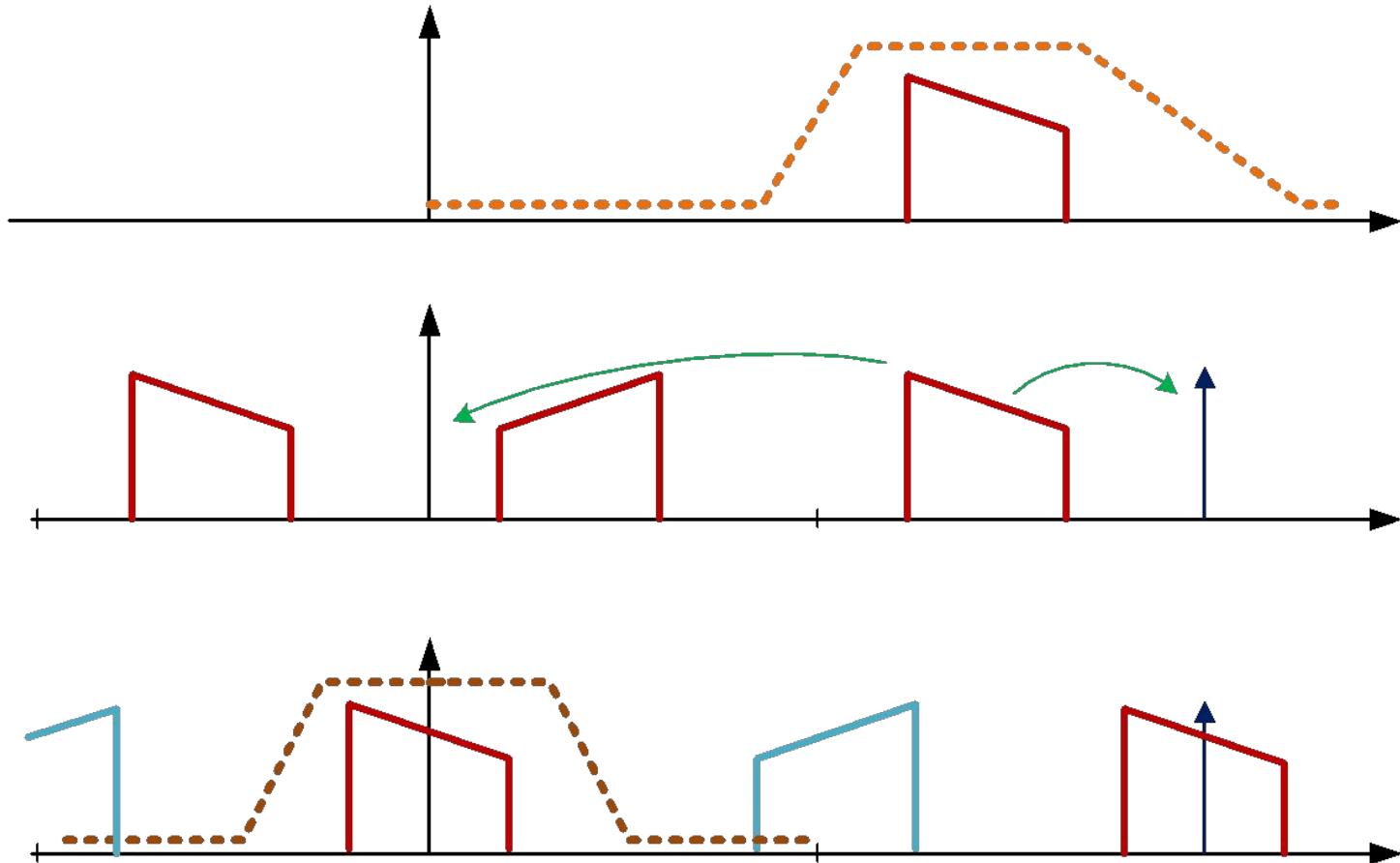
# Цифровое гетеродинирование

Пример гетеродинирования действительного сигнала



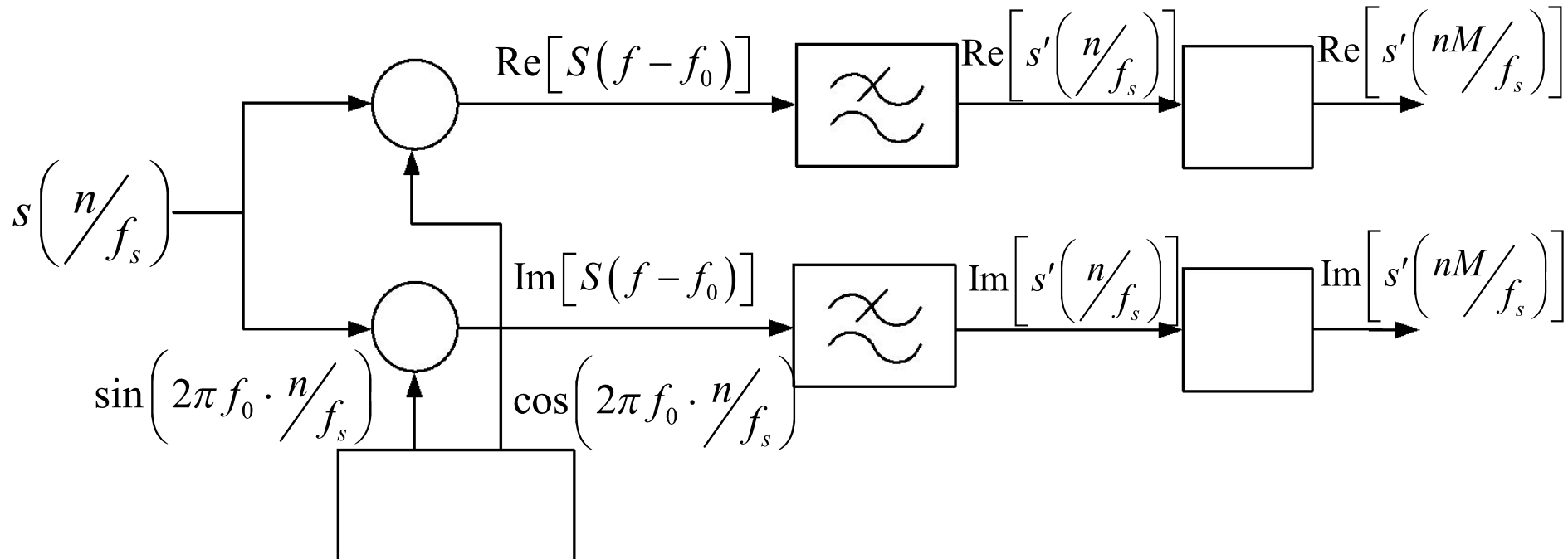
# Цифровое гетеродинирование

Пример гетеродинирования действительного сигнала (II зона Найквиста)

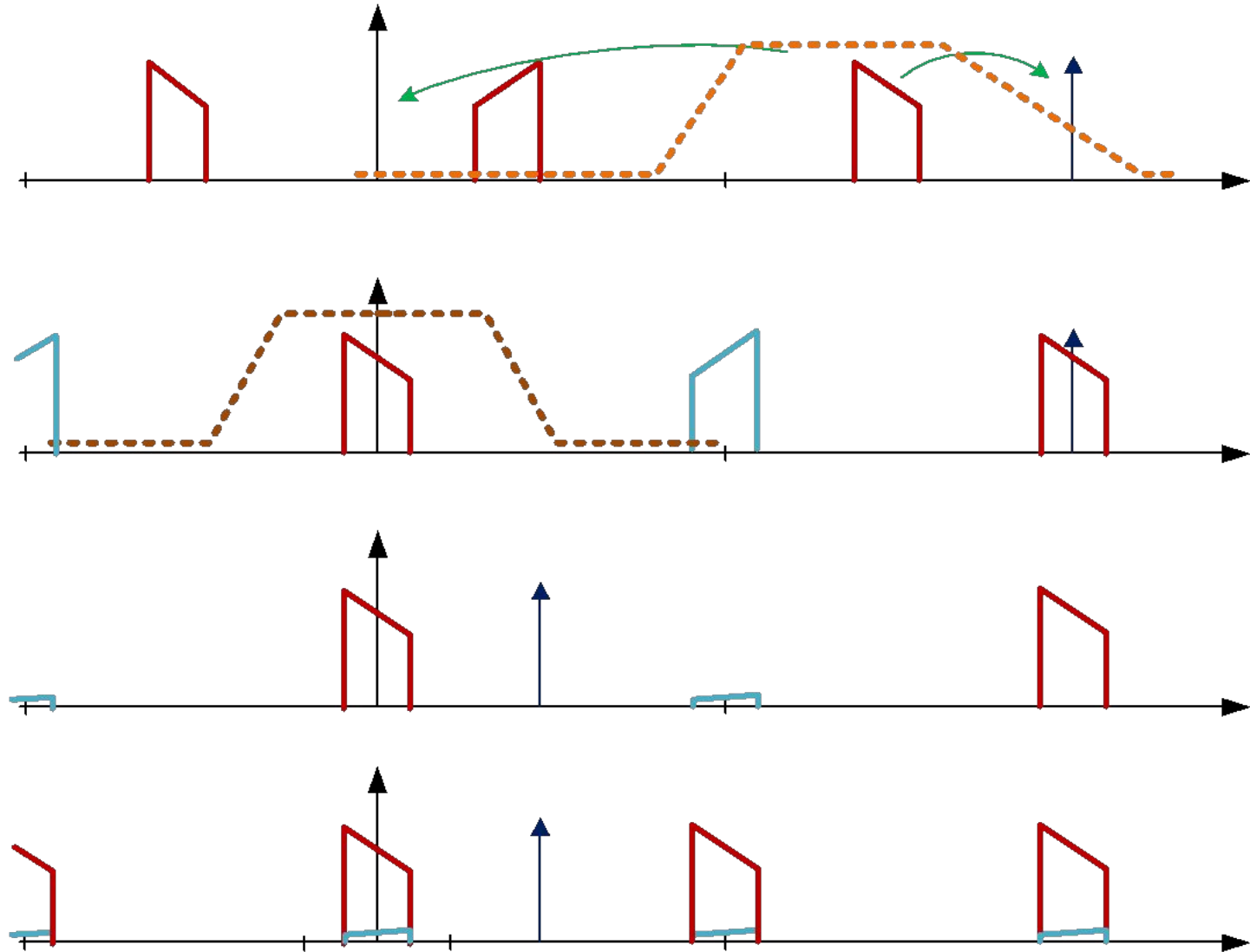


# Цифровое гетеродинирование

Для подавления нежелательной зеркальной составляющей в спектре сигнала используют фильтр нижних частот



# Цифровое гетеродинирование



# Прямой цифровой синтез

---

В качестве гетеродинов используются синтезаторы частоты (перестраиваемые генераторы), формирующие заданную частоту (набор частот) при фиксированной тактовой частоте.

Методы синтеза частоты:

- прямой цифровой синтез (Direct Digital Synthesis - DDS)
- косвенный (на основе ФАПЧ)
- гибридный (комбинация ФАПЧ и DDS)

Основные характеристики синтезаторов частоты

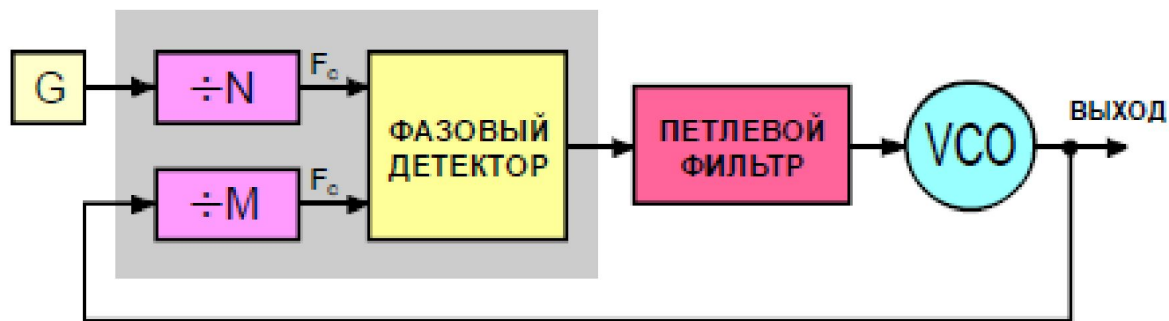
- диапазон перестройки (полоса частот выходного сигнала)
- скорость перестройки
- разрешение по частоте
- уровень паразитных спектральных составляющих (чистота спектра сигнала)
- неразрывность фаз выходного сигнала при перестройке



# Прямой цифровой синтез

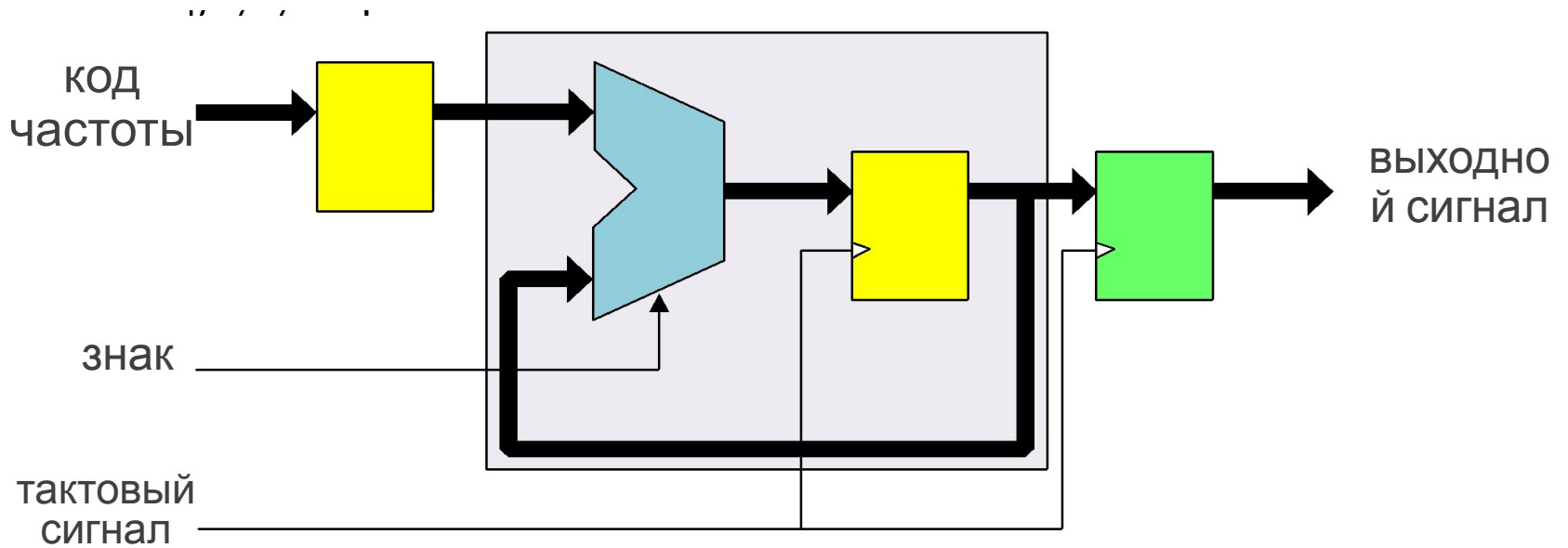
Синтезатор частоты на основе ФАПЧ (Phase Locked Loop - PLL)

Основной принцип: сравнение частоты (и фазы) выходного и опорного сигналов



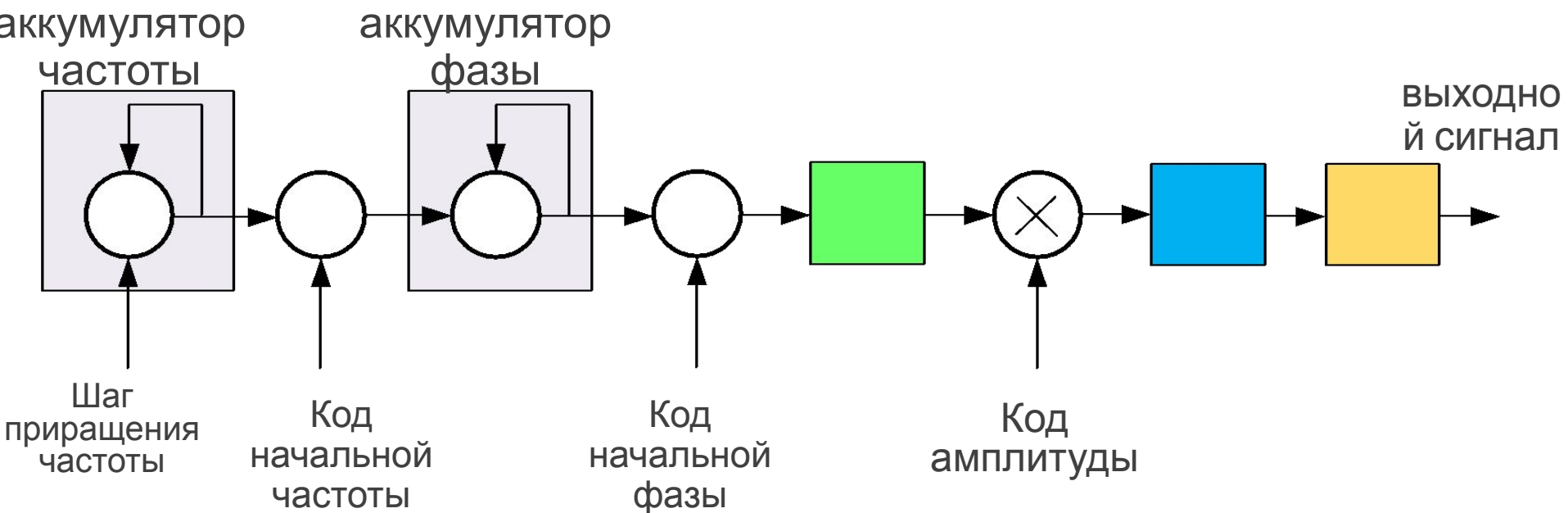
# Прямой цифровой синтез

Схема генератора прямого синтеза на основе накапливающего сумматора



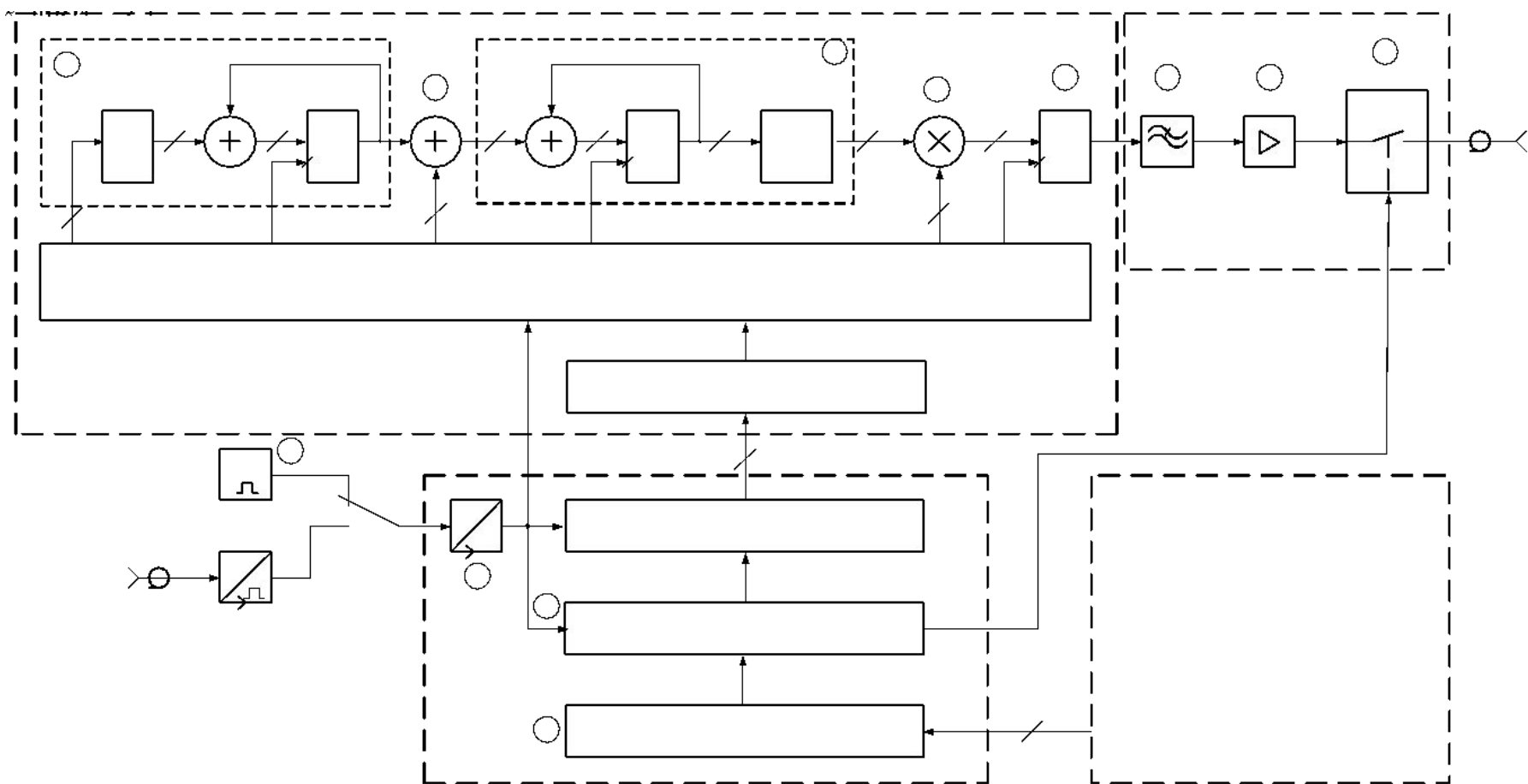
# Прямой цифровой синтез

Пример упрощённой схемы формирователя аналогового сигнала DDS



# Прямой цифровой синтез

Пример упрощённой схемы формирователя зондирующего сигнала РЛС методом прямого синтеза



# CIC-фильтры

---

Cascaded Integral-Comb filters - каскадные интегрально-гребенчатые фильтры, фильтры Хогенауэра (Hogenauer)

- наиболее часто применяются во многоскоростных системах
- не требуют операций умножения
- АЧХ соответствует фильтрам нижних частот
- легко перестраиваются при изменении коэффициента децимации (интерполяции)
- используют целочисленную арифметику

**Главный недостаток: плохие частотно-избирательные свойства**

1. E. B. Hogenauer. An economical class of digital filters for decimation and interpolation. IEEE Transactions on Acoustics, Speech and Signal Processing, ASSP-29(2):155–162, 1981

2. Использование CIC фильтров в задачах децимации и интерполяции сигналов [Электронный ресурс] // DSPlib.ru URL: <http://www.dsplib.ru/content/cicid/cicid.html> (Дата обращения: 19.10.2014)

# СИС-фильтры

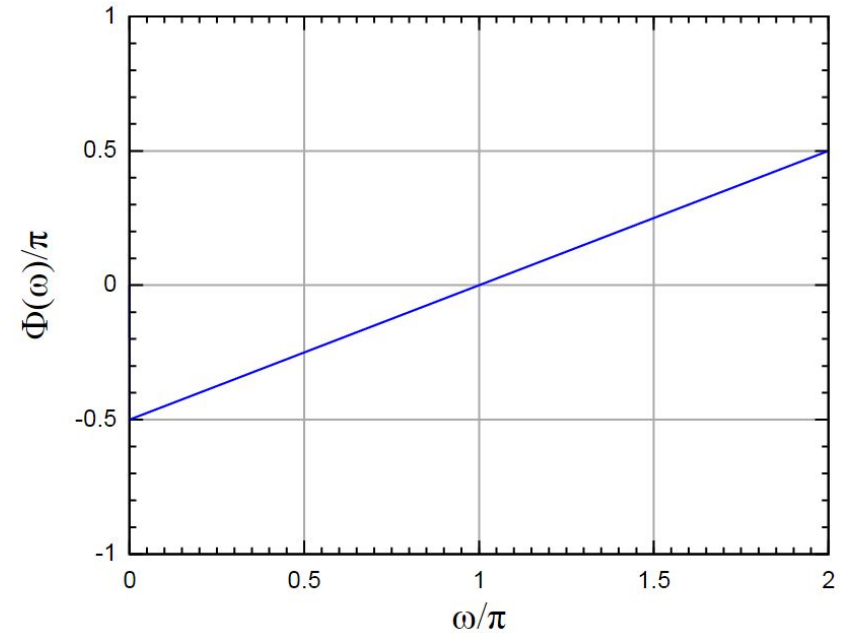
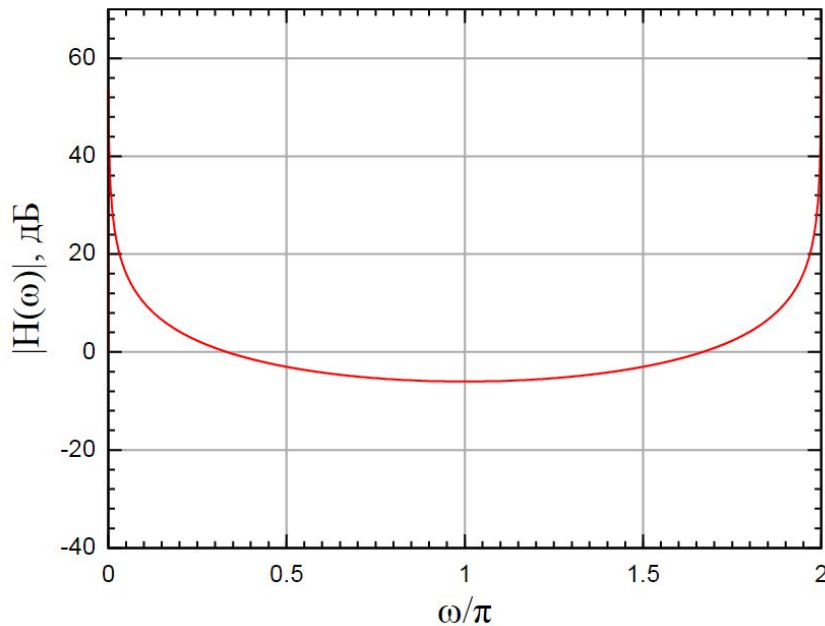
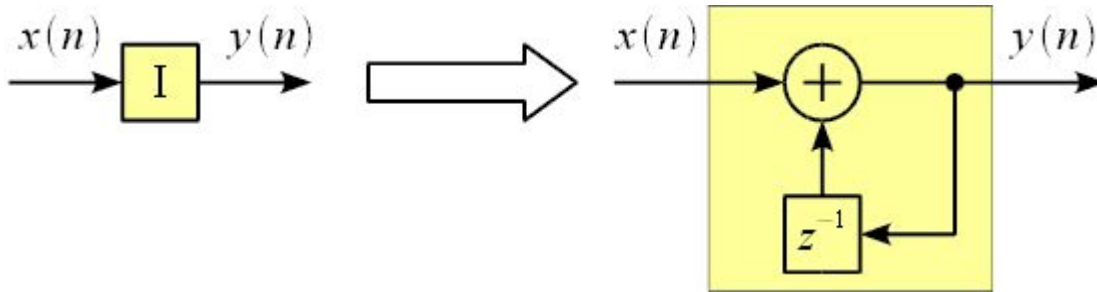
## Интегратор

$$y(n] = y[n-1] + x[n]$$

$$H_I(z) = \frac{1}{1-z^{-1}}$$

$$|H_I(e^{j\omega})| = \frac{1}{2 \cdot \sin(\omega/2)}$$

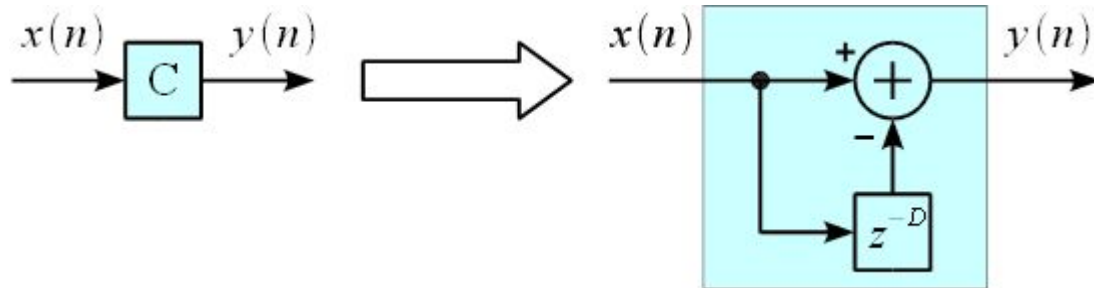
$$\Phi(H_I(e^{j\omega})) = \omega/2 - \pi/2$$



# СIS-фильтры

Гребенчатый  
КИХ

$$y(n) = x(n) - x(n-D), \quad H_c(z) = 1 - z^{-D}$$

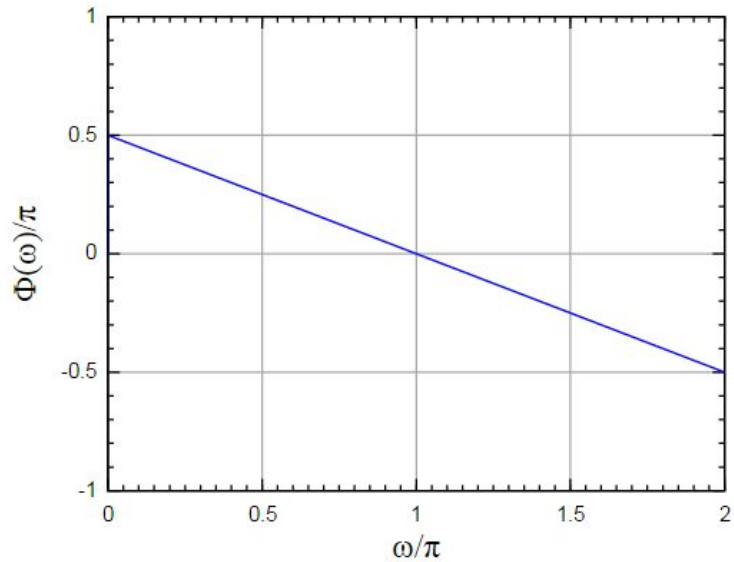
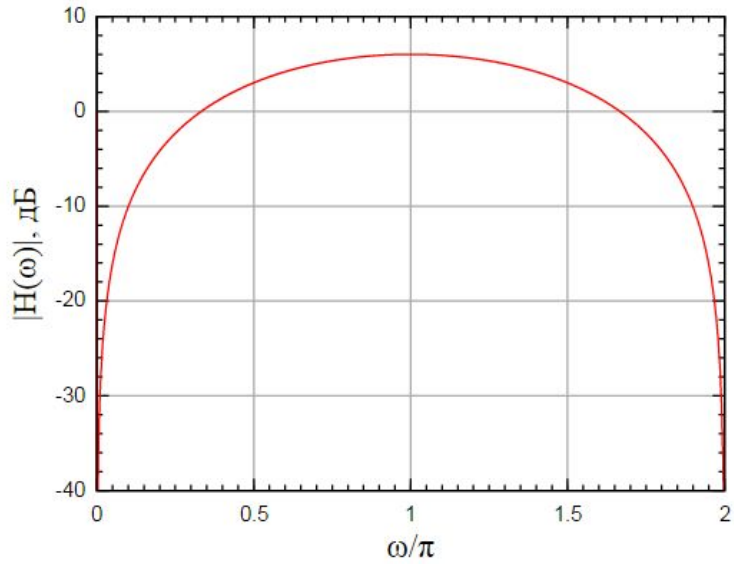


Частотная  
характеристика:

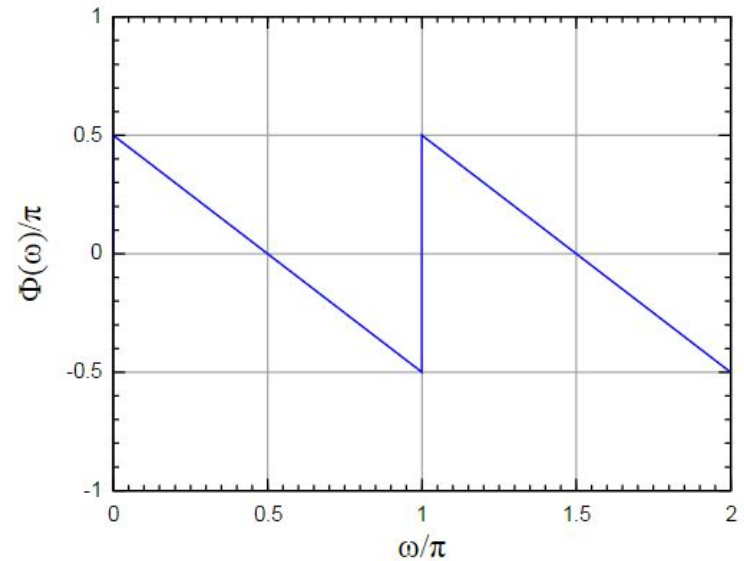
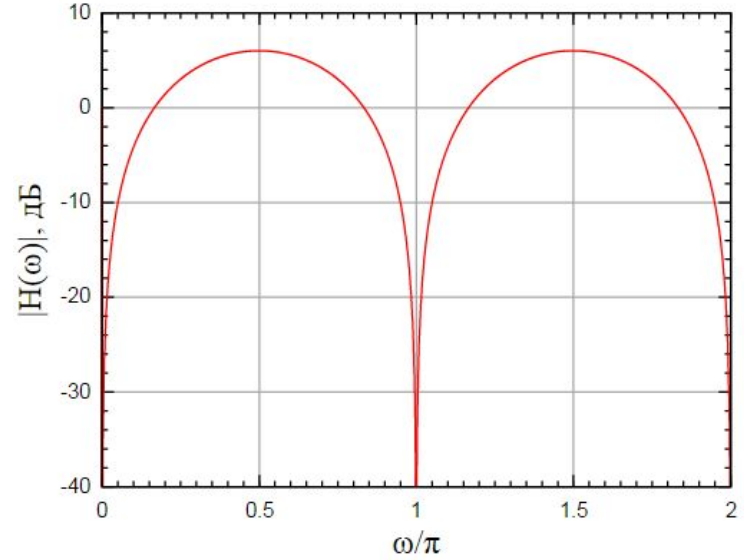
$$\begin{aligned} |H_c(e^{j\omega})| &= 2 \cdot |\sin(D \cdot \omega / 2)|; \\ \Phi(H_c(e^{j\omega})) &= -D \cdot \omega / 2 + \pi / 2. \end{aligned}$$

# Гребенчатый КИХ фильтр (ЧХ)

$D=1$



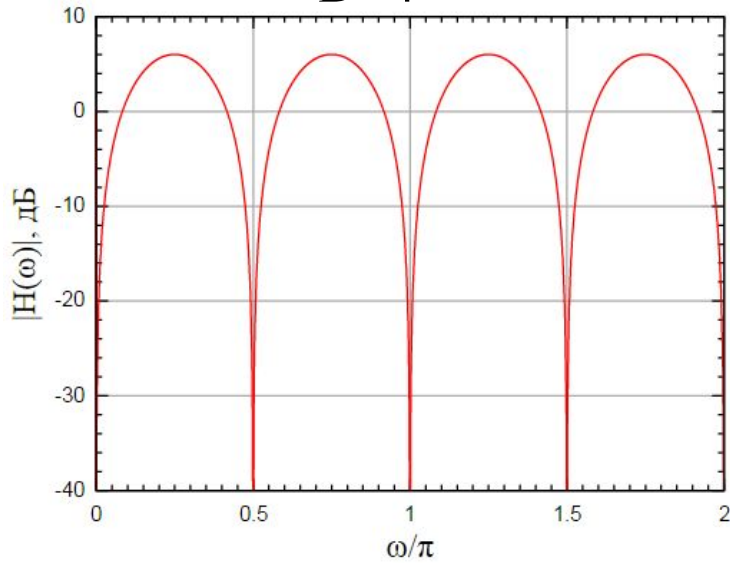
$D=2$



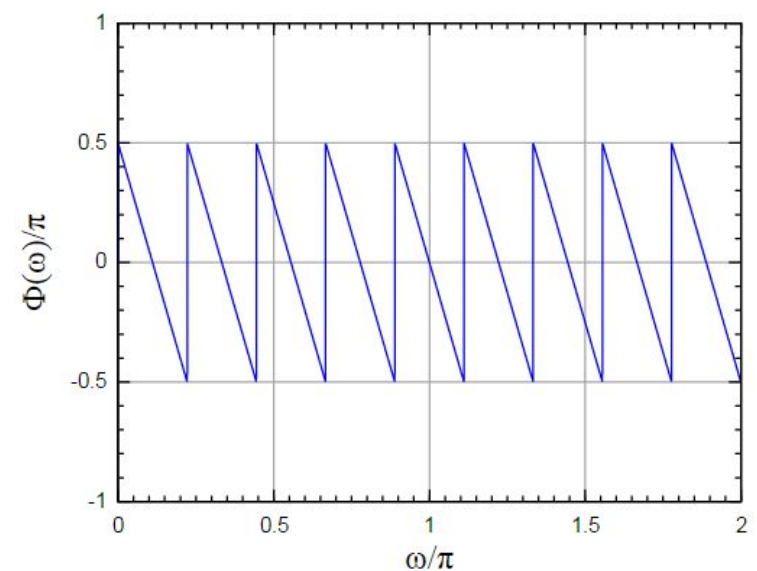
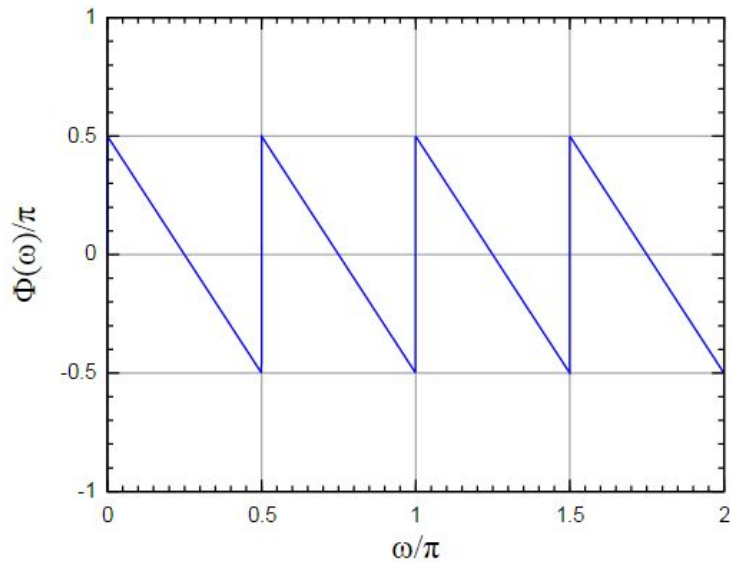
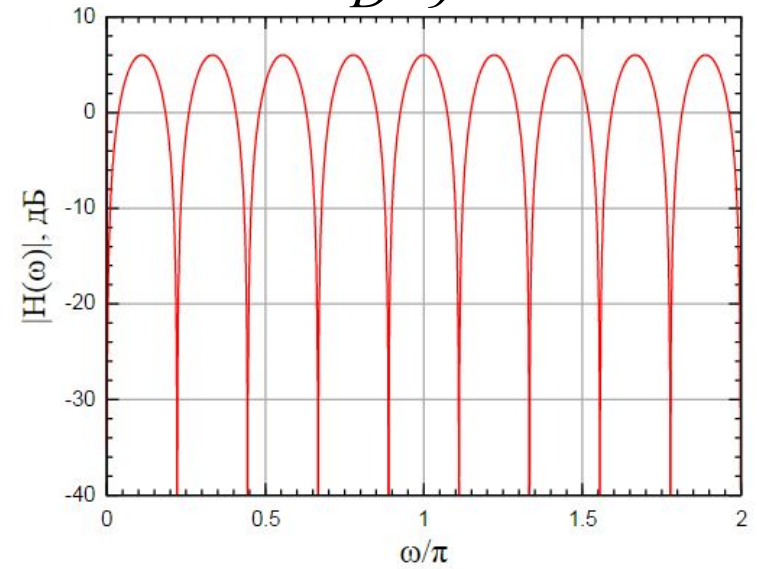


# Гребенчатый КИХ фильтр (ЧХ)

$D=4$



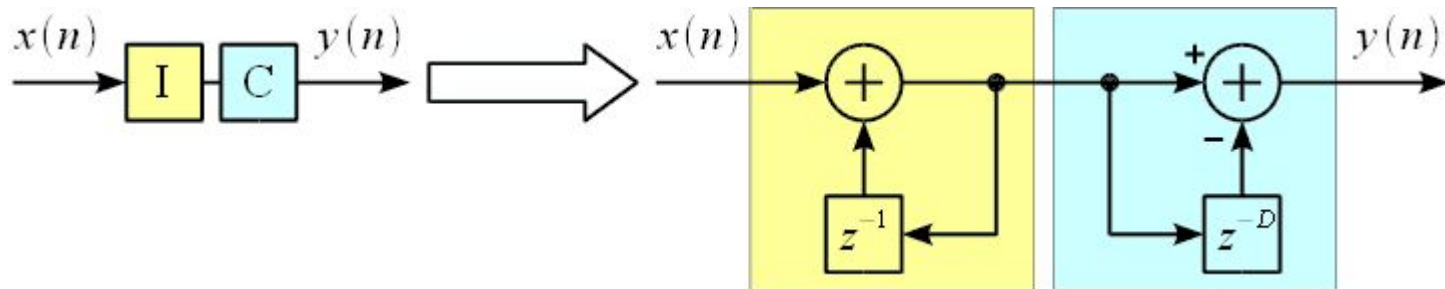
$D=9$



# Каскадное соединение интегратора и гребенчатого КИХ

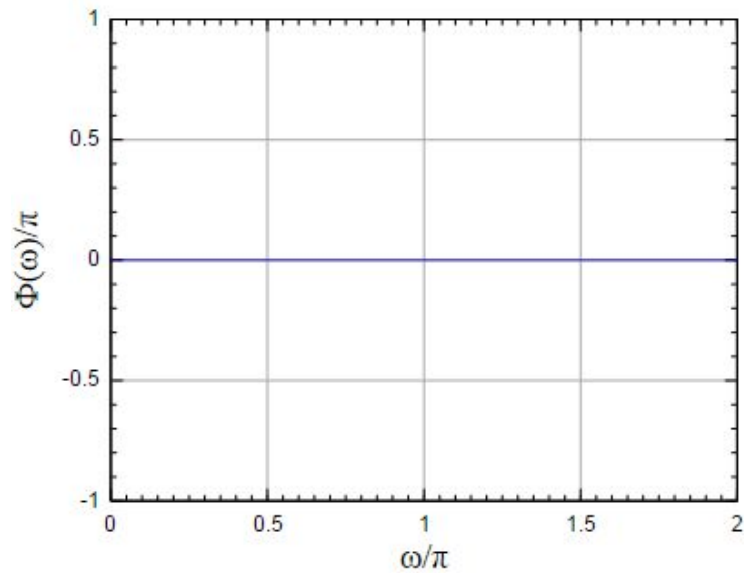
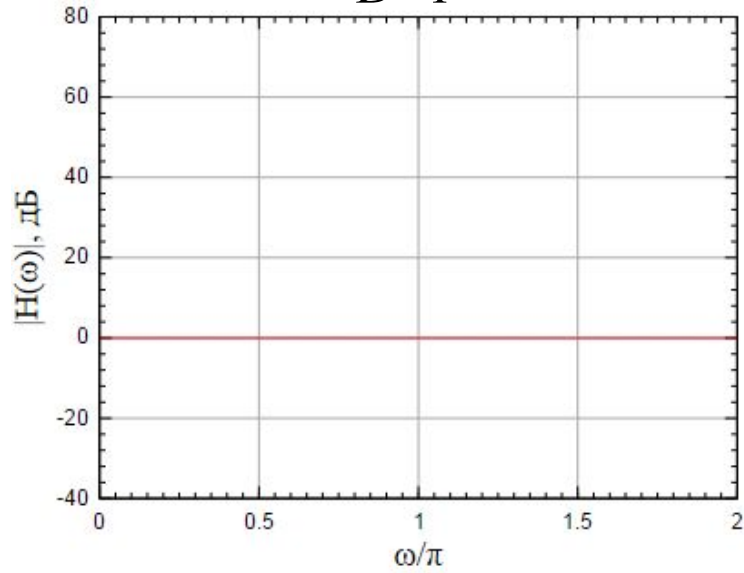
$$H_{CK}(z) = \frac{1 - z^{-D}}{1 - z^{-1}}$$

$$H_{CK}(e^{j\omega}) = \frac{\sin(D \cdot \omega/2)}{\sin(\omega/2)} \cdot \exp(-j \cdot (D-1) \cdot \omega/2)$$

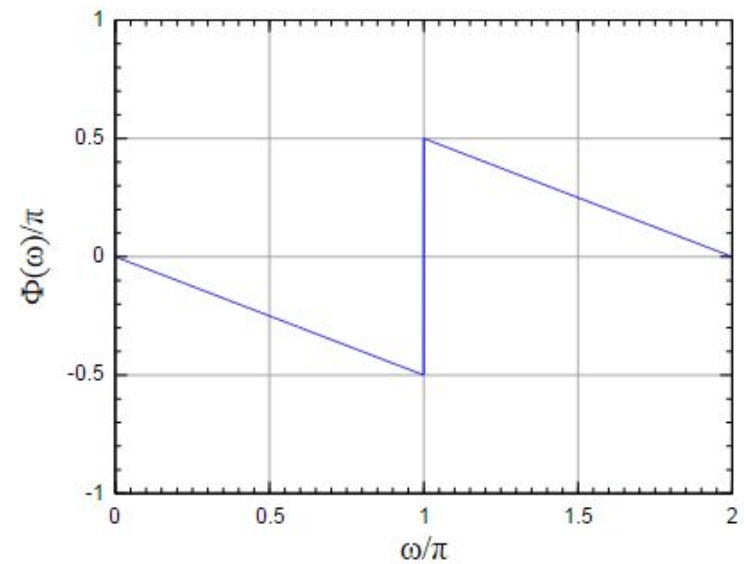
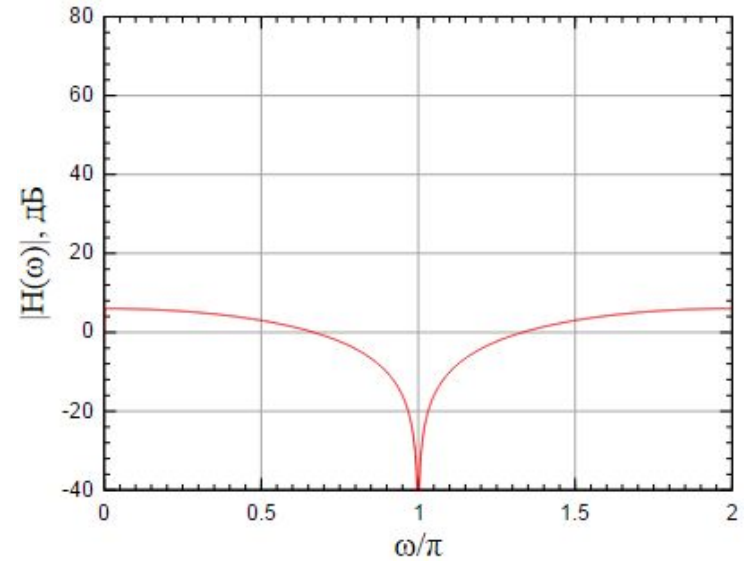


# Частотная характеристика СС фильтра

$D=1$

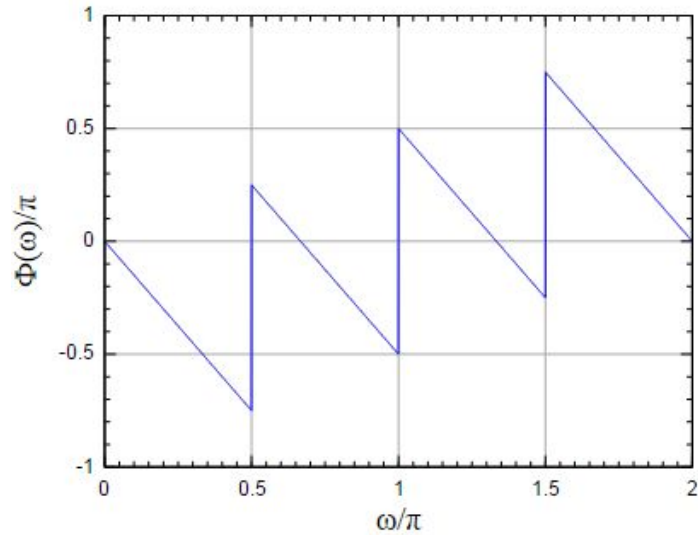
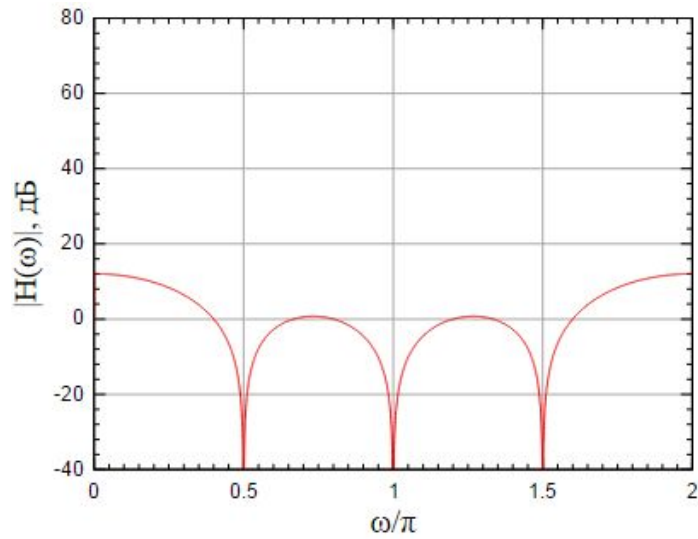


$D=2$

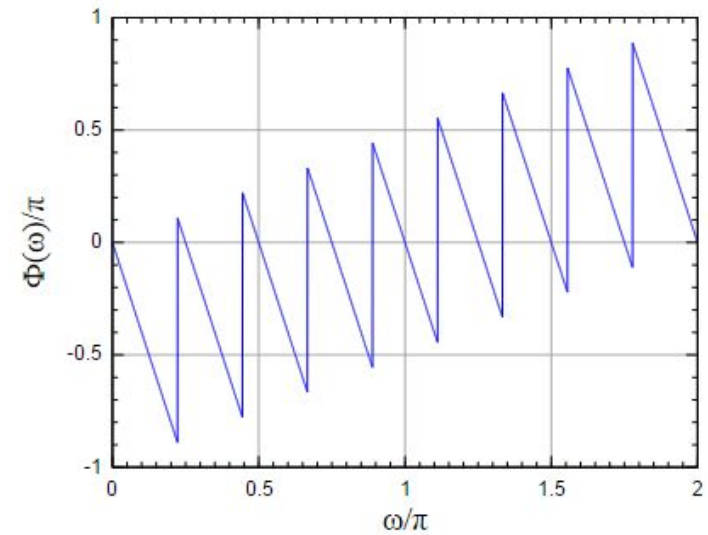
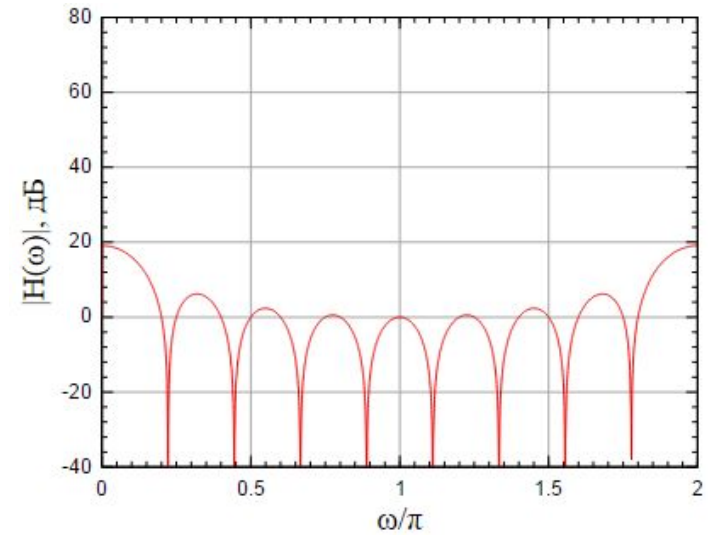


# Частотная характеристика СИС фильтра

$D=4$



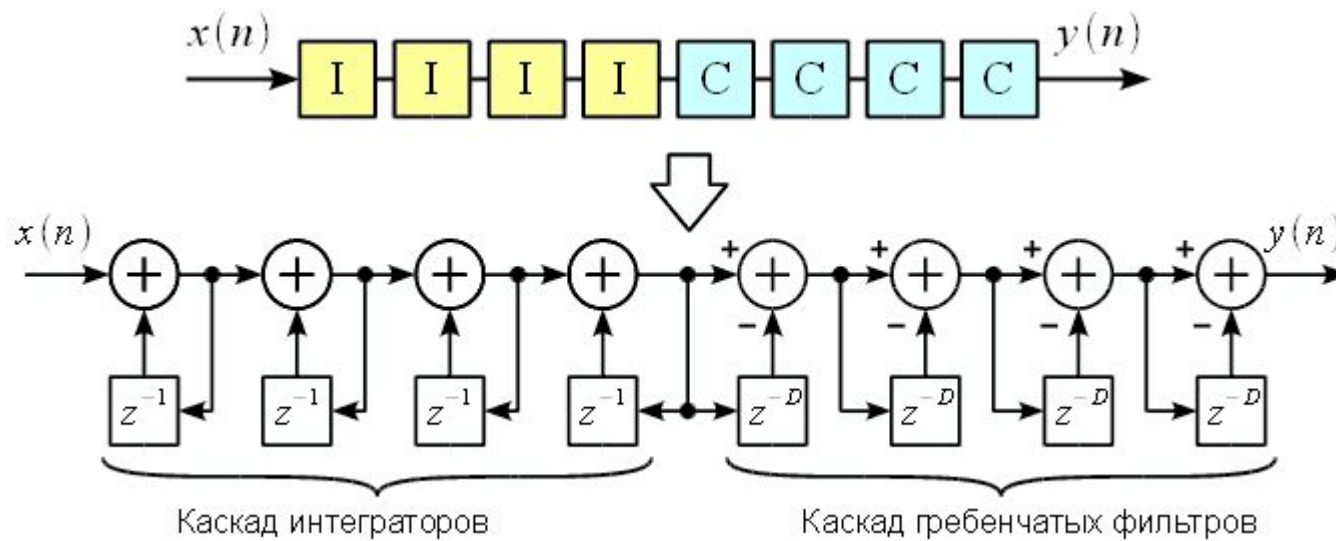
$D=9$



# СІС фильтр высших порядков

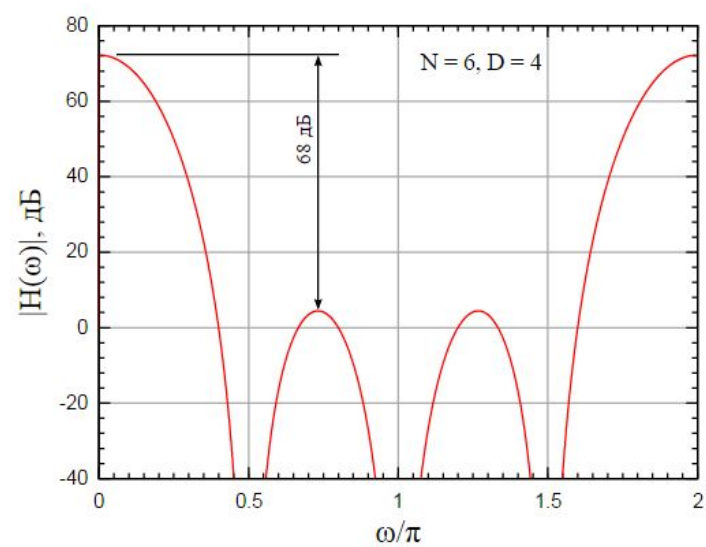
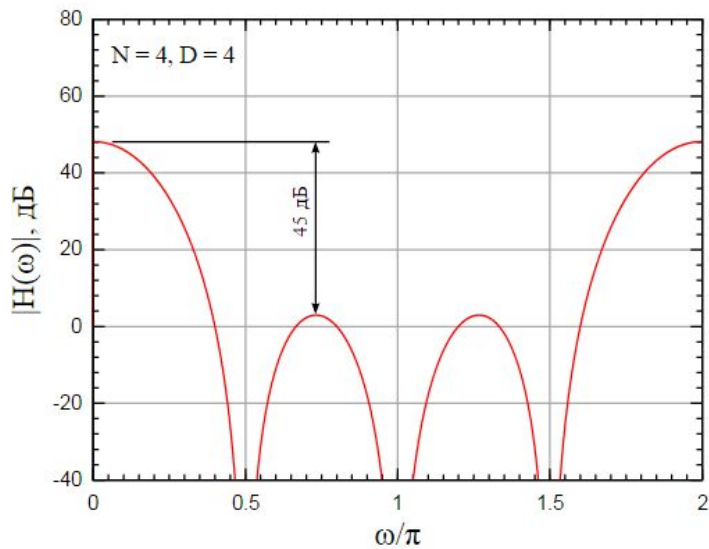
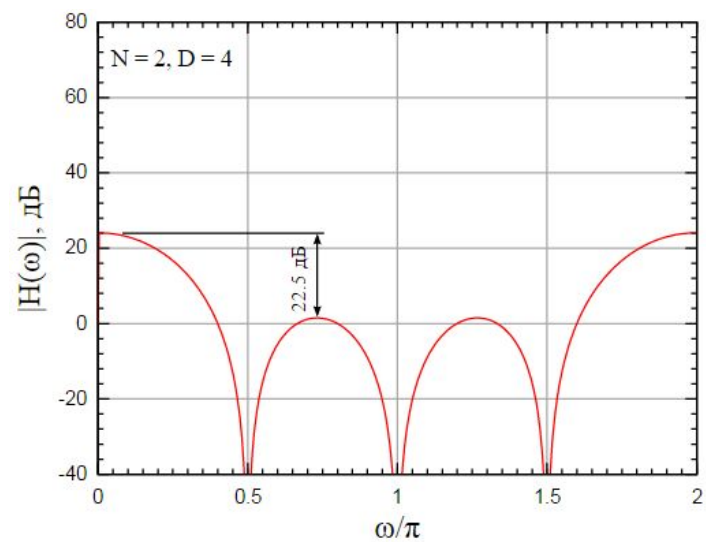
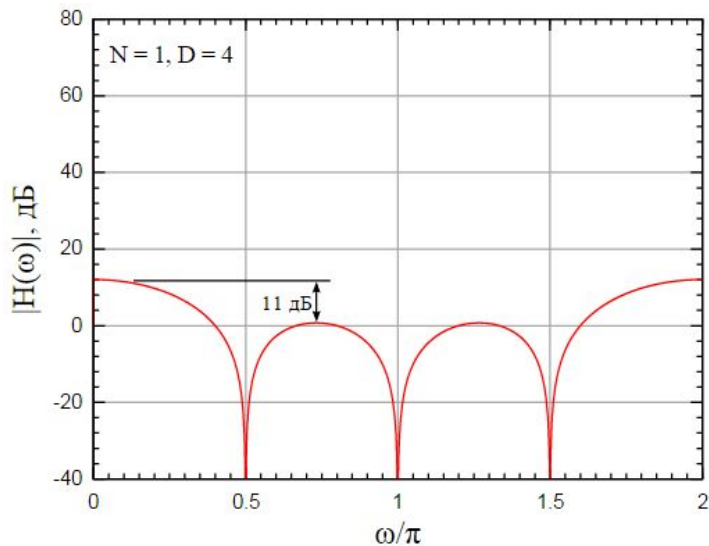
$$H_{CIC}(z) = \left( \frac{1 - z^{-D}}{1 - z^{-1}} \right)^N$$

$$N=4$$



$$H_{CIC}(e^{j\omega}) = \left[ \frac{\sin(D\omega/2)}{\sin(\omega/2)} \right]^N \cdot \exp(-j \cdot N \cdot (D-1) \cdot \omega/2).$$

# ЧХ СЧ фильтров в зависимости от порядка N



# Рост разрядности СІС фильтров

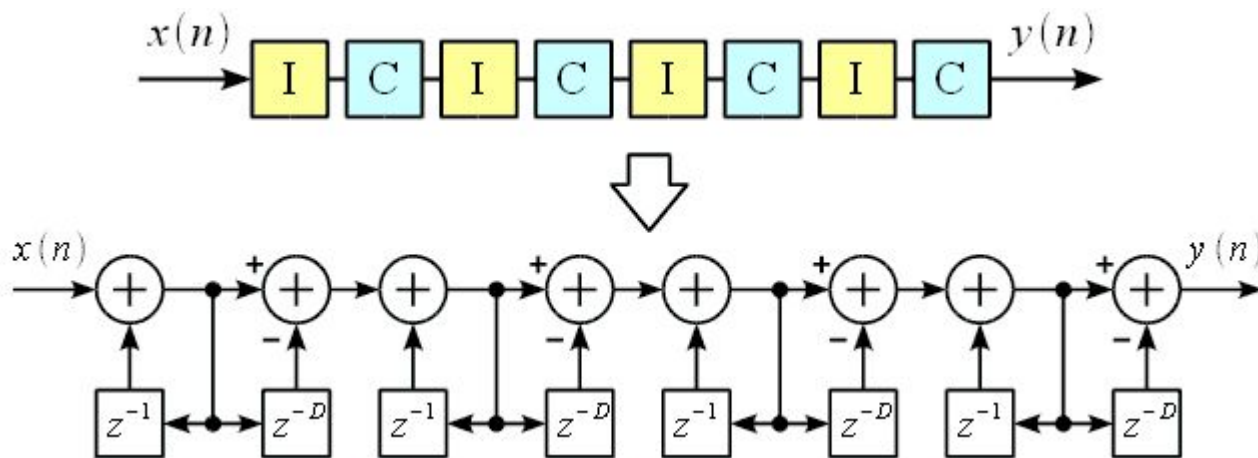
Коэффициент передачи на нулевой частоте:

$$K_0 = 20 \cdot \log_{10}(D^N) = N \cdot 20 \cdot \log_{10}(D) \text{ дБ}$$

Например, при  $D=4$ ,  $N=6$ :  $K_0=72$  дБ – рост разрядности данных в фильтре

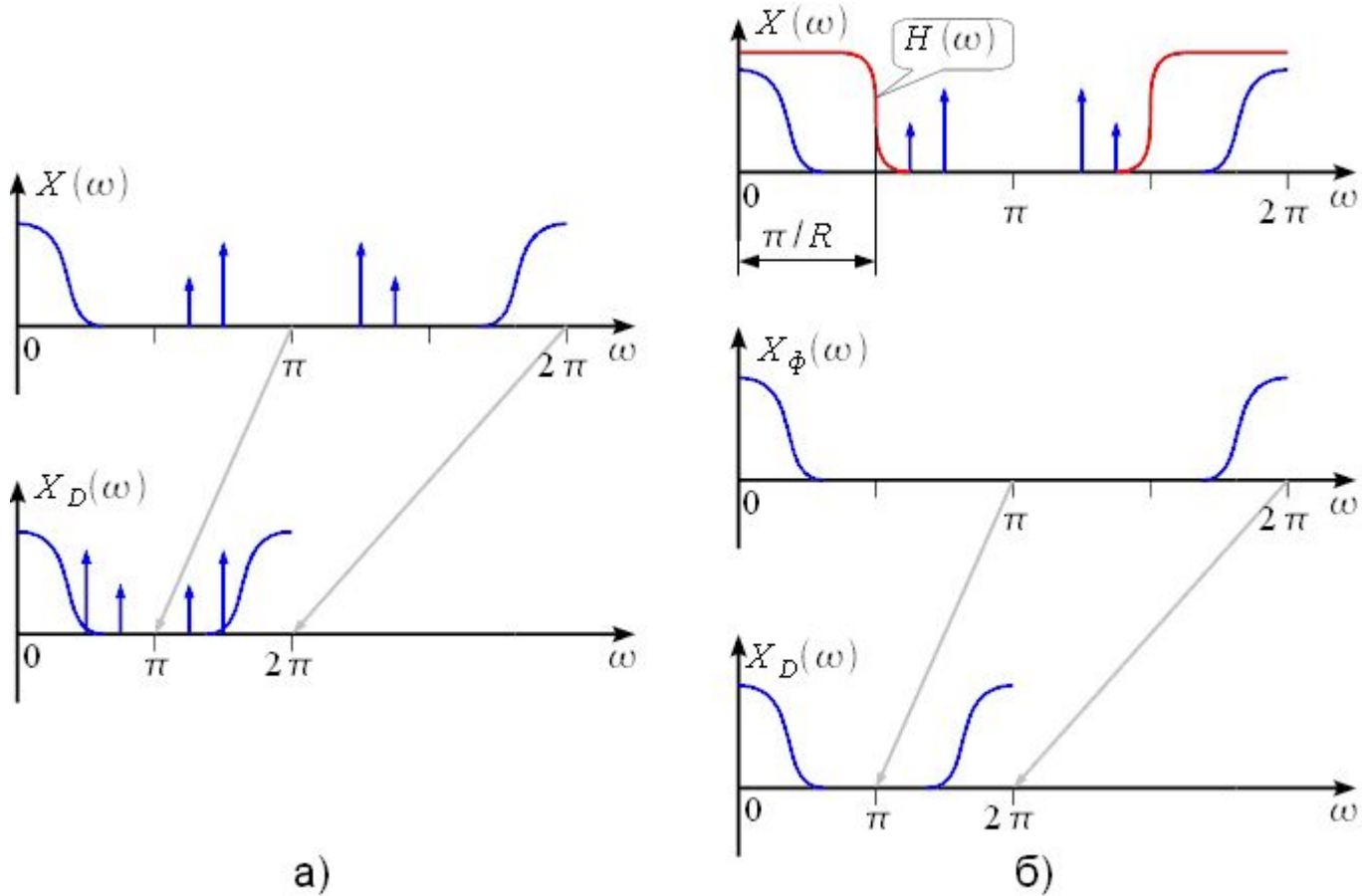
при  $D=9$ ,  $N=6$   $K_0=114,5$  дБ

Округление на промежуточных стадиях не допустимо!



# СIS фильтры в задачах децимации

Коэффициент прореживания (децимации)  $R=2$



Требуемая частота среза  $\pi/R$   
ФНЧ:



# СІС фильтры в задачах децимации

Требуемая частота среза  $\pi/R$

ФНЧ:

Первый ноль АЧХ СІС

$$\omega = 2 \cdot \pi / D \Rightarrow D = 2 \cdot R$$

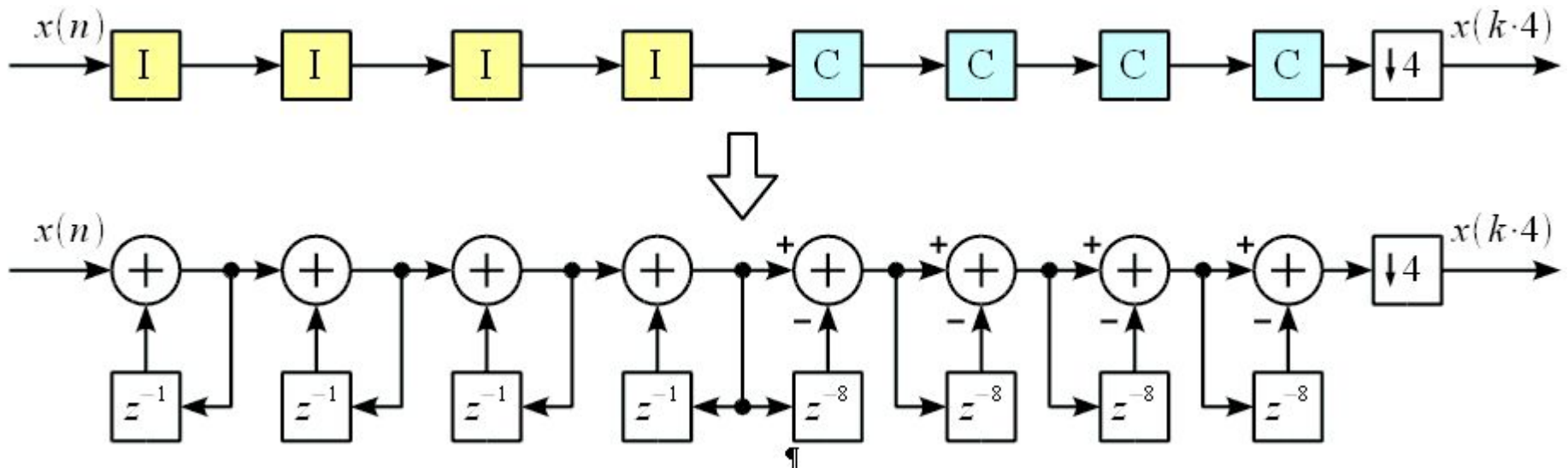
фильтра:

Порядок  $N$  обеспечивает требуемое подавление в полосе заграждения

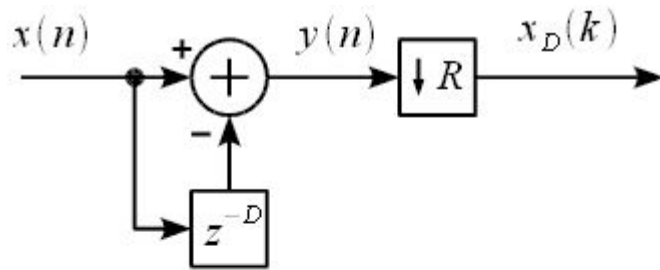
Пример: для прореживания в 4 раза с подавлением  $>40$ дБ

$$R = 4, D = 8, N \geq \left\lceil \frac{40}{(13 \dots)} \right\rceil \geq 4$$

$$K_0 = N \cdot 20 \cdot \log_{10}(D) = 80 \cdot \log_{10}(8) = 72.25 \text{ дБ}$$

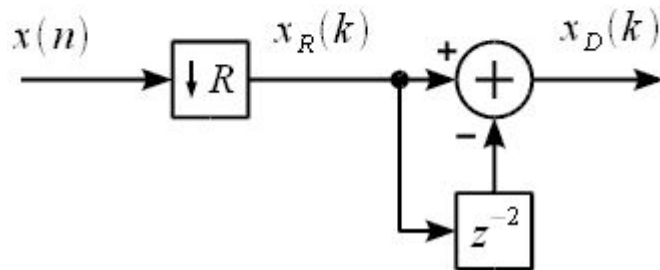


# СІС фильтры в задачах децимации



$$\begin{aligned}y(n) &= x(n) - x(n-D); \\x_D(k) &= y(R \cdot k); \\x_D(k) &= x(k \cdot R) - x(k \cdot R - D).\end{aligned}$$

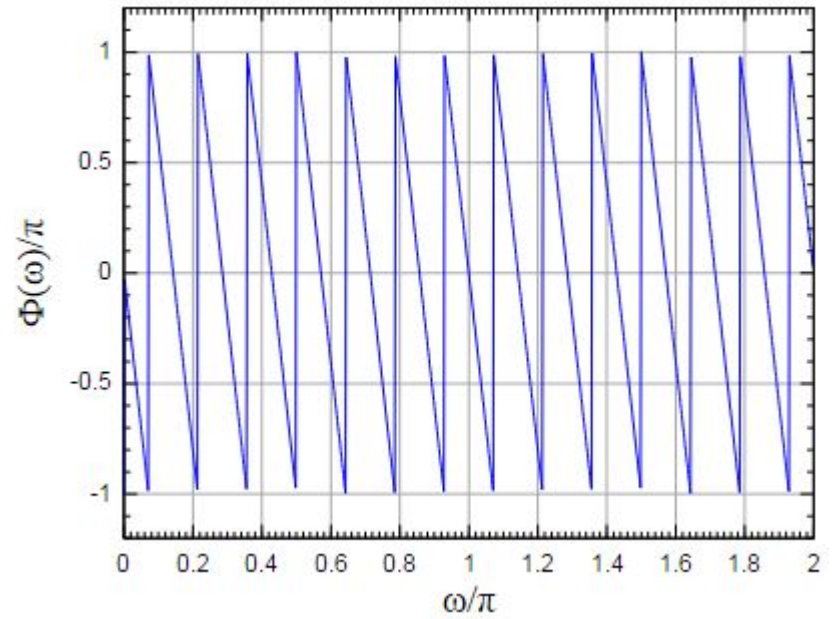
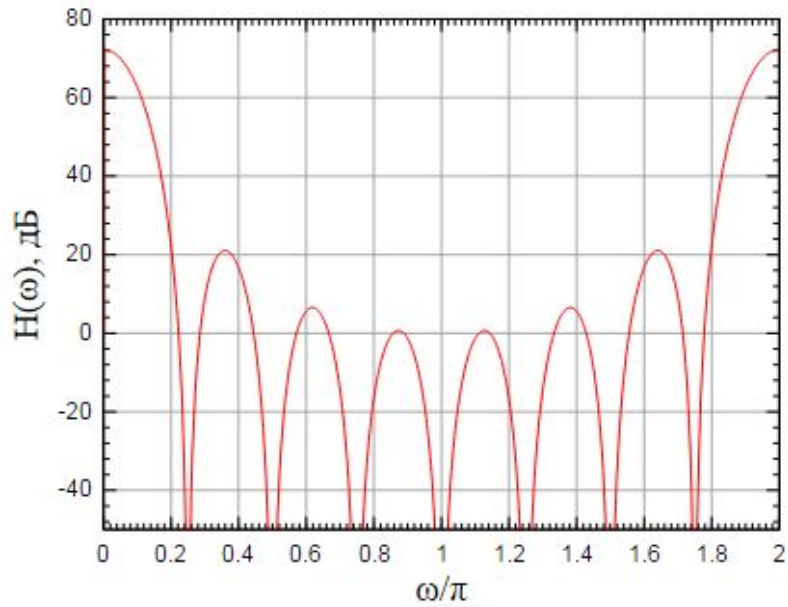
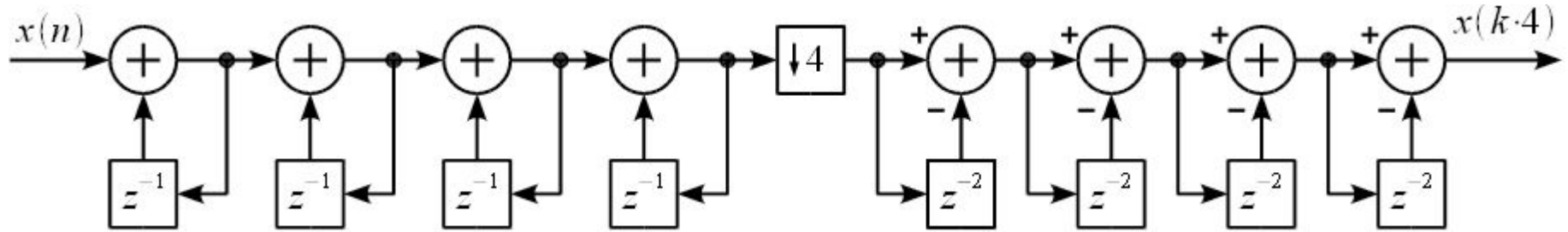
Для вычислений в фильтре используются только  $R$ -ые отсчёты входного сигнала, задержанные на  $D$ -тактов =>



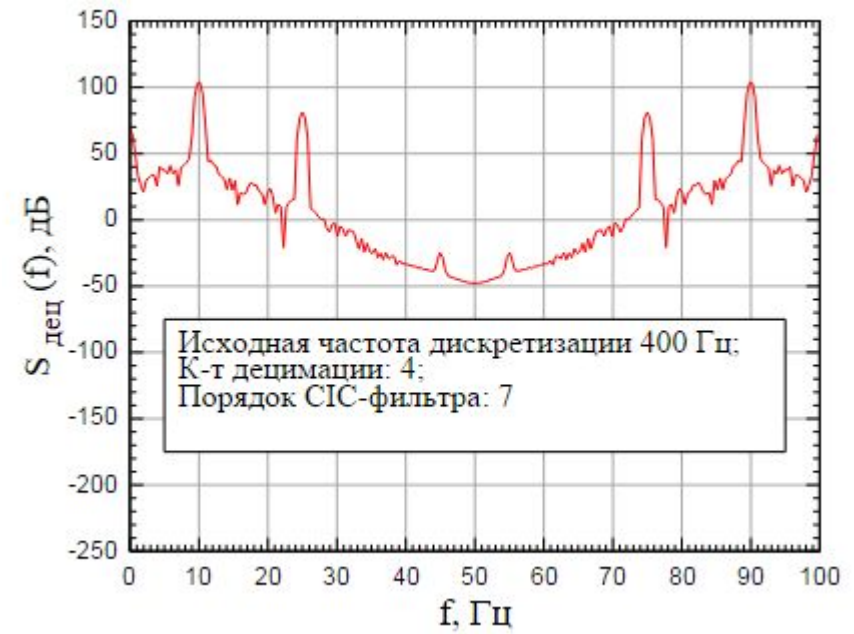
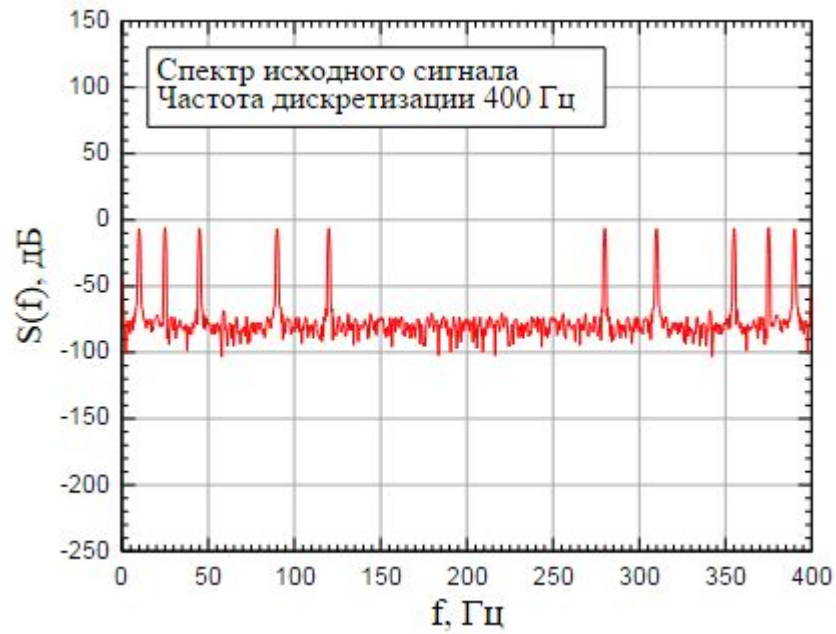
$$x_D(k) = x_R(k) - x_R(k-2).$$

$$x(k \cdot R) = x_R(k) \Rightarrow x_D(k) = x_R(k) - x_R(k - D/R),$$

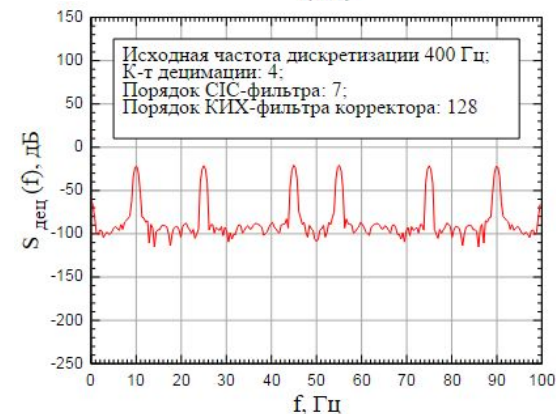
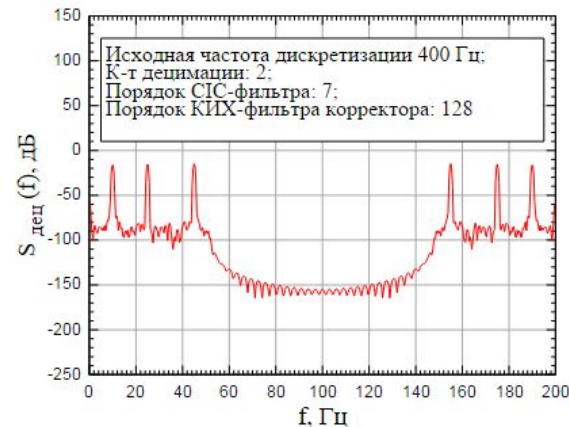
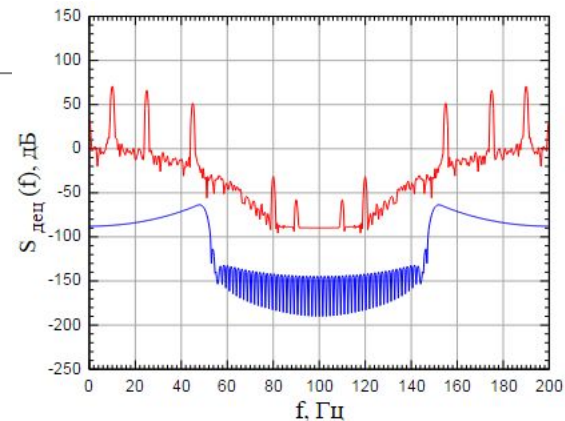
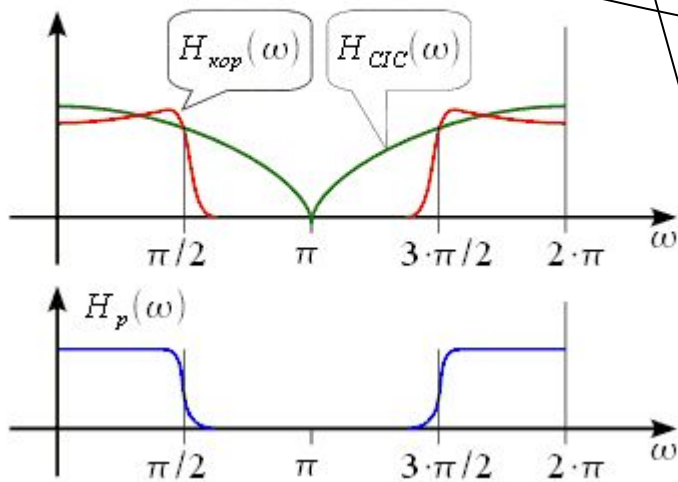
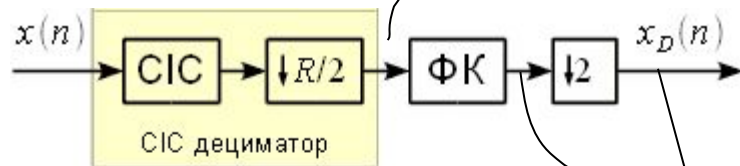
# СИС фильтры в задачах децимации



# СІС фильтры в задачах децимации

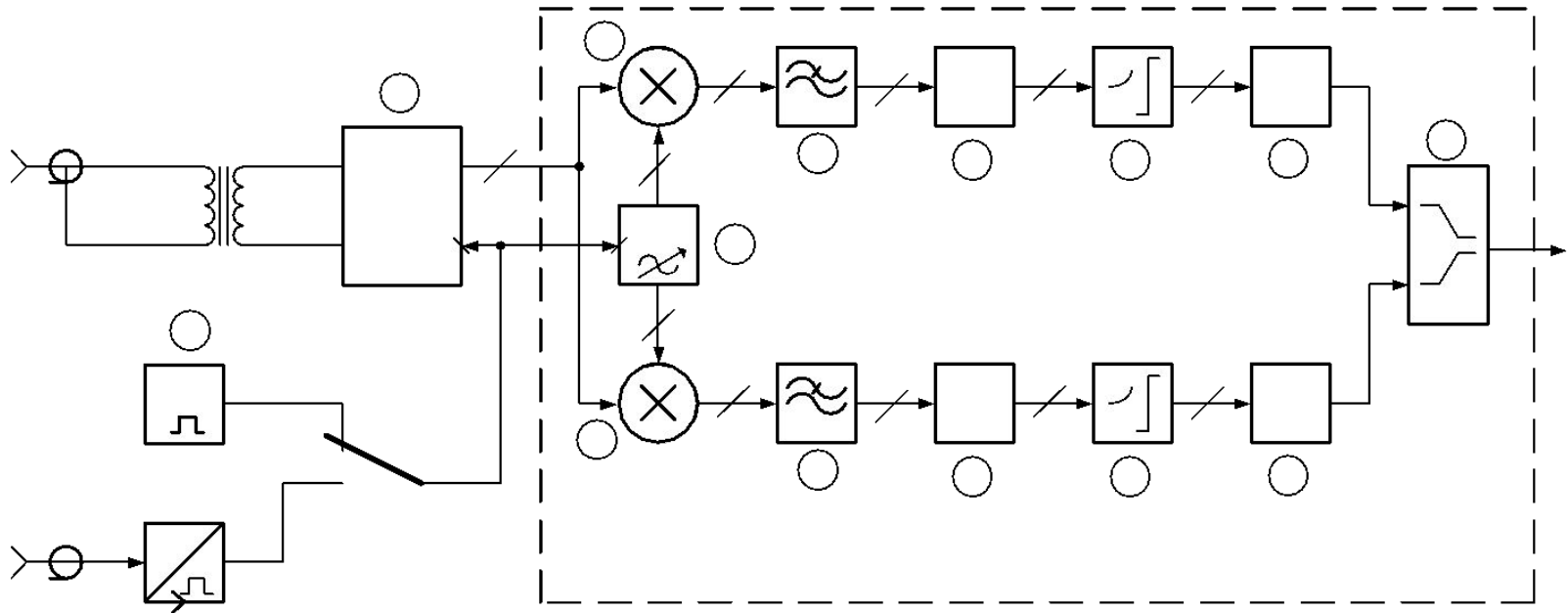


# Коррекция АЧХ СИС фильтров



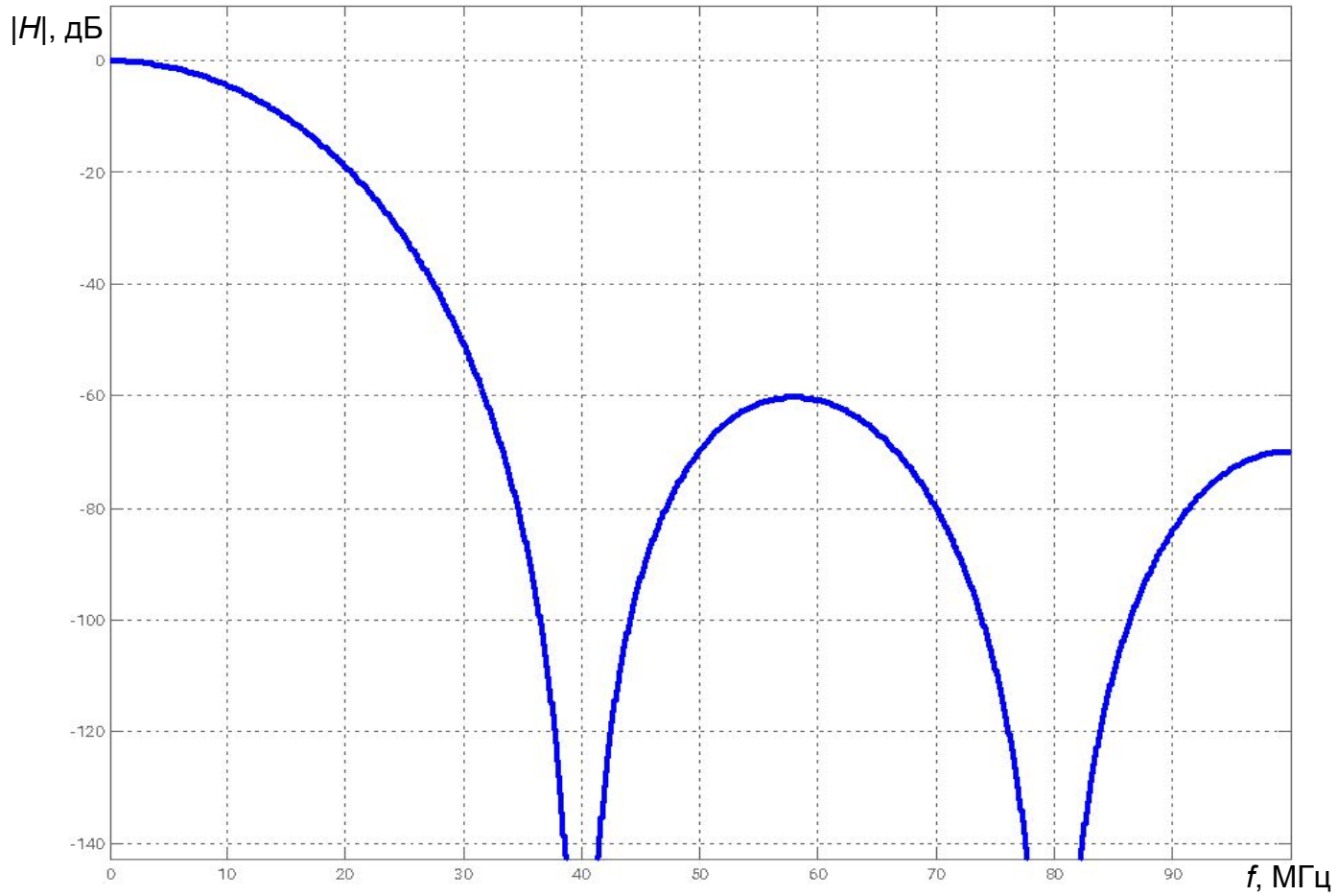
# Пример цифрового приемника на основе СЧФ

## фильтра



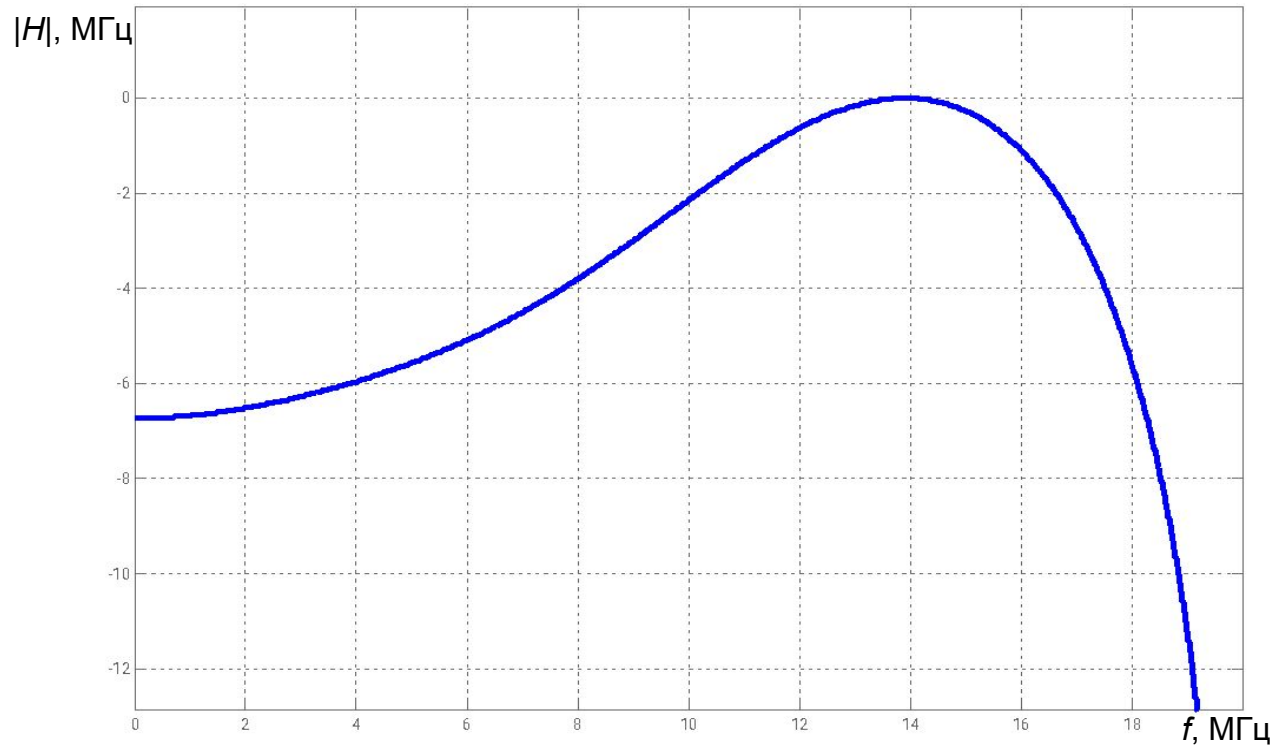
# Пример цифрового приемника на основе СЧФ фильтра

АЧХ СЧФ N=5



# Пример цифрового приемника на основе СЧФ фильтра

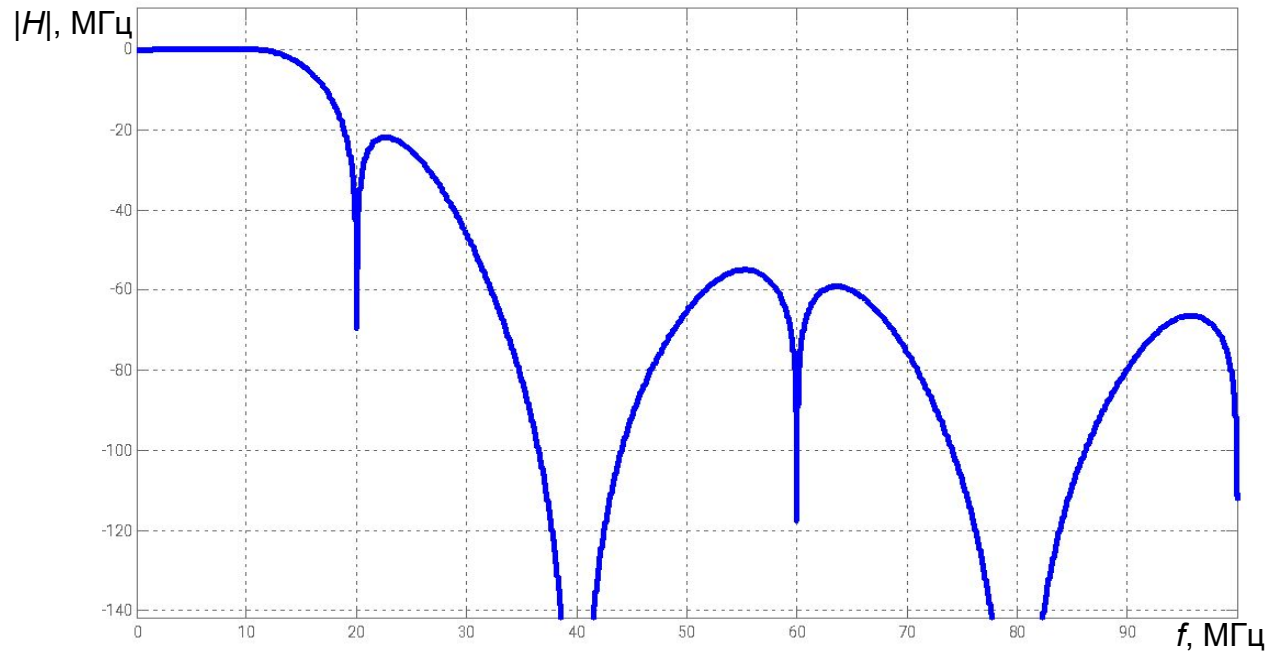
АЧХ корректирующего фильтра



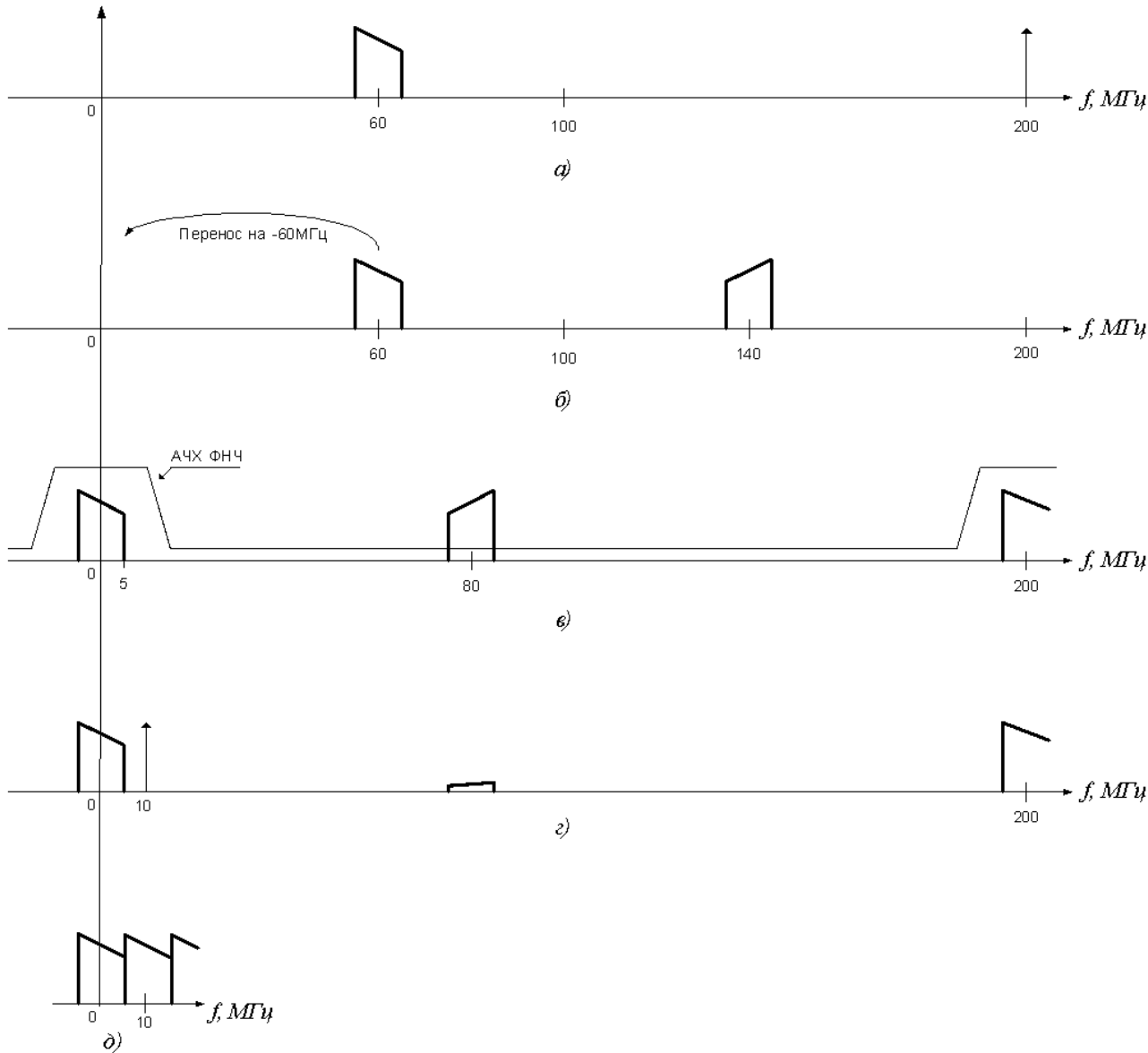


# Пример цифрового пресмника на основе СЧФ фильтра

Результирующая АЧХ фильтра



# Пример цифровой приемника на основе СЧФ фильтра



# Полуполосные фильтры

---

half band filters – экономичные фильтры, которые широко применяются при многоскоростной обработке сигналов, могут быть реализованы без умножителей

- КИХ фильтр
- чётное число коэффициентов
- 3 ненулевых коэффициента в центре ИХ
- от центра ИХ в оба направления ненулевые коэффициенты чередуются с нулевыми значениями
- общий вид ИХ от центра к краям – sinc функция
- образуется из КИХ фильтра от с линейной ФЧХ типа 2 (N четное) с добавлением чередующихся нулевых отсчетов и вставкой центрального коэффициента значением 0,5

# Полуполосные фильтры

half band filters

