3. Цифровые системы частотной селекции на основе многоскоростной обработки сигналов

3.1. Цифровые многоскоростные системы анализа-синтеза сигналов на основе эффекта прореживания по времени

3.1.1. Общая структура системы анализа-синтеза сигналов

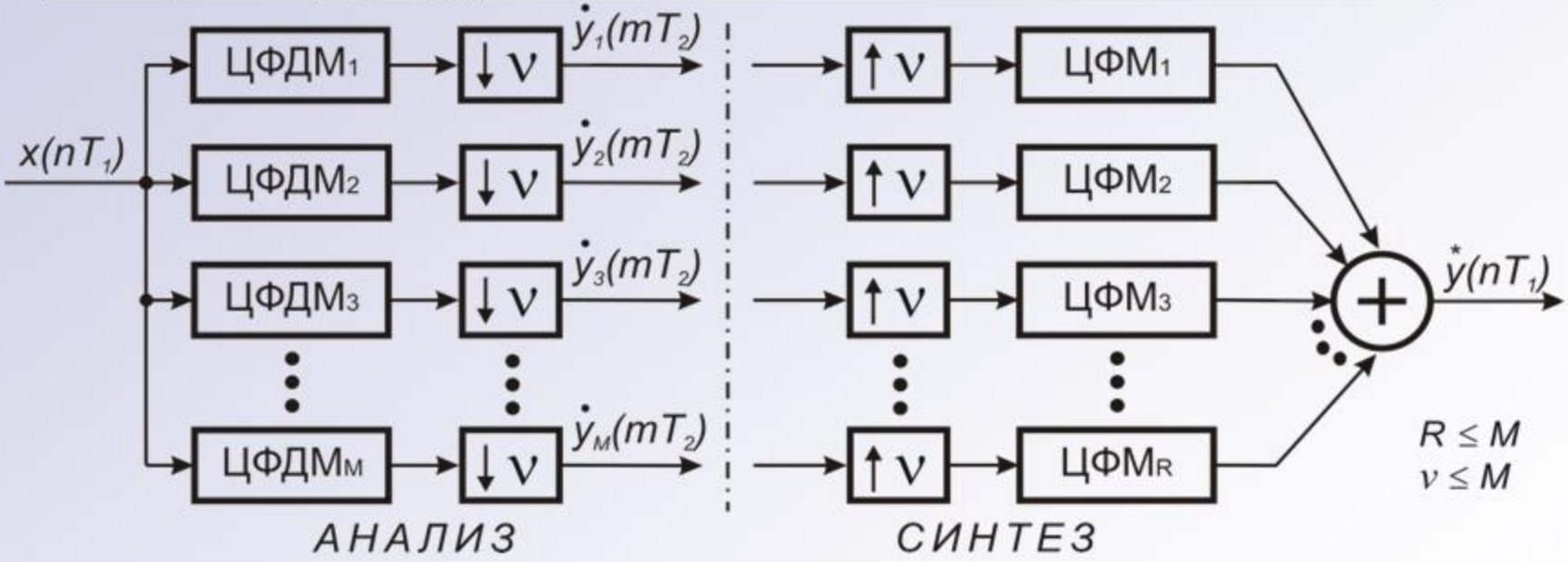


Рис.3.1. Система анализа-синтеза сигналов

Рис.3.2. Два способа структурной реализации ЦФДМ

3.1.2. Классификация методов синтеза набора ЦФДМ

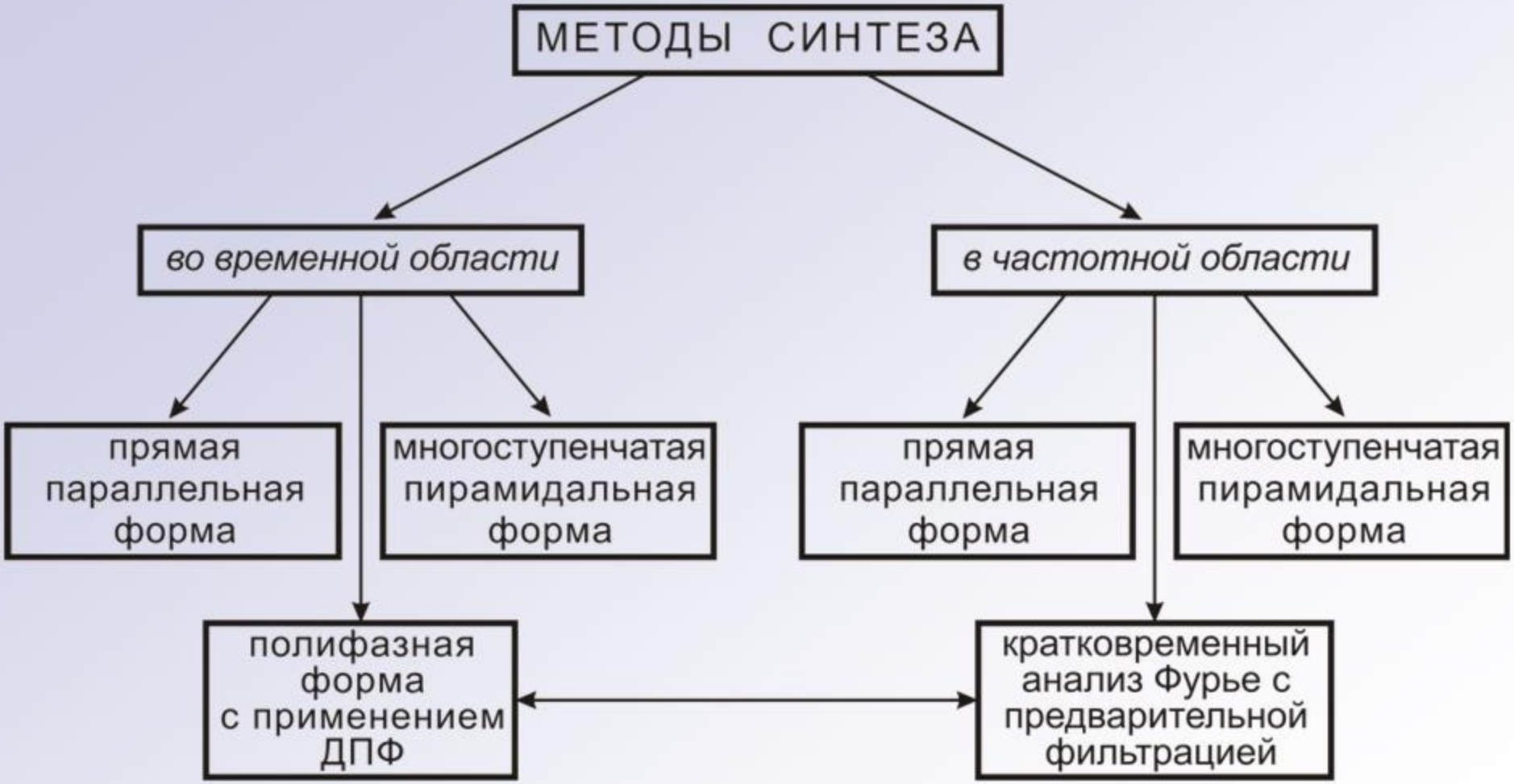


Рис.3.3. Классификация методов синтеза

3.1.3. Методы синтеза структуры набора ЦФДМ во временной области

1. *Прямая параллельная форма*
2. *Многоступенчатая пирамидальная форма*

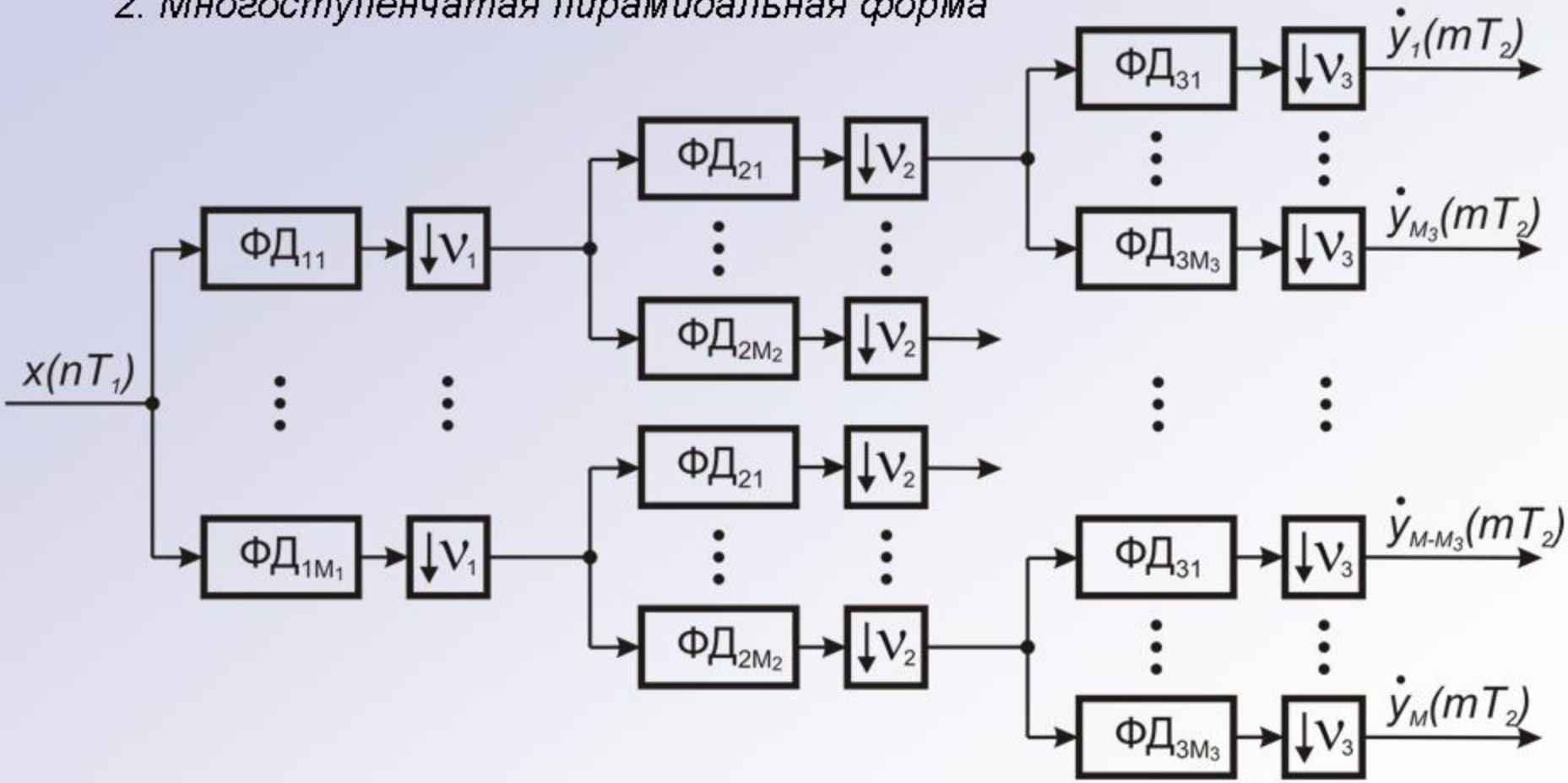


Рис.3.4. Пирамидальная трехступенчатая форма набора ЦФДМ.

3. Полифазная форма набора ЦФДМ с однотипными частотными характеристиками

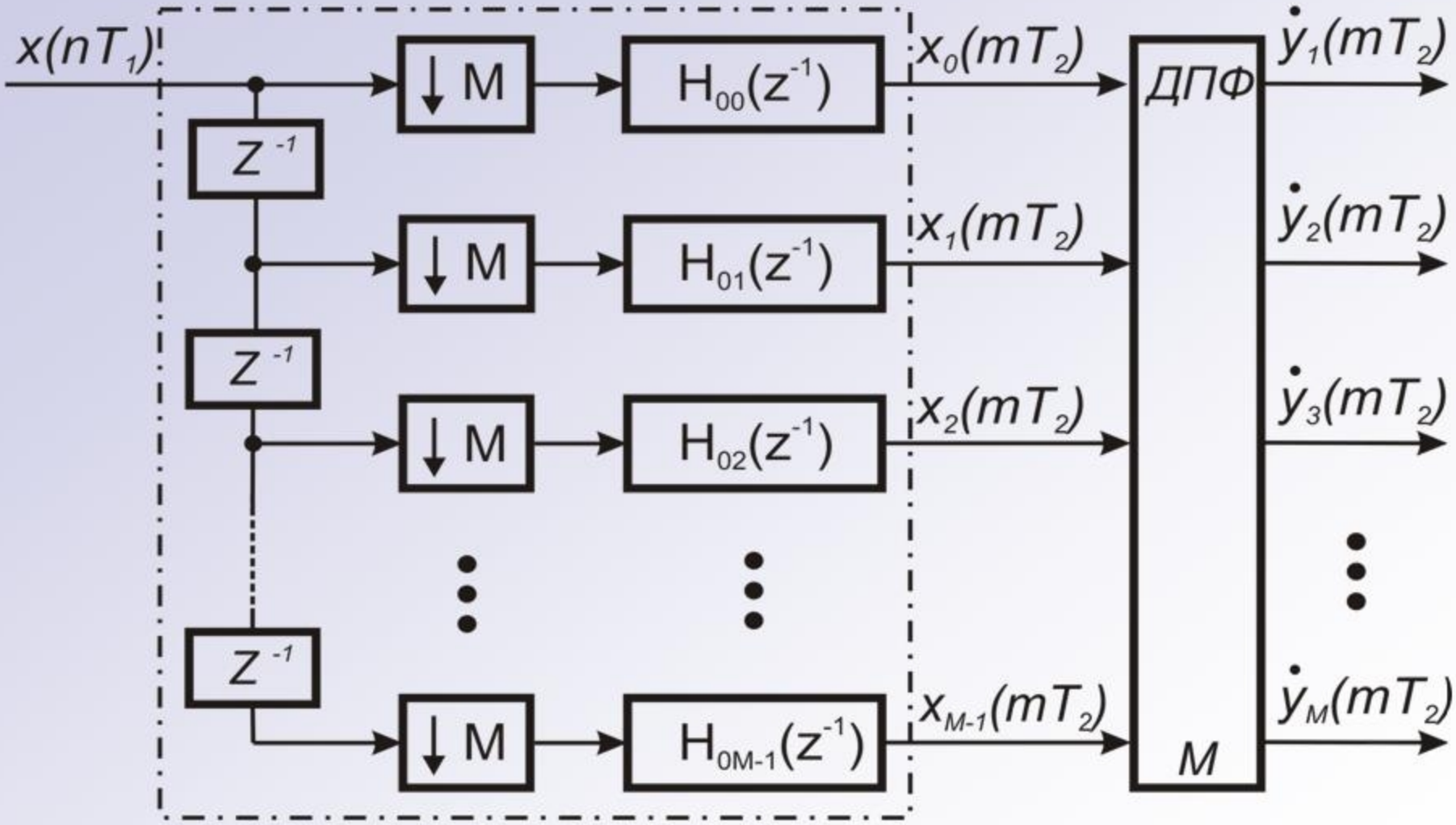


Рис.3.5. Полифазная форма М-канальной системы

3.1.4. Методы синтеза структуры набора ЦФДМ в частотной области

1. *Прямая параллельная форма на основе двойного БПФ с усечением дискретной АЧХ*

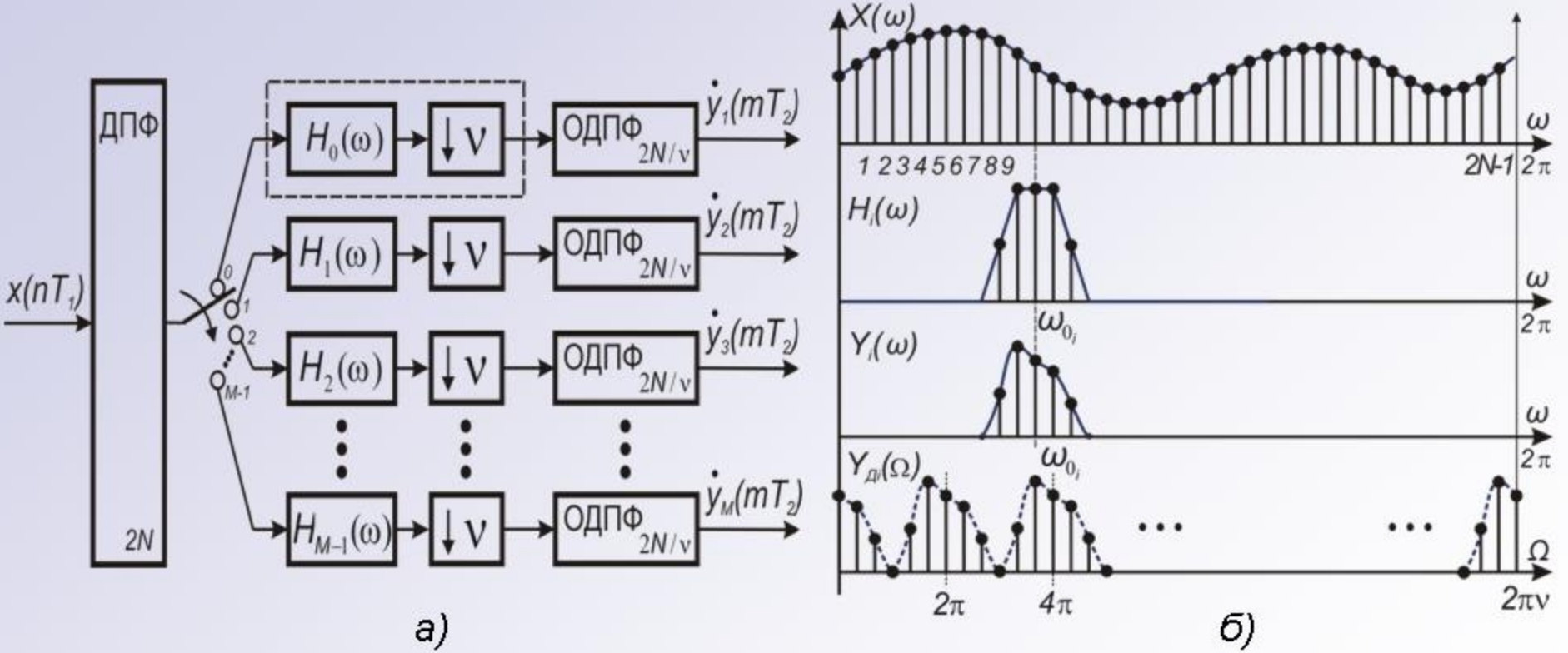


Рис.3.6. Прямая параллельная форма набора ЦФДМ: *а)* структурная схема, *б)* преобразования спектра сигнала i-го частотного канала

3.2. Цифровые системы частотной селекции сигналов на основе эффекта прореживания по частоте

3.2.1. Двухступенчатая структура набора ЦПФ

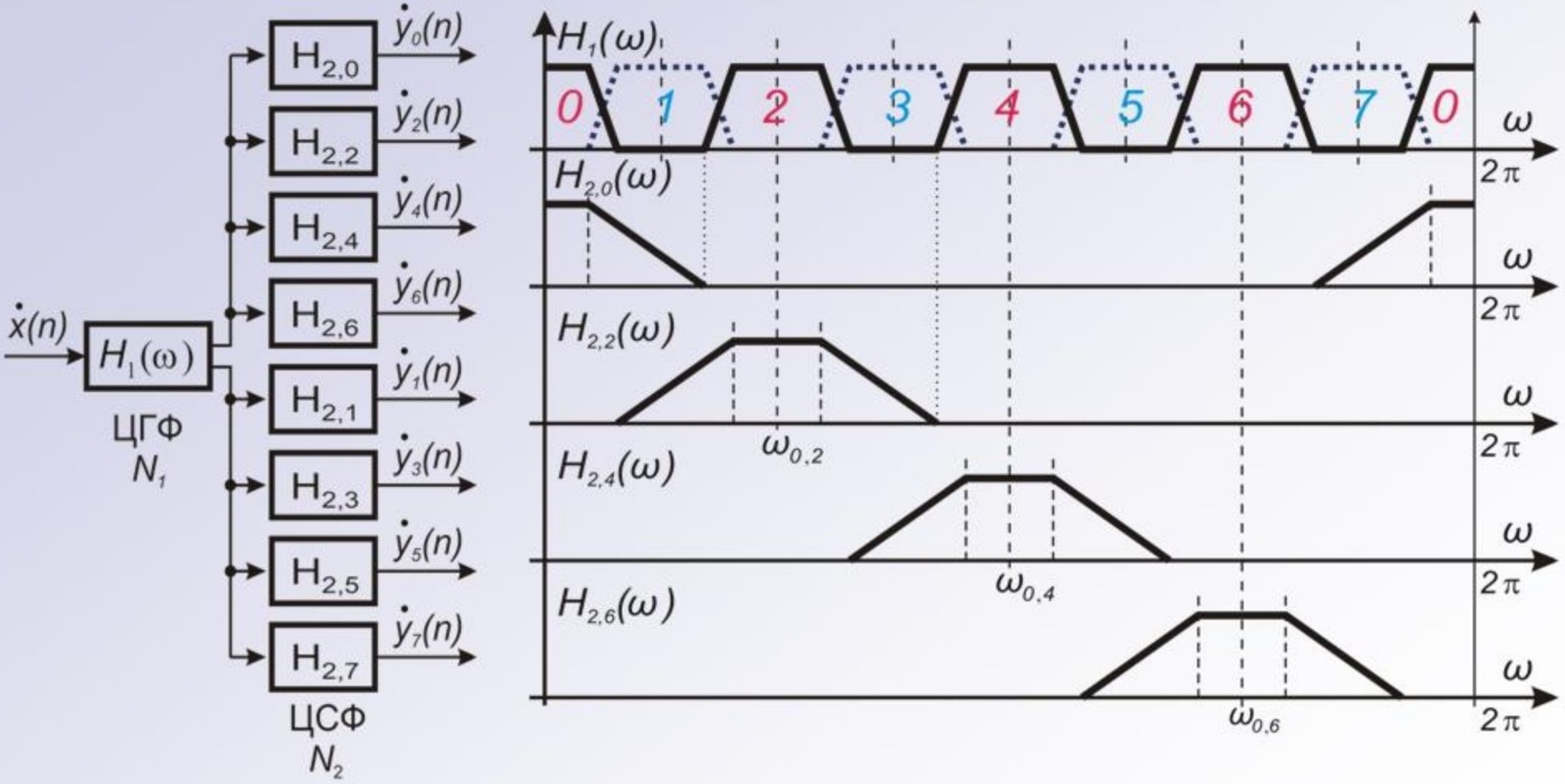
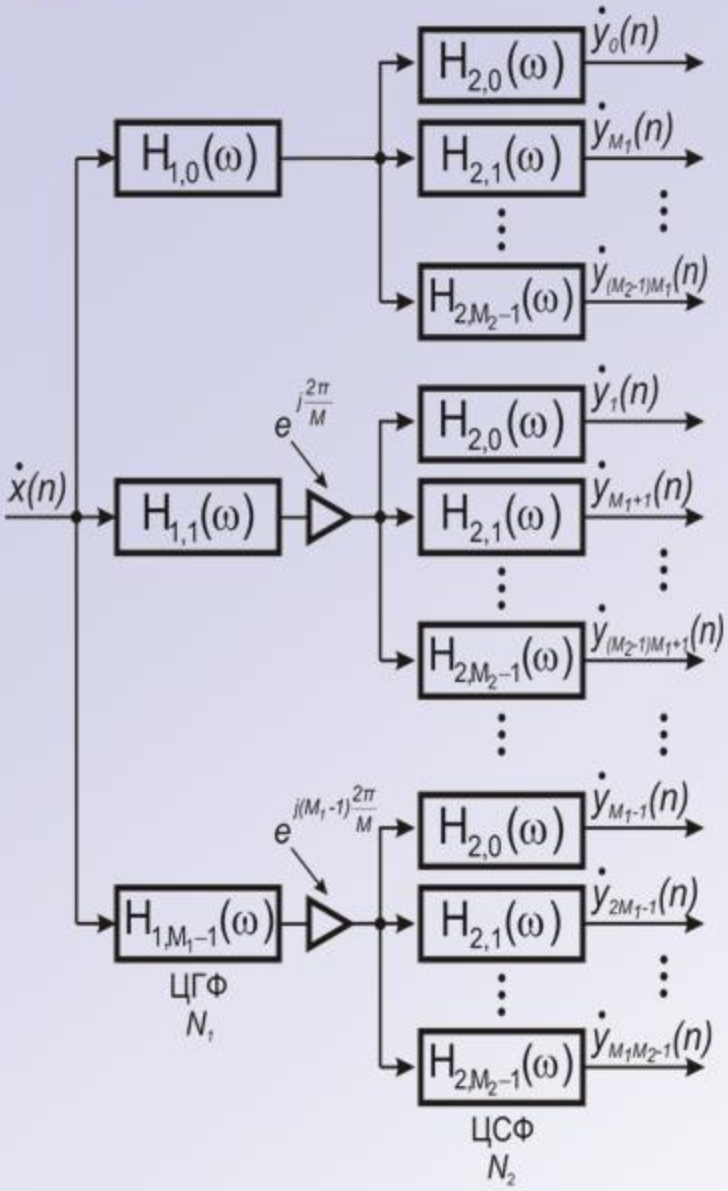
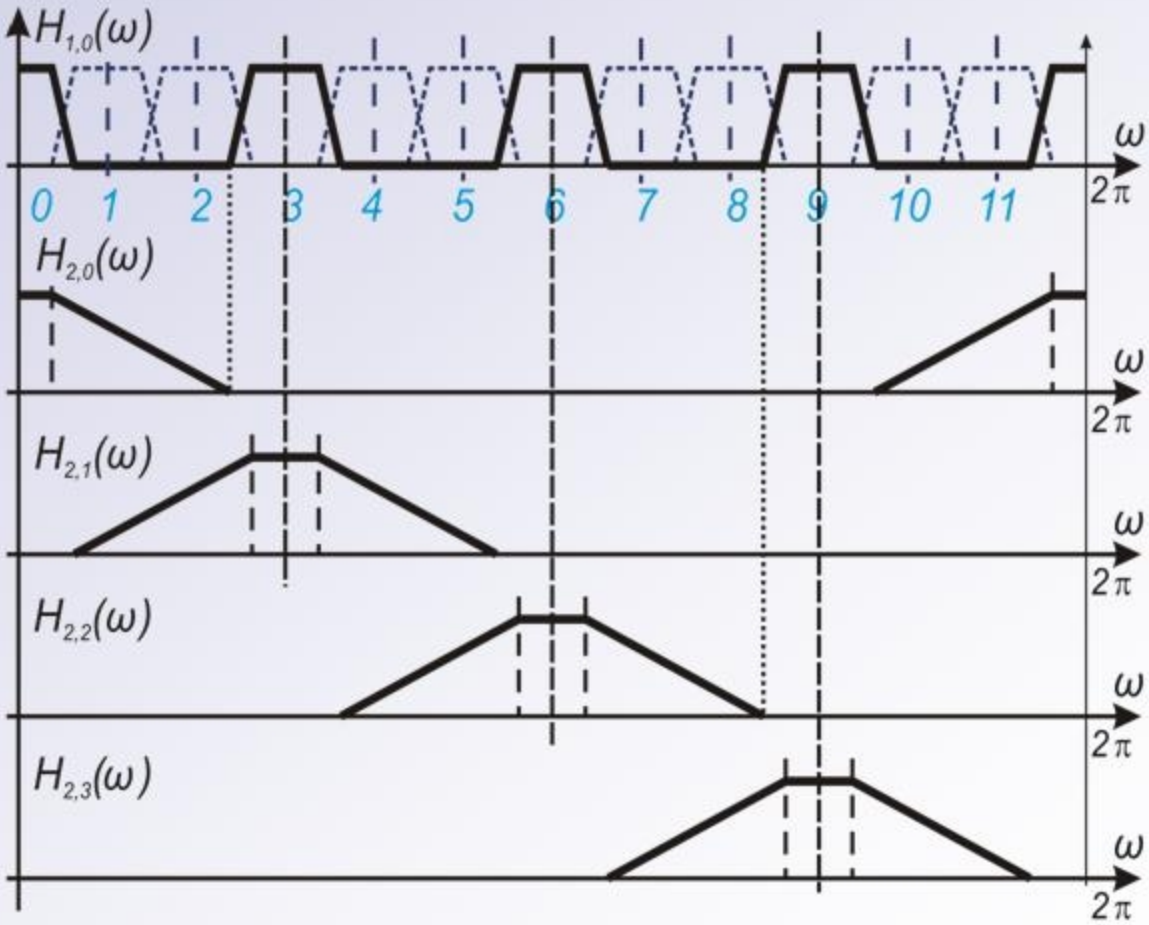


Рис.3.7. Двухступенчатая реализация набора цифровых полосовых фильтров на основе формирующего АЧХ гребенчатого фильтра





$$M = M_1 \cdot M_2 = 3 \cdot 4 = 12$$

Рис.3.8. Оптимальная двухступенчатая структура набора ЦПФ

3.2.2. Пирамидальная многоступенчатая структура набора ЦПФ на основе полуполосных гребенчатых фильтров

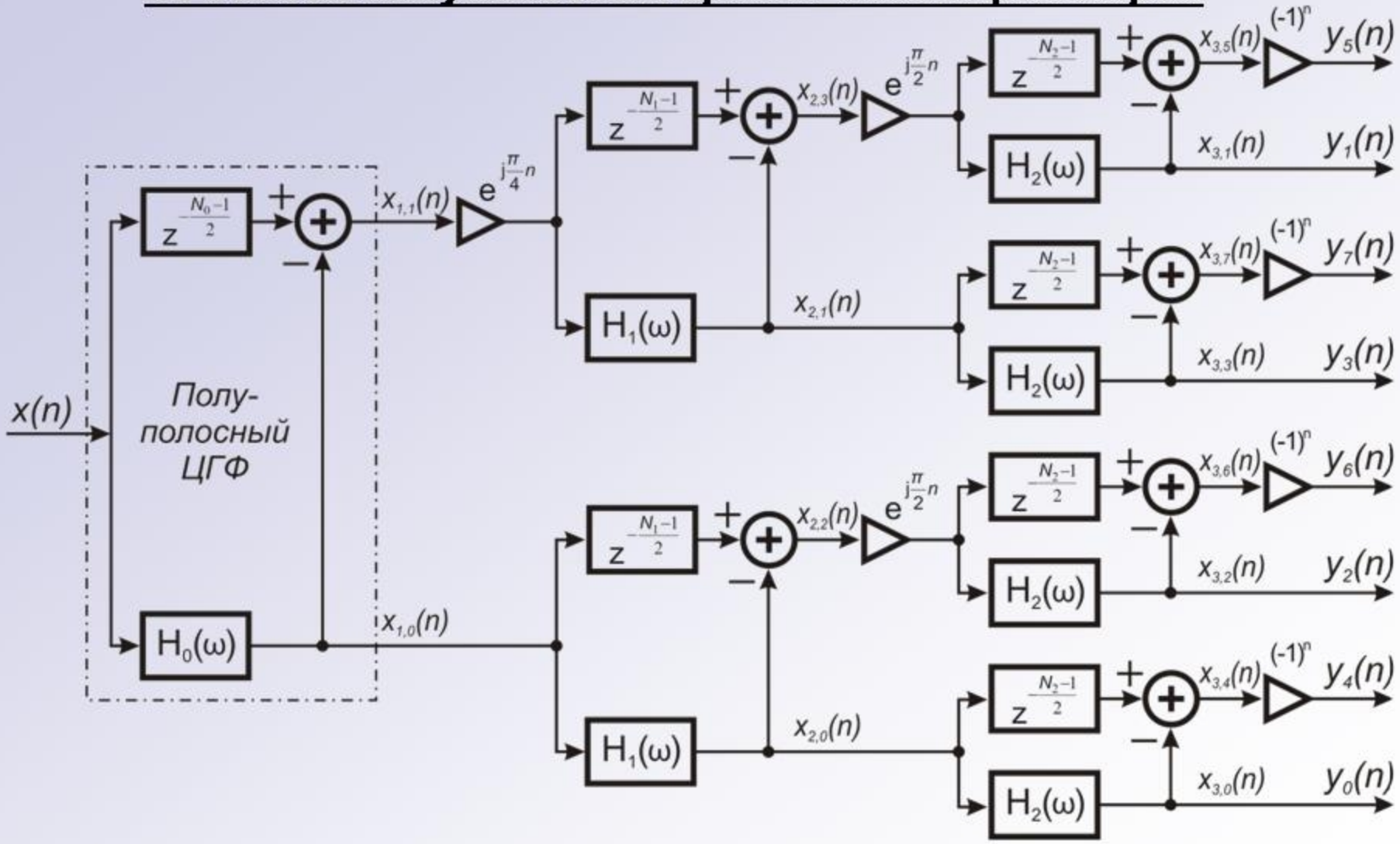


Рис.3.9. Трехступенчатая пирамидальная структура 8-канальной системы частотной селекции сигналов

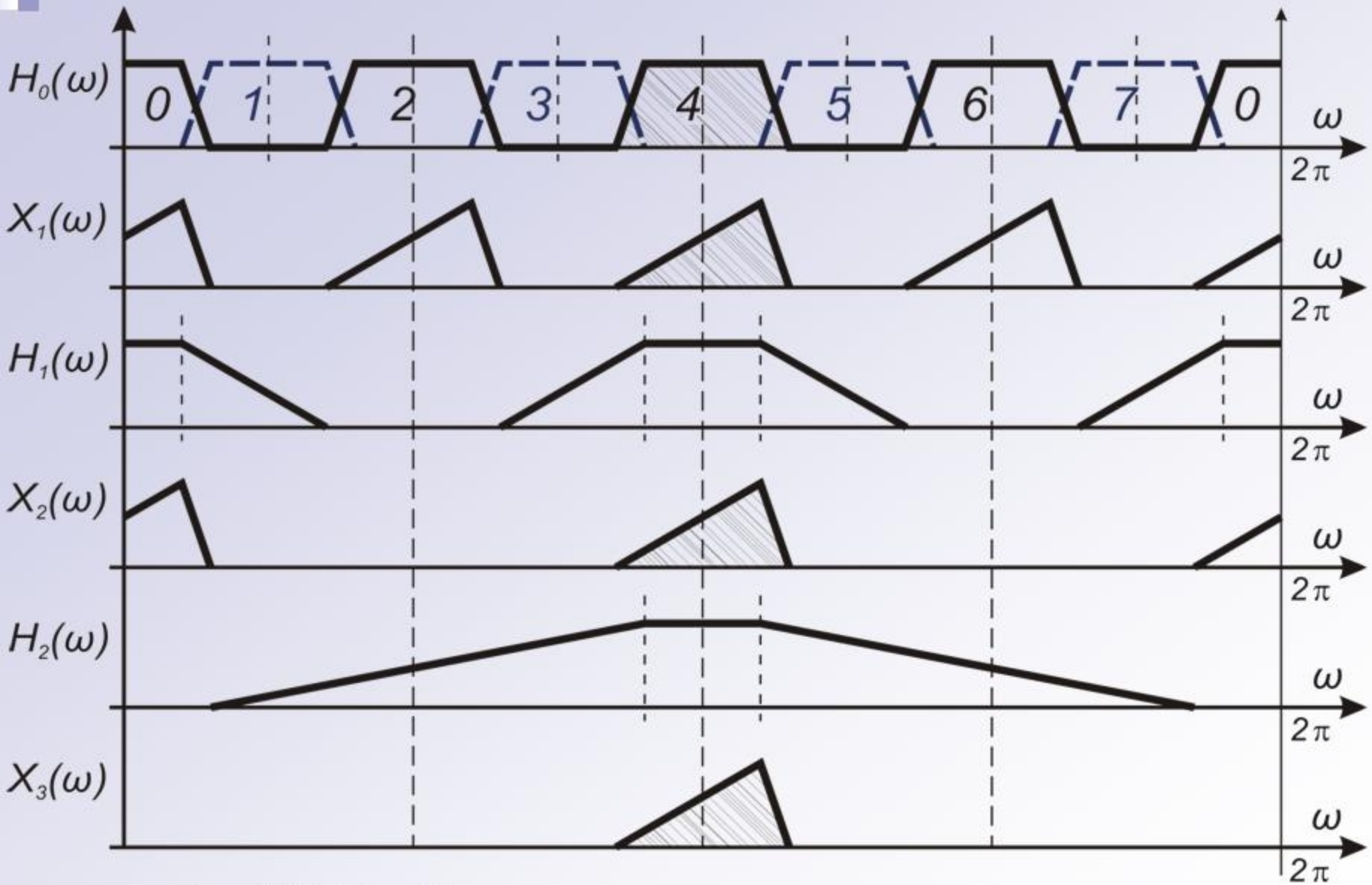


Рис.3.10. Преобразования спектра сигнала при формировании выхода $y_4(n)$ пятого частотного канала

Пирамидальная структура набора цифровых полосовых фильтров: оценка вычислительной эффективности

$V(M) = V_T(M) + V_\Phi(M)$ - число *операций умножения за период дискретизации T.*

$V_T(M) = 2M(1 + \frac{1}{2} + \frac{1}{4} + ... + \frac{2}{M}) \approx 4M$ - *число операций умножения за период дискретизации T.*

$V_\Phi(M) = \sum_{i=0}^{m-1} 2^i \frac{N_i}{\nu_i}$ - *число операций умножения на фильтрацию.*

$S(M) = 2\sum_{i=0}^{m-1} 2^i N_i$ - *число ячеек памяти данных.*

С учетом того, что на последних ступенях трансформирующие множители принимают значения: $(-1)^n$ *и* $\{(1+j0), (0+j1), (-1+j0), (0-j1)\}$ $V_T(M) \approx M$,

а $\frac{N_i}{\nu_i} = \frac{2N_0}{M}; (\nu_0 = \frac{M}{2})$, *то* $V_\Phi(M) = \sum_{i=0}^{m-1} 2^i \frac{2N_0}{M} = 2N_0(\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + ...) \approx 2N_0$.

Таким образом, $V(M) \approx M + 2N_0; S(M) \approx 2N_0 \log_2 M$

4. Адаптивная обработка сигналов и ее применение

4.1. Адаптивные фильтры: назначение, классификация и применение

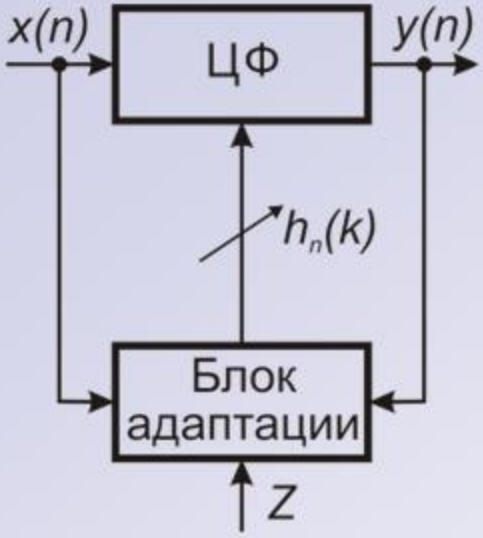
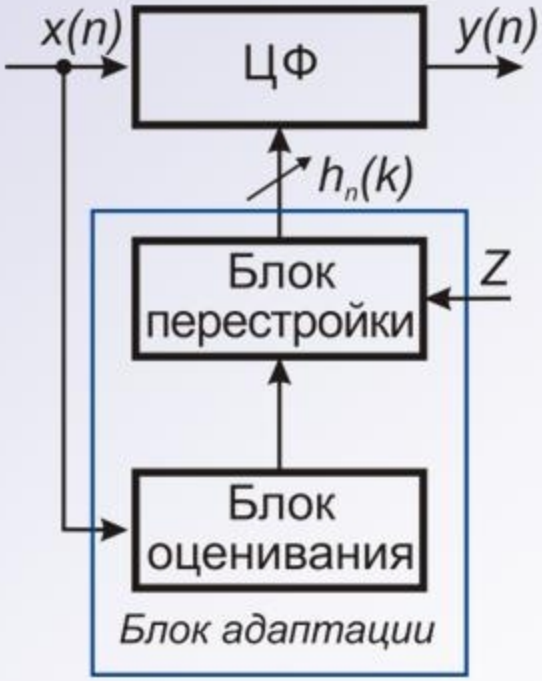


Рис.4.1. Общая структура адаптивного фильтра

Классификация АФ

1.Без обратной связи
(без предварительного обучения)

2.С обратной связью
(с предварительным обучением)



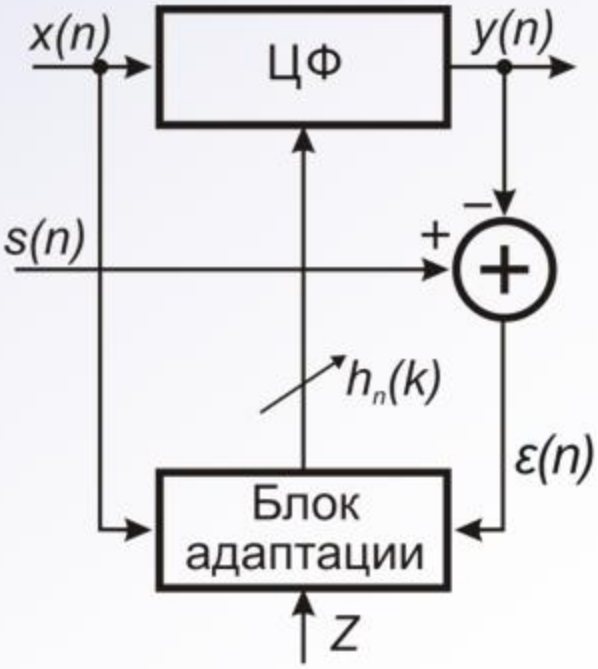
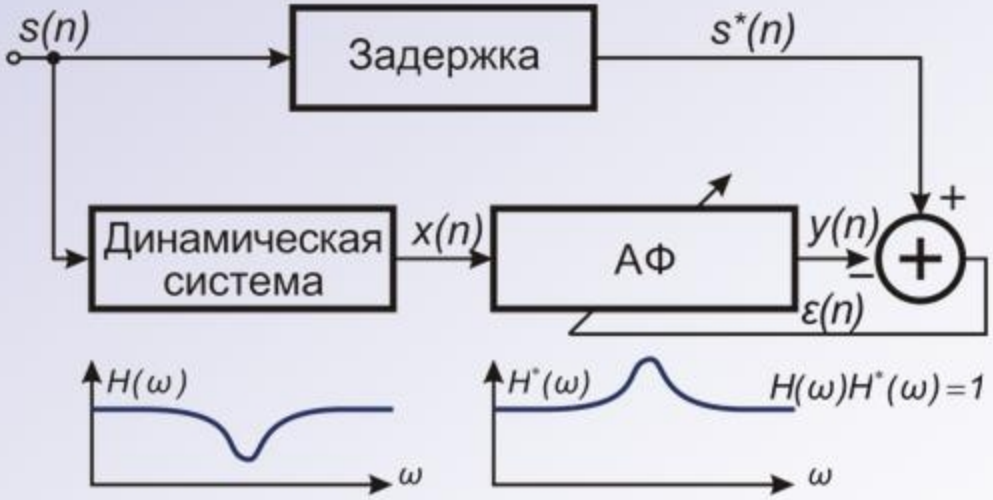


Рис.4.2. Два варианта построения АФ

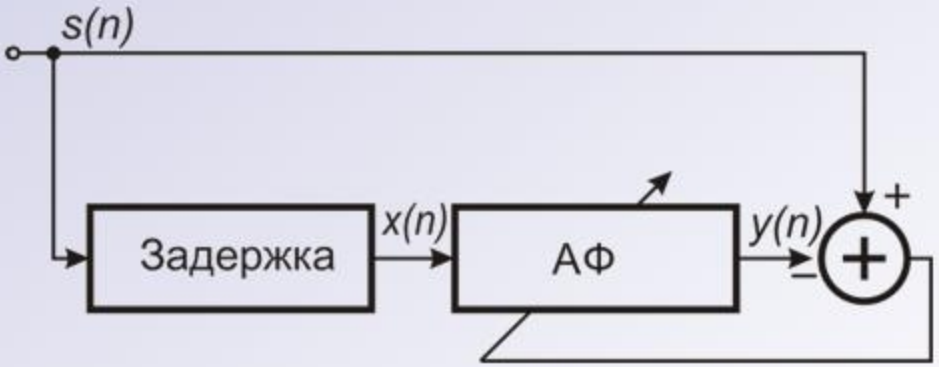
1. Прямое моделирование

$\varepsilon(n) = s(n) - y(n)$ - ошибка обучения

2. Обратное моделирование



3. Устройство предсказания



4. Компенсатор помех

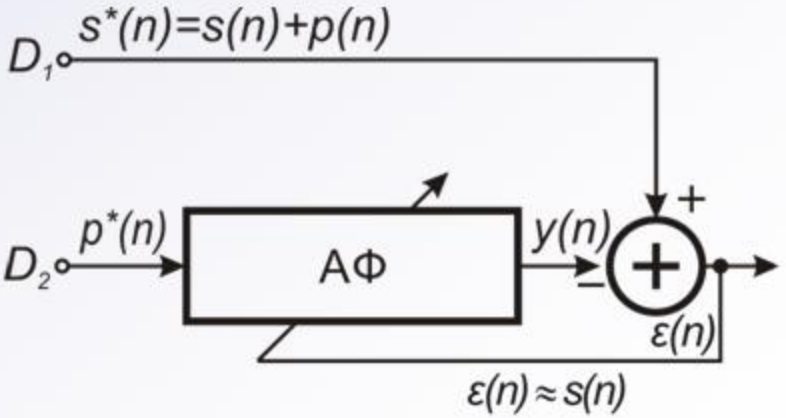


Рис.4.3. Применение АФ с обратной связью (обучением)

4.2. Адаптивные КИХ-фильтры: общее описание и методы синтеза

$$y(n) = \sum_{k=0}^{N-1} x(n-k) h_n(k) \qquad (4.1)$$

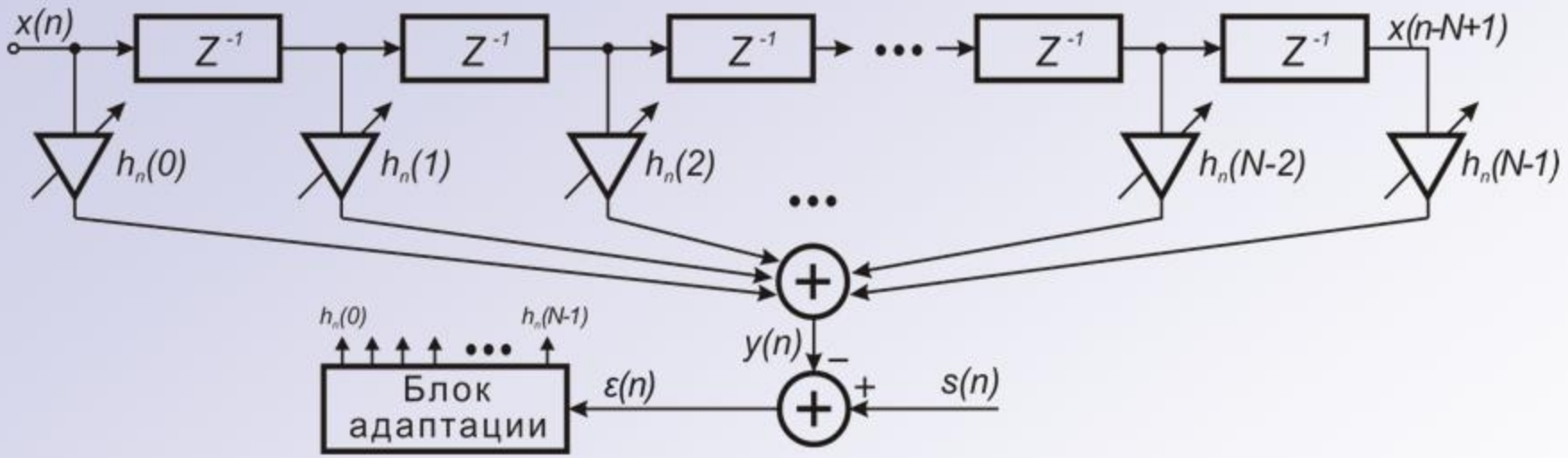


Рис.4.4. Прямая форма реализации адаптивного КИХ-фильтра

Пусть $y(n) = X^T(n)\, H(n) = H^T(n)\, X(n),$ (4.2)

где $X^T(n) = \{x(n),\ x(n-1),\ x(n-2),\ \ldots\ x(n-N+1)\}$

$H^T(n) = \{h_n(0),\ h_n(1),\ h_n(2),\ \ldots\ h_n(N-1)\}$

Тогда $\varepsilon(n) = s(n) - y(n) = s(n) - X^T(n)\, H(n)$ и

$\varepsilon^2(n) = s^2(n) + H^T(n)\, X(n)\, X^T(n)\, H(n) - 2\, s(n)\, X^T(n)\, H(n)$

Пусть $\varepsilon(n)$, $s(n)$ и $X(n)$ – стационарные эргодические случайные процессы.

Тогда *СКО* при заданном векторе весовых коэффициентов H

$$\delta(H) = E[\varepsilon^2(n)] = E[s^2(n)] + H^T E[X(n)X^T(n)]\, H - 2\, E[s(n)X^T(n)]\, H =$$
$$= E[s^2(n)] + H^T R\, H - 2\, P^T H, \qquad (4.3)$$

где $R = E[X(n)\, X^T(n)]$ – корреляционная матрица входного сигнала

$P^T = E[s(n)\, X^T(n)]$ – транспонированный вектор взаимной корреляции входного и обучающего сигналов

Глобальный оптимум *СКО* определяется из уравнения

$$\nabla = \frac{\partial \delta(H)}{\partial H} = 2RH - 2P = 0 \qquad (4.4)$$

Отсюда $\boxed{H_{opt} = R^{-1}P}$ (4.5)

При этом $\delta_{min} = E[s^2(n)] + (R^{-1}P)^T R\, R^{-1}P - 2\, P^T R^{-1}P$

Используя свойства: $R R^{-1} = I$; $(R^{-1})^T = R^{-1}$ и $(R^{-1}P)^T = P^T (R^{-1})^T = P^T R^{-1}$,

получим: $\delta_{min} = E[s^2(n)] - P^T R^{-1}P = E[s^2(n)] - P^T H_{opt}$ (4.6)

Альтернативный подход на основе градиентных методов поиска экстремума рабочей функции (СКО)

Предполагается, что *СКО* является квадратичной функцией вида

$$\delta(H) = \delta_{min} + (H - H_{opt})^T R (H - H_{opt}) \qquad (4.7)$$

Действительно, раскрывая в (4.7) скобки и выполнив соответствующие векторно-матричные преобразования, с учетом (4.5) и (4.6), получим

$$\delta(H) = \delta_{min} + H^T R H + H_{opt}^T R H_{opt} - \underline{H^T R H_{opt}} - \underline{H_{opt}^T R H} =$$

$$= E[s^2(n)] - P^T R^{-1} P + H^T R H + (R^{-1}P)^T R R^{-1} P - 2 H^T R R^{-1} P =$$

$$= E[s^2(n)] + H^T R H - 2 H^T P$$

Таким образом, $\delta(H) = E[s^2(n)] + H^T R H - 2 H^T P$,

что полностью соответствует исходной рабочей функции (4.3)

Квадратичную функцию (4.7) можно привести к более простому виду

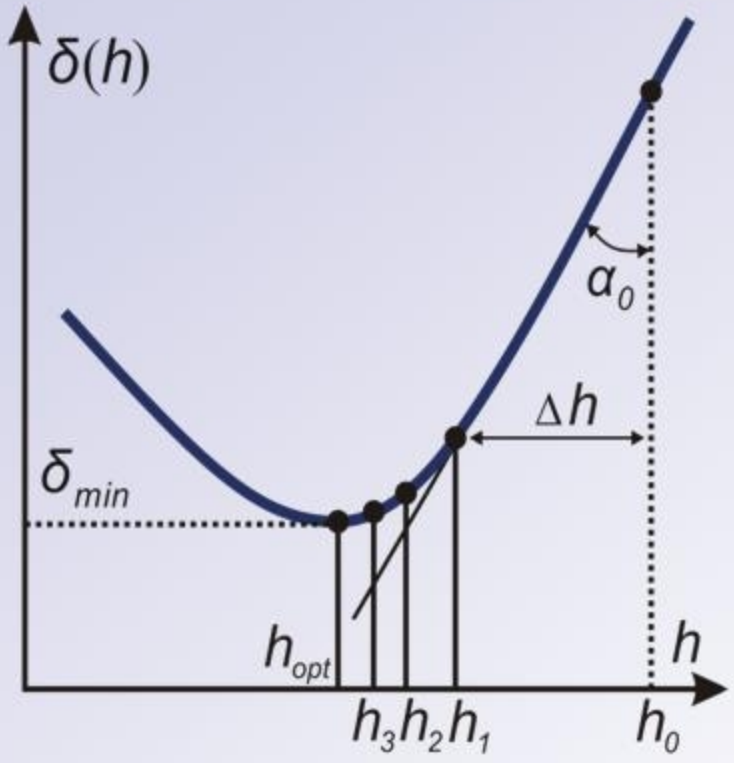
$$\delta(V) = \delta_{min} + V^T R V, \quad \text{где} \quad V = H - H_{opt} \qquad (4.8)$$

При этом оптимум достигается при $V_{opt}=0$.

4.3. Поиск параметров рабочей функции в задачах адаптивной фильтрации

Пусть $\delta(h) = \delta_{min} + \lambda(h - h_{opt})^2$ (4.9)

Градиентные методы поиска экстремума рабочей функции: метод Ньютона и метод наискорейшего спуска



$$h_{k+1} = h_k + \mu(-\nabla_k), \quad (4.10)$$

где ∇_k – градиент рабочей функции,
k – номер итерации, μ – константа

$$\nabla_k = \left.\frac{d\delta(h)}{dh}\right|_{h=h_k} = 2\lambda(h_k - h_{opt}) \quad (4.11)$$

Подставив (4.11) в (4.10), получим

$$h_{k+1} = h_k - 2\mu\lambda(h_k - h_{opt}) = (1 - 2\mu\lambda)h_k + 2\mu\lambda h_{opt} \quad (4.12)$$

$$h_k = h_{opt} + (1 - 2\mu\lambda)^k(h_0 - h_{opt}) \quad (4.13)$$

Устойчивость и скорость сходимости

$r = (1 - 2\mu\lambda)$ - знаменатель геометрической прогрессии (4.13)

Условие устойчивости:

$$|r| = |1 - 2\mu\lambda| < 1 \quad \text{или} \quad 0 < \mu < 1/\lambda \qquad (4.14)$$

При этом $\lim_{k\to\infty}[h_k] = h_{opt}$

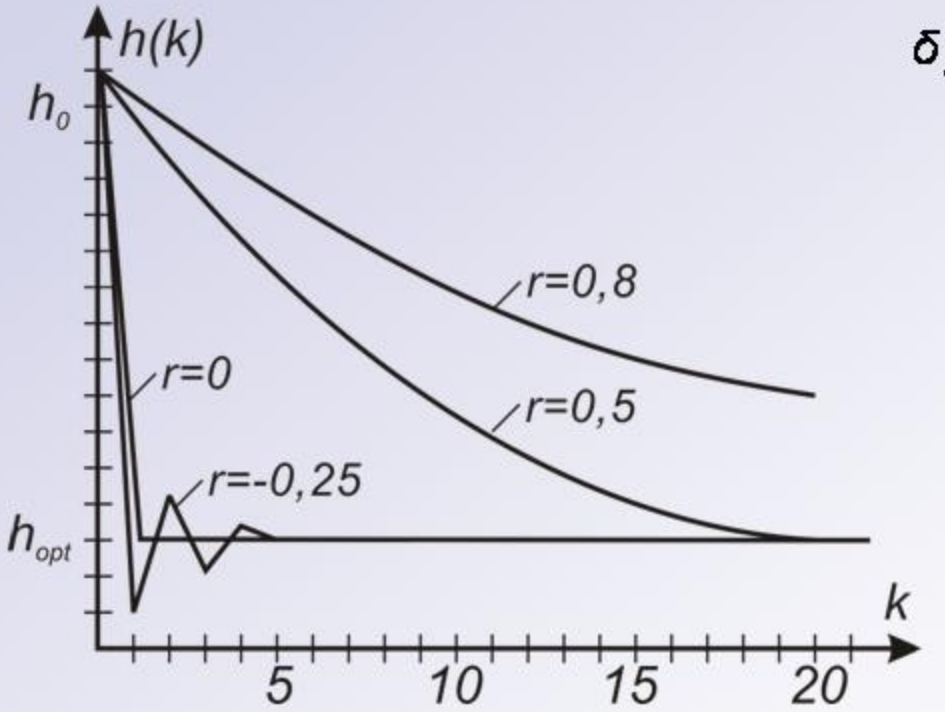


Рис.4.5. Процесс коррекции весовых коэффициентов

Обучающая кривая:

Подставив (4.13) в (4.9), получим

$$\delta_k(h_k) = \delta_{min} + \lambda(1 - 2\mu\lambda)^{2k}(h_0 - h_{opt})^2 \qquad (4.15)$$

$$r_{СКО} = r^2 = (1 - 2\mu\lambda)^2 > 0$$

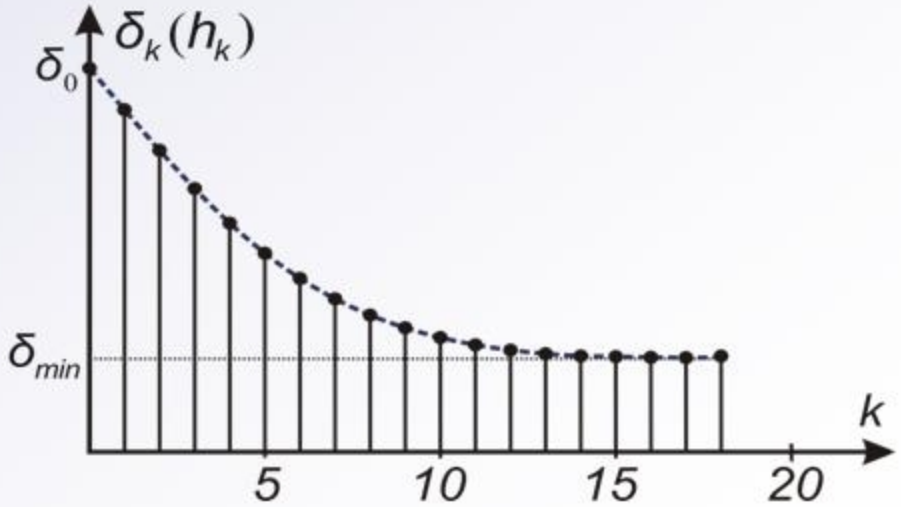


Рис.4.6. Обучающая кривая

Градиентные методы поиска минимума рабочей функции в многомерном пространстве

1. Метод Ньютона

Для квадратичной рабочей функции градиент

$$\nabla = 2RH - 2P \qquad (4.16)$$

Умножив обе части (4.16) слева на ½ R^{-1}, получим

$$\frac{1}{2}R^{-1}\nabla = H - R^{-1}P$$

Отсюда $$H_{opt} = H - \frac{1}{2}R^{-1}\nabla \qquad (4.17)$$

Представим (4.17) в форме пошагового алгоритма

$$H_{k+1} = H_k - \frac{1}{2}R^{-1}\nabla_k$$

или в более общем случае

$$\boxed{H_{k+1} = H_k - \mu R^{-1}\nabla_k} \qquad (4.18)$$

2. Метод наискорейшего спуска

$$\boxed{H_{k+1} = H_k + \mu(-\nabla_k)} \qquad (4.19)$$

$$0 < \mu < 1/\lambda_{max}$$

3. *Метод наименьших квадратов (МНК)*

Предполагает, что

$$\hat{\nabla}_k = -2\varepsilon_k X_k \qquad (4.20)$$

Подставив (4.20) в (4.19), получим

$$H_{k+1} = H_k + 2\mu\varepsilon_k X_k \qquad (4.21)$$

Для ускорения процесса адаптации можно использовать модифицированный МНК на основе метода Ньютона в форме

$$H_{k+1} = H_k + 2\mu\hat{R}^{-1}\varepsilon_k X_k \qquad (4.22)$$

Сравнение градиентных методов:

1. *Метод Ньютона* – самый быстрый, но наиболее затратный в реализации
2. *Метод наискорейшего спуска* – относительно простой в реализации и устойчиво работающий при воздействии шумов
3. *Метод наименьших квадратов (МНК)* – самый простой в реализации, но и самый медленный с позиции скорости сходимости.

4.4. Синтез адаптивных БИХ-фильтров

Пусть передаточная функция БИХ-фильтра принимает вид

$$H(z) = \frac{A(z)}{1 - B(z)}, \qquad (4.23)$$

где $A(z) = a_0 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} + \ldots + a_L z^{-L}$;

$B(z) = b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3} + \ldots + b_M z^{-M}$.

Прямая схема синтеза БИХ-фильтра

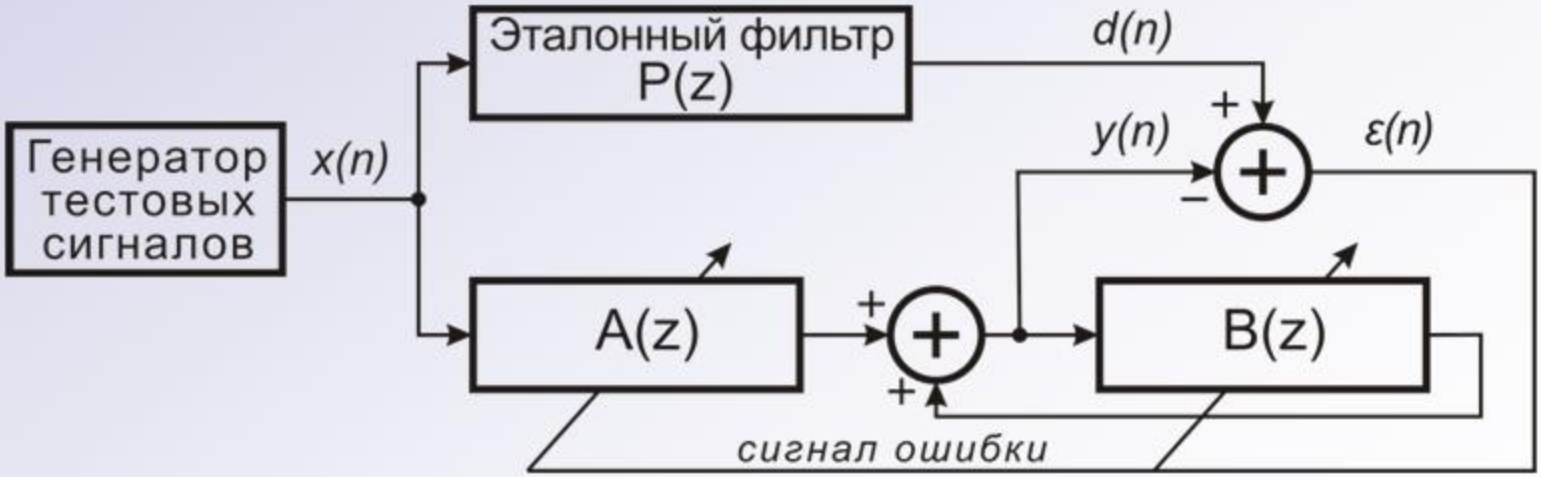


Рис.4.7. Прямая форма реализации адаптивного БИХ-фильтра

Знаменатель передаточной функции (4.23) представим в виде:

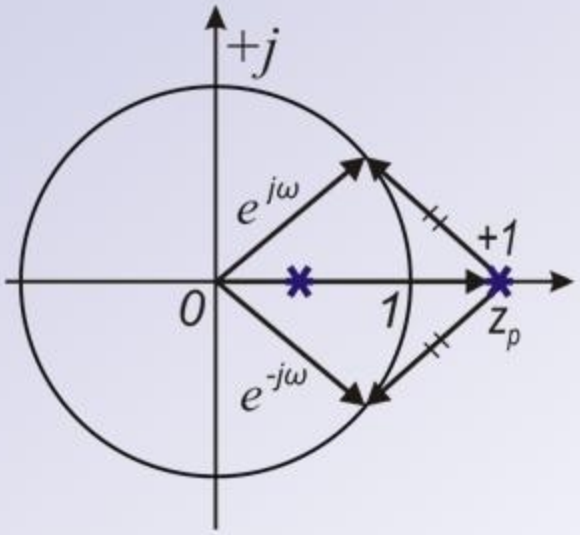
$$1 - B(z) = z^{-M}(z - z_1)(z - z_2)(z - z_3)\ldots(z - z_M) \qquad (4.24)$$

Методика перехода к устойчивому фильтру:

Пусть множитель $(z - z_p)$ в (4.24) отвечает условию $|z_p| > 1$. Тогда в (4.24) выполняется следующая замена: $(z - z_p) \rightarrow (z^{-1} - z_p)$, которая не влияет на АЧХ фильтра, но гарантирует устойчивость.

Пусть z_p – действительная величина, при этом $z_p > 1$.

Очевидно, что $|z - z_p| = |z^{-1} - z_p|$ для всех $z = e^{j\omega}$.



Применение фазового компенсатора

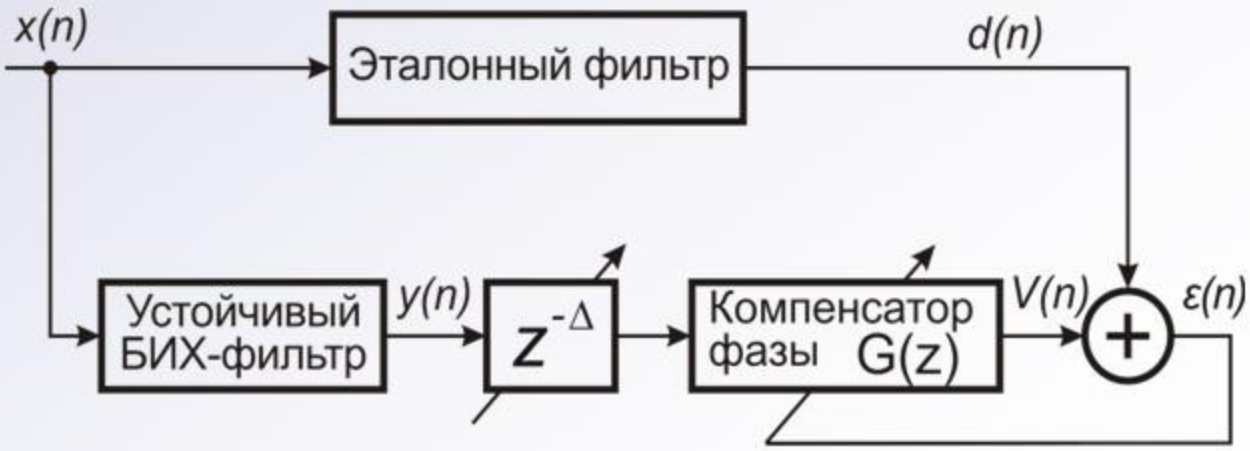


Рис.4.8. Метод фазовой компенсации

Метод прямого и обратного моделирования

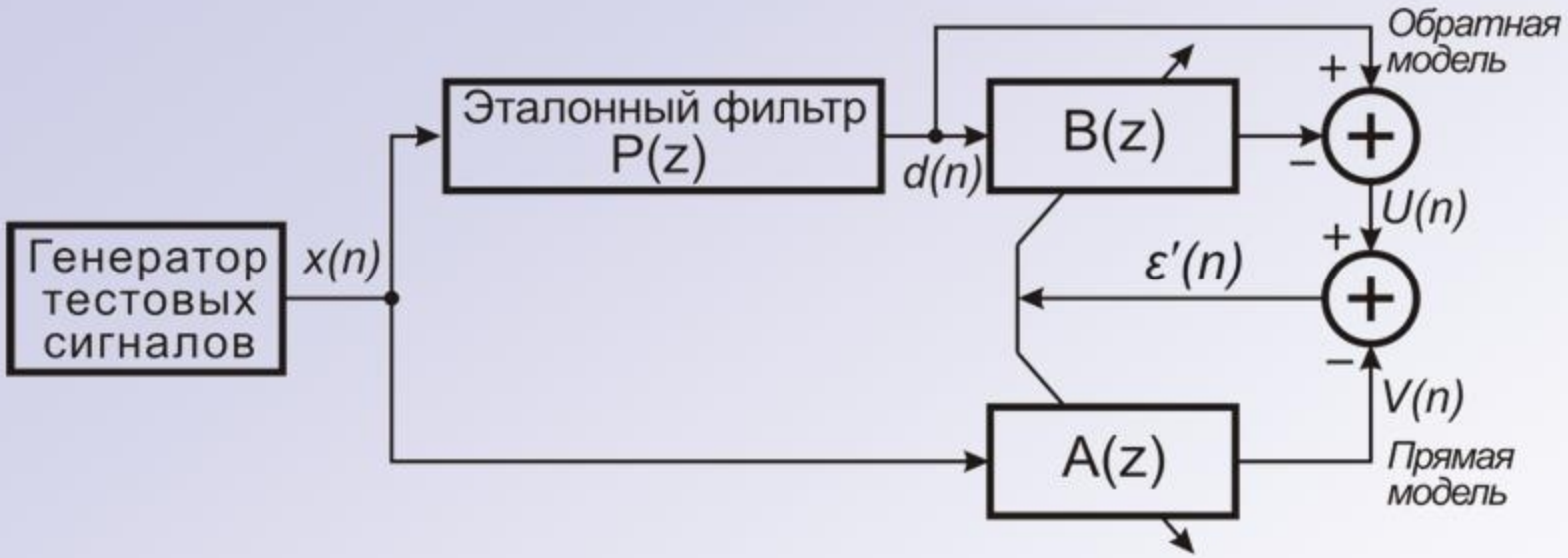


Рис.4.9. Схема построения адаптивного БИХ-фильтра на основе прямого и обратного моделирования

Для *прямой схемы*:

$$E(z) = D(z) - Y(z) = X(z)\left[P(z) - \frac{A(z)}{1 - B(z)}\right]$$

Для *обратной схемы*:

$$E'(z) = U(z) - V(z) = X(z)P(z)\left[1 - B(z)\right] - X(z)A(z) =$$
$$= X(z)\left[P(z)\left[1 - B(z)\right] - A(z)\right] = E(z)\left[1 - B(z)\right]$$

Таким образом,

$$E'(z) = E(z)\left[1 - B(z)\right] \qquad (4.25)$$

4.5. Применение адаптивной обработки сигналов в телекоммуникационных системах

4.5.1. Прямое моделирование многолучевого канала связи

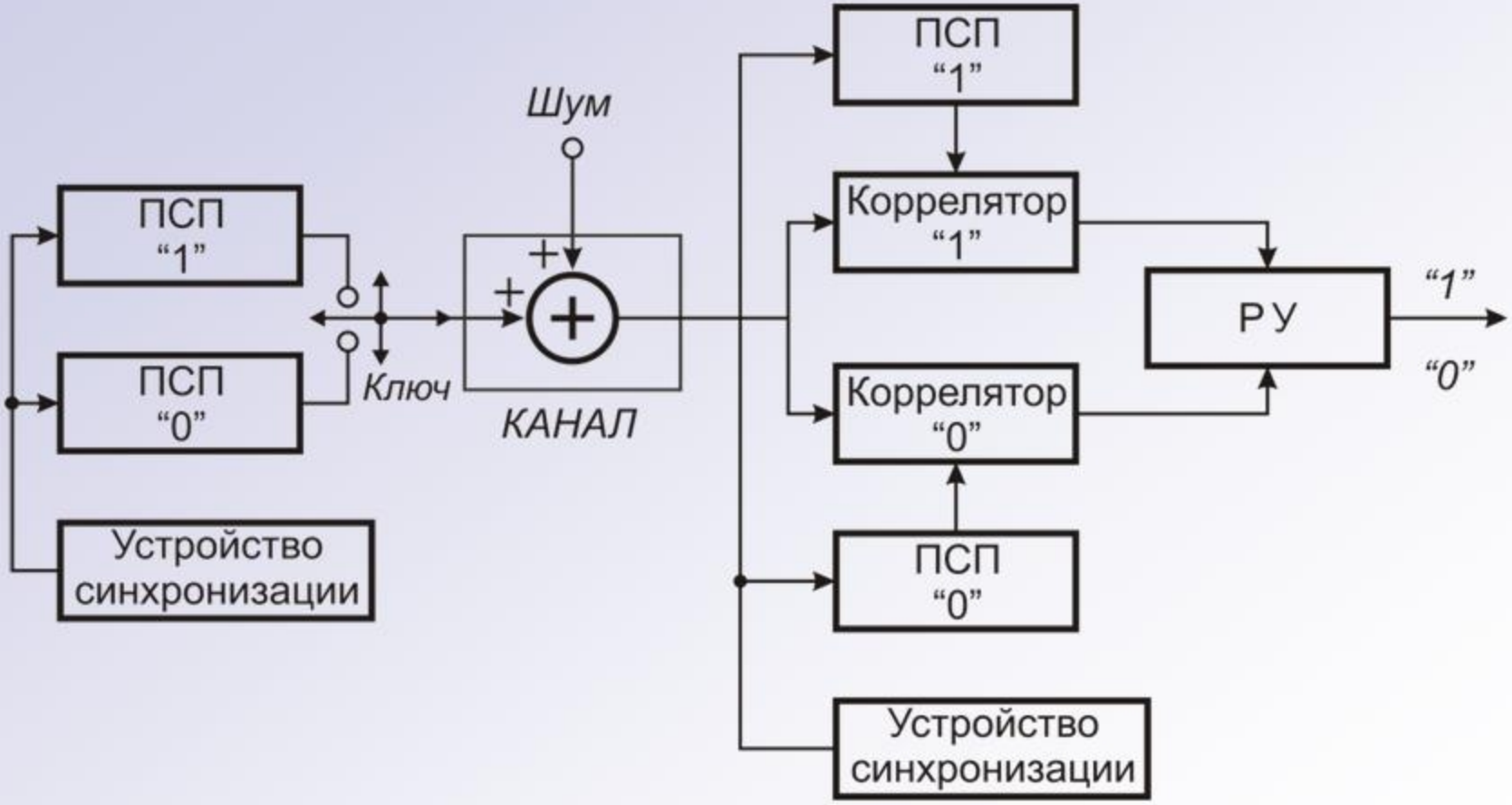
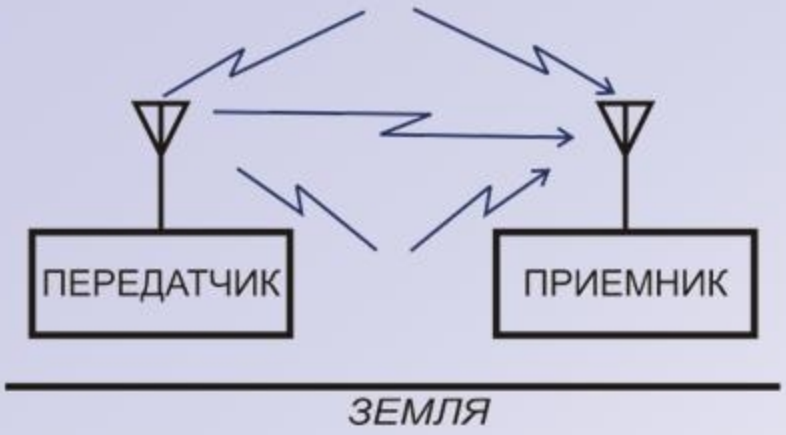


Рис.4.10. Структурная схема широкополосной системы передачи данных



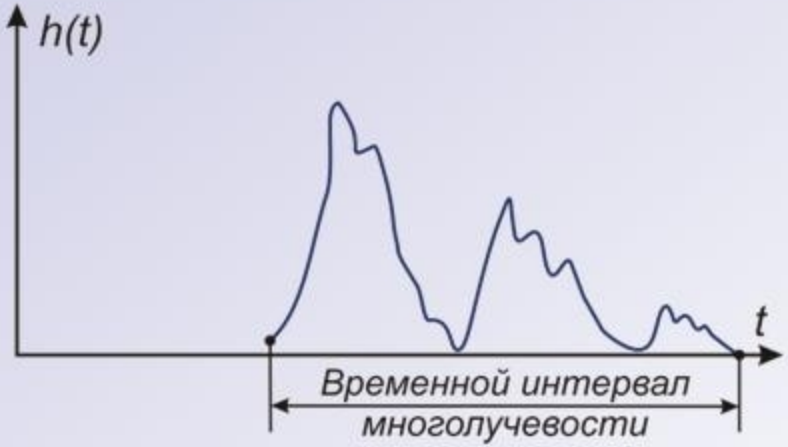


Рис.4.11. Типовой дисперсионный канал и его импульсная характеристика

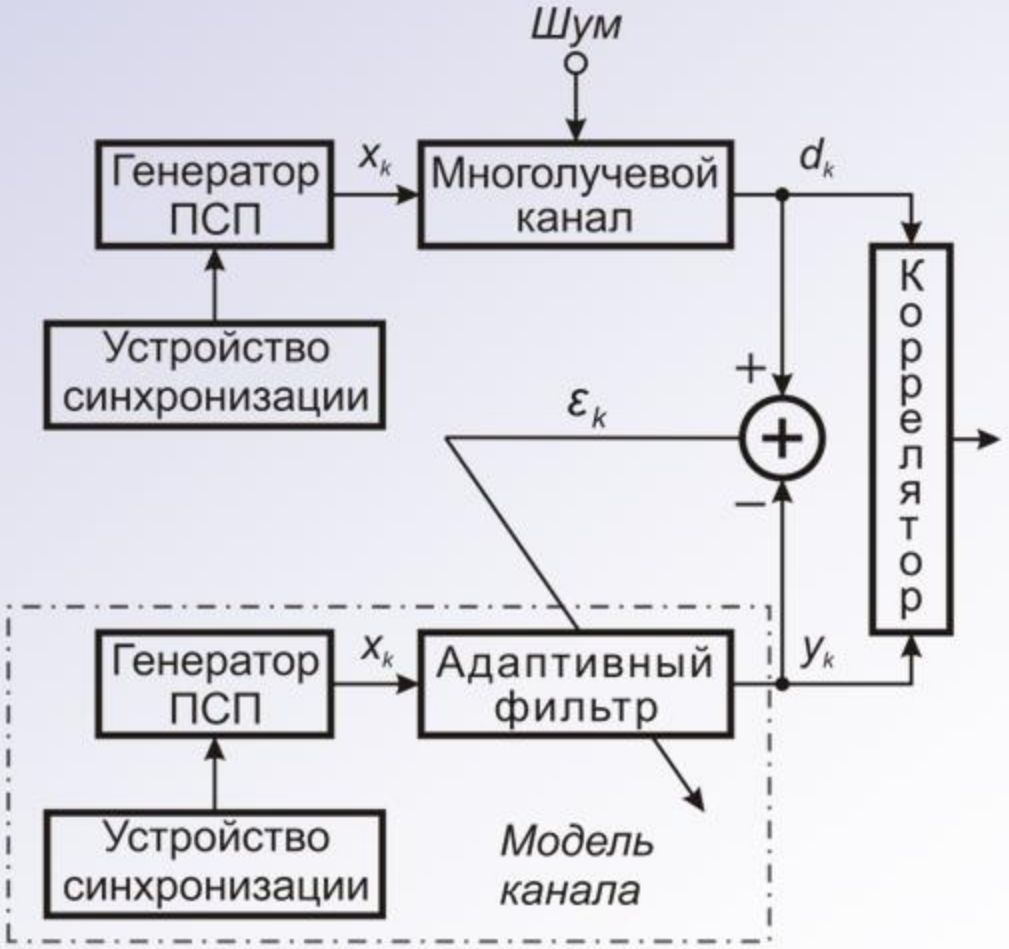


Рис.4.12. Схема адаптивного моделирования многолучевого канала

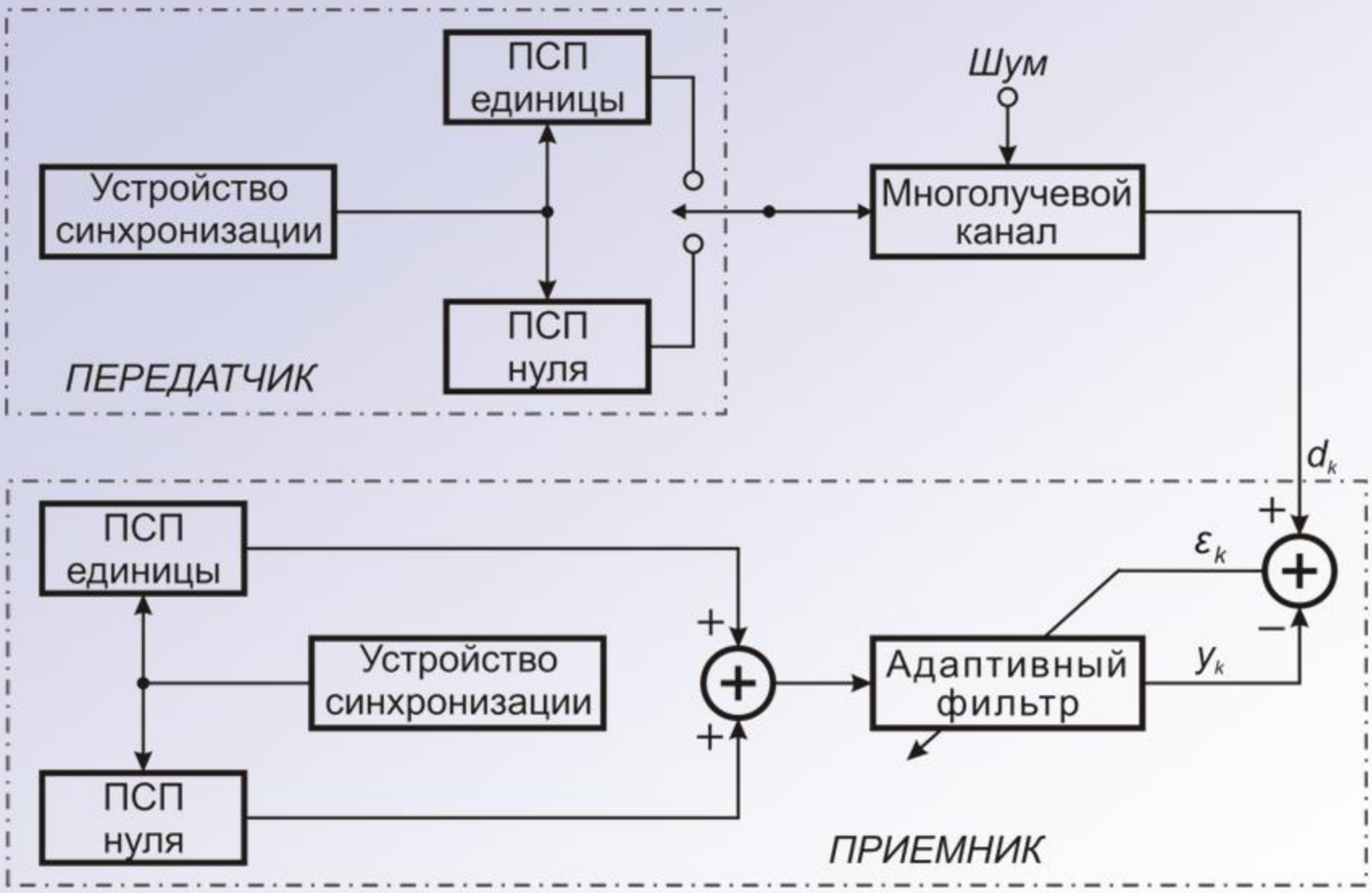


Рис.4.13. Схема прямого моделирования канала с одновременной передачей информации

4.5.2. Эхо-компенсация в телефонных сетях

Проблема электрического эха в телефонных сетях

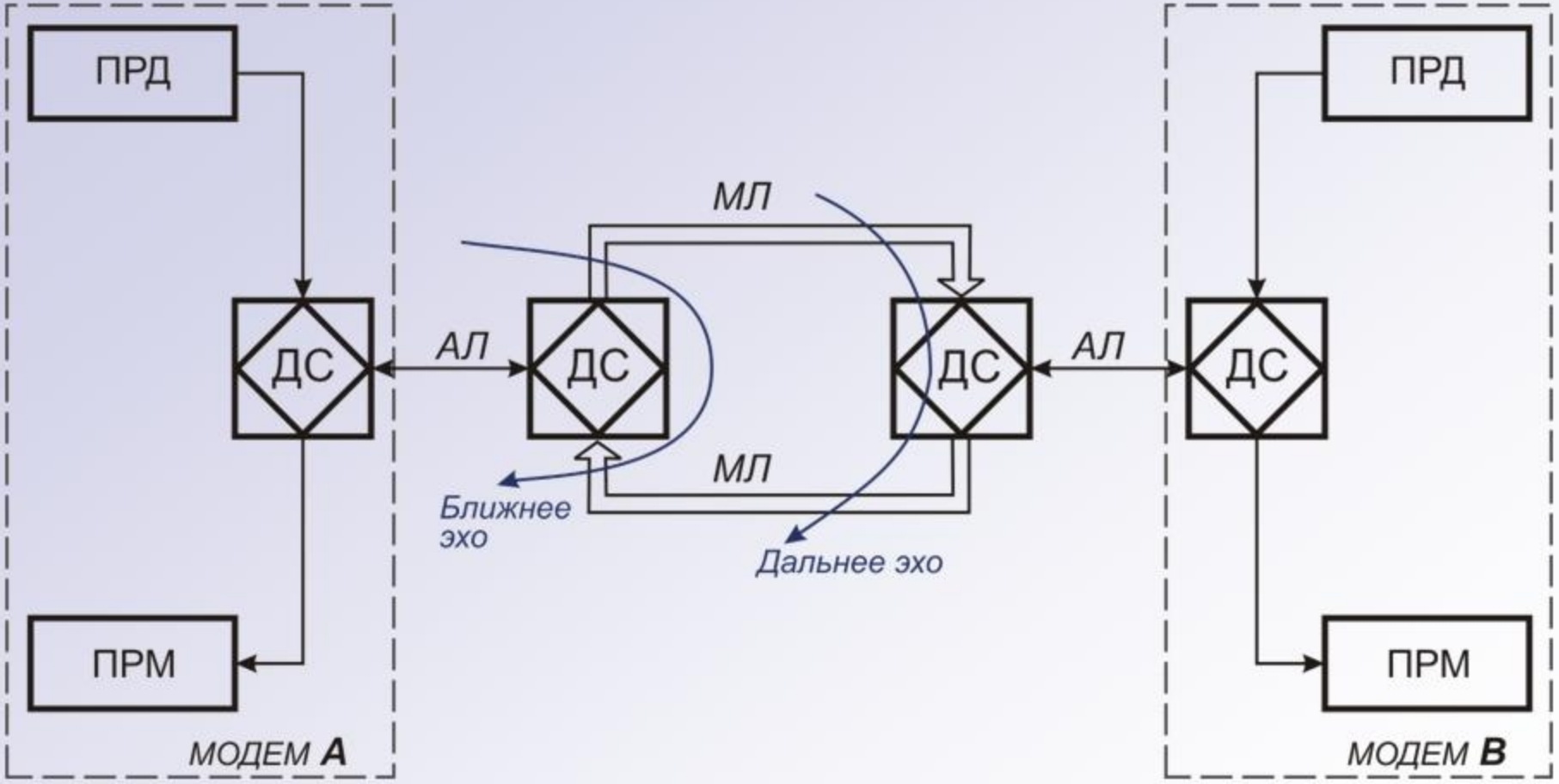


Рис.4.14. Передача данных по КТК с использованием модемов: источники и пути эхо-сигналов

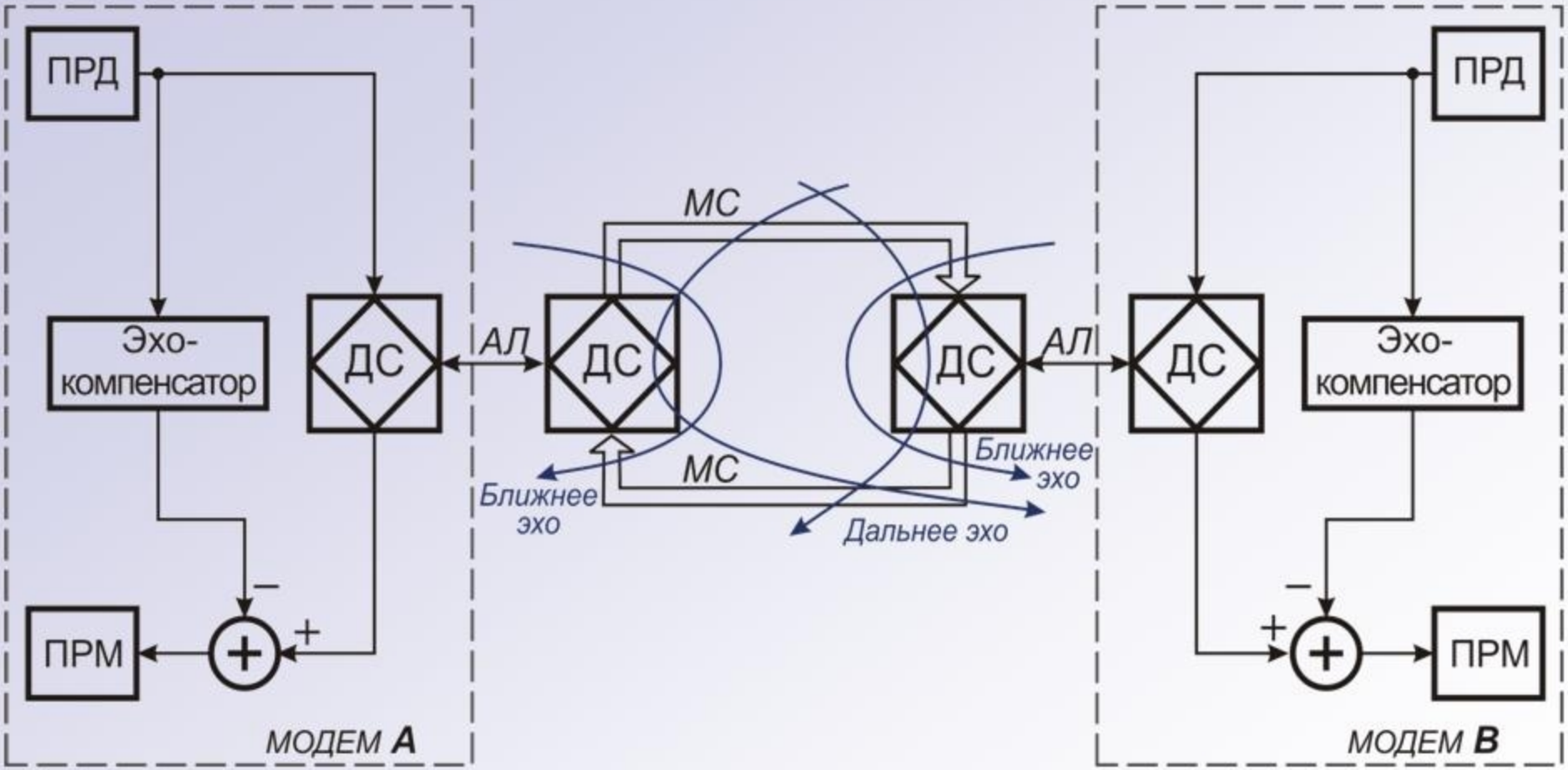


Рис.4.15. Передача данных по КТК с использованием модемов: дуплексный режим с применением эхо-компенсаторов

4.5.3. Адаптивное выравнивание частотных характеристик телефонных каналов (эквалайзеры)

Проблема передачи данных по реальным телефонным каналам

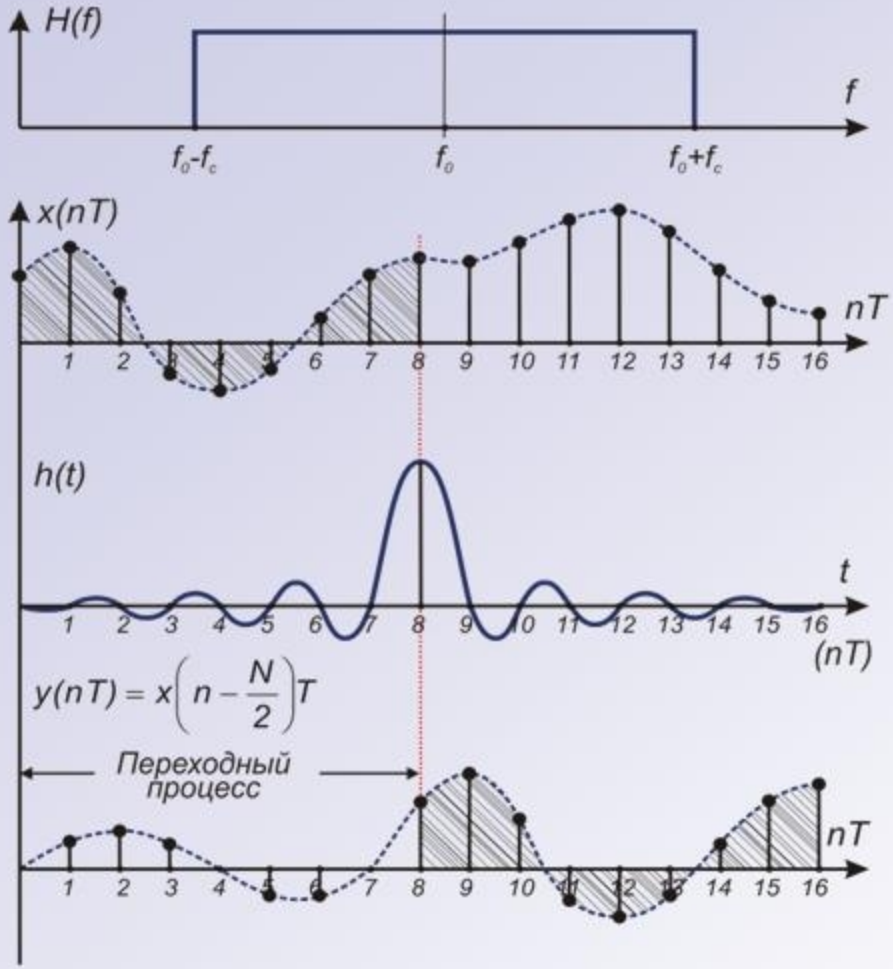


Рис.4.16. Цифровая связь в идеальном канале

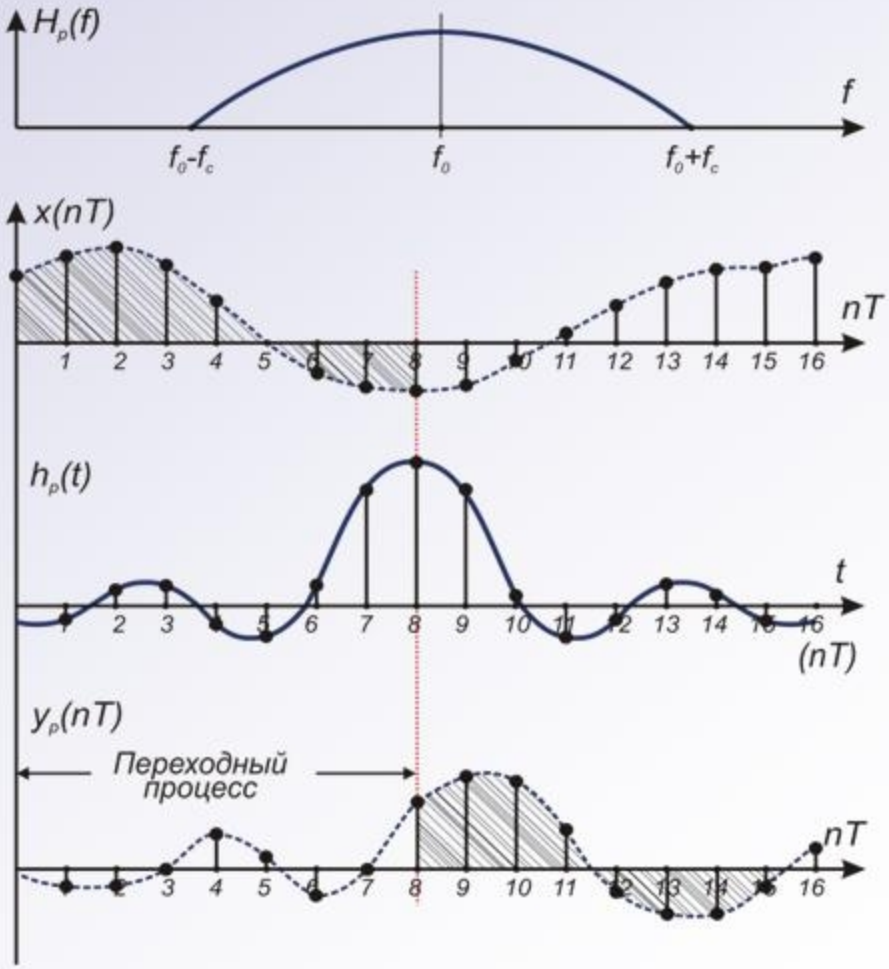


Рис.4.17. Цифровая связь в реальном канале

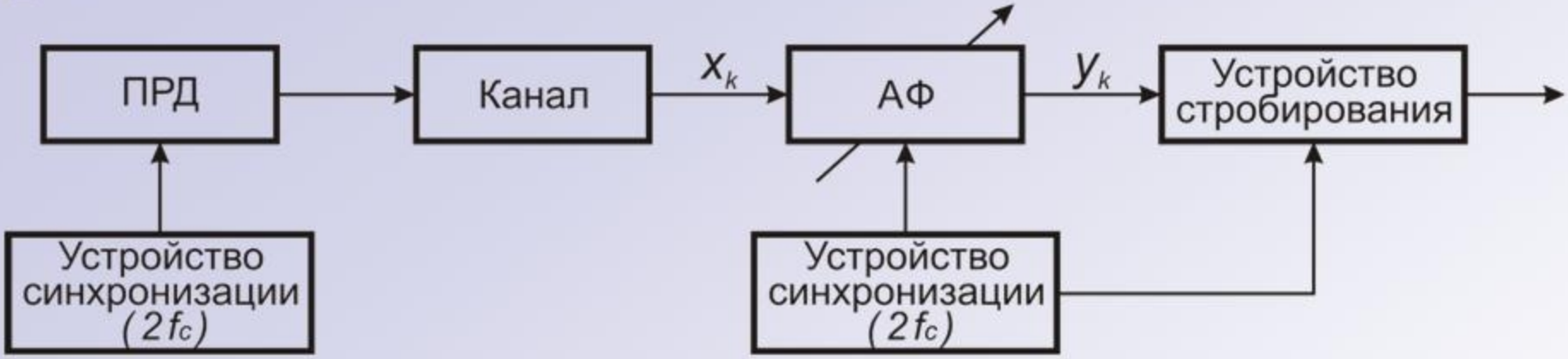


Рис.4.18. Схема связи с адаптивным выравниванием характеристик канала

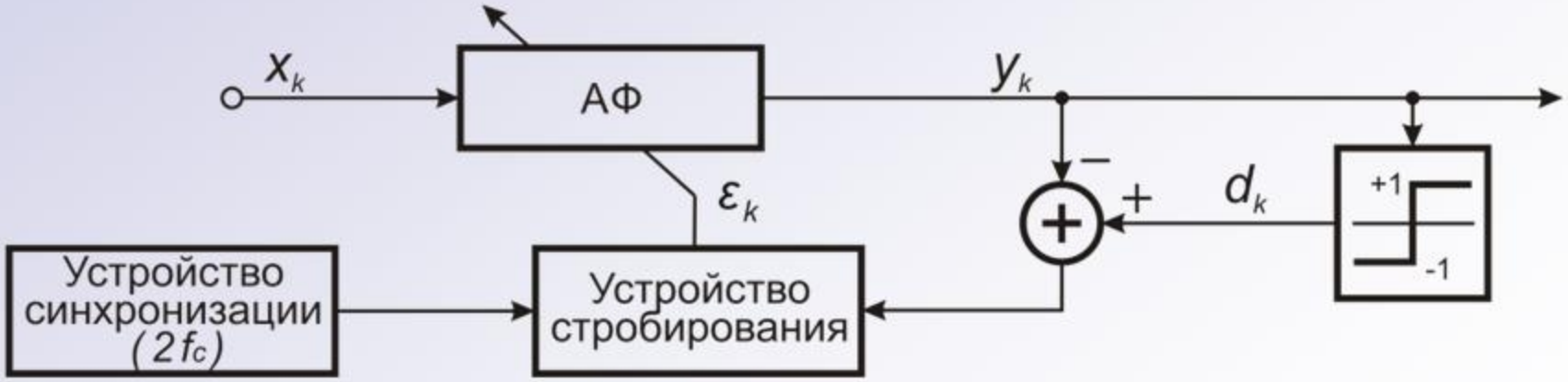


Рис.4.19. Схема адаптивного устройства выравнивания с обучением по решению

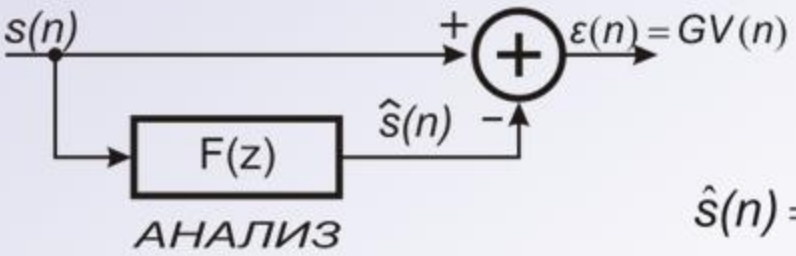
4.5.4. Кодирование речи с линейным предсказанием

Рис.4.20. Модель речеобразования

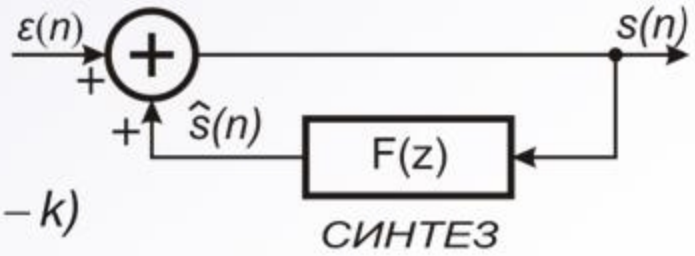
$$S(z) = V(z)H(z), \quad \text{где} \quad H(z) = \frac{G}{1 - \sum_{k=1}^{p} a_k z^{-k}} = \frac{G}{A(z)};$$

$$A(z) = 1 - F(z); \qquad F(z) = \sum_{k=1}^{p} a_k z^{-k}; \qquad s(n) = GV(n) + \sum_{k=1}^{p} a_k s(n-k).$$

Метод "анализа-синтеза"



$$\hat{s}(n) = \sum_{k=1}^{p} a_k s(n-k)$$



4.5.5. Подавление и фильтрация периодических сигналов с помощью адаптивного устройства предсказания

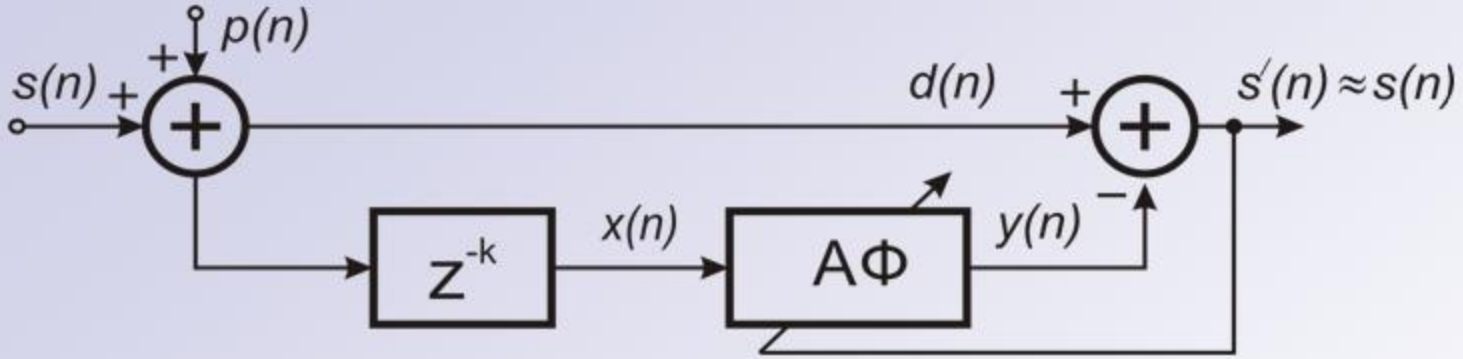


Рис.4.21. Адаптивное устройство подавления периодической узкополосной помехи

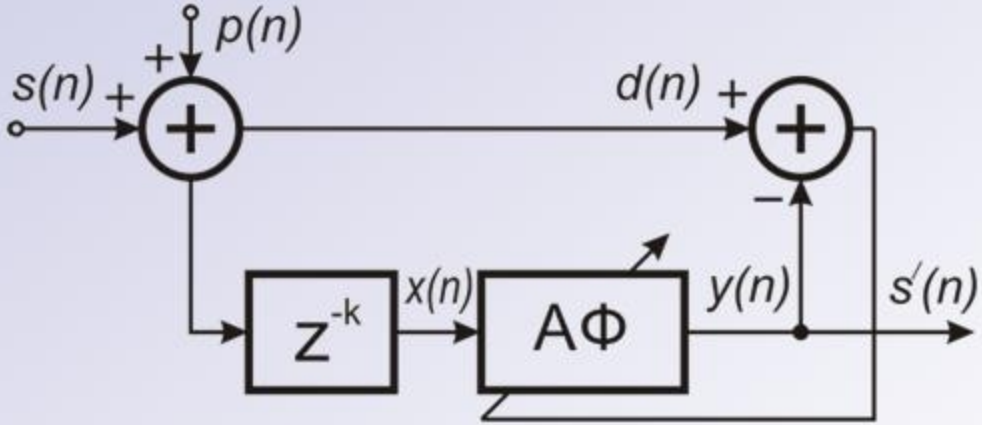


Рис.4.22. Адаптивное устройство подавления широкополосного шума

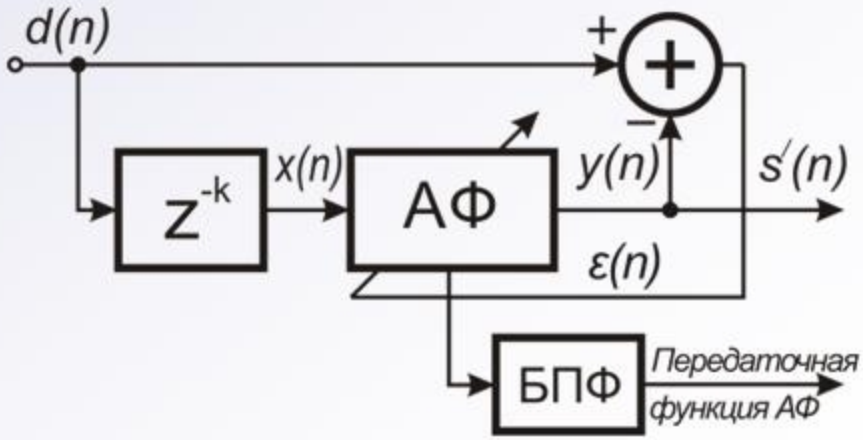


Рис.4.23. Адаптивный накопитель-обнаружитель

Цифровые сигнальные процессоры

1. Назначение и классификация

$$y(nT) = \sum_{k=0}^{N-1} x[(n-k)T]h(kT)$$ – КИХ-фильтр N-го порядка

$$y(nT) = \sum_{l=0}^{L-1} b_l x[(n-l)T] + \sum_{k=1}^{M} a_k y[(n-k)T]$$ – БИХ-фильтр M-го порядка

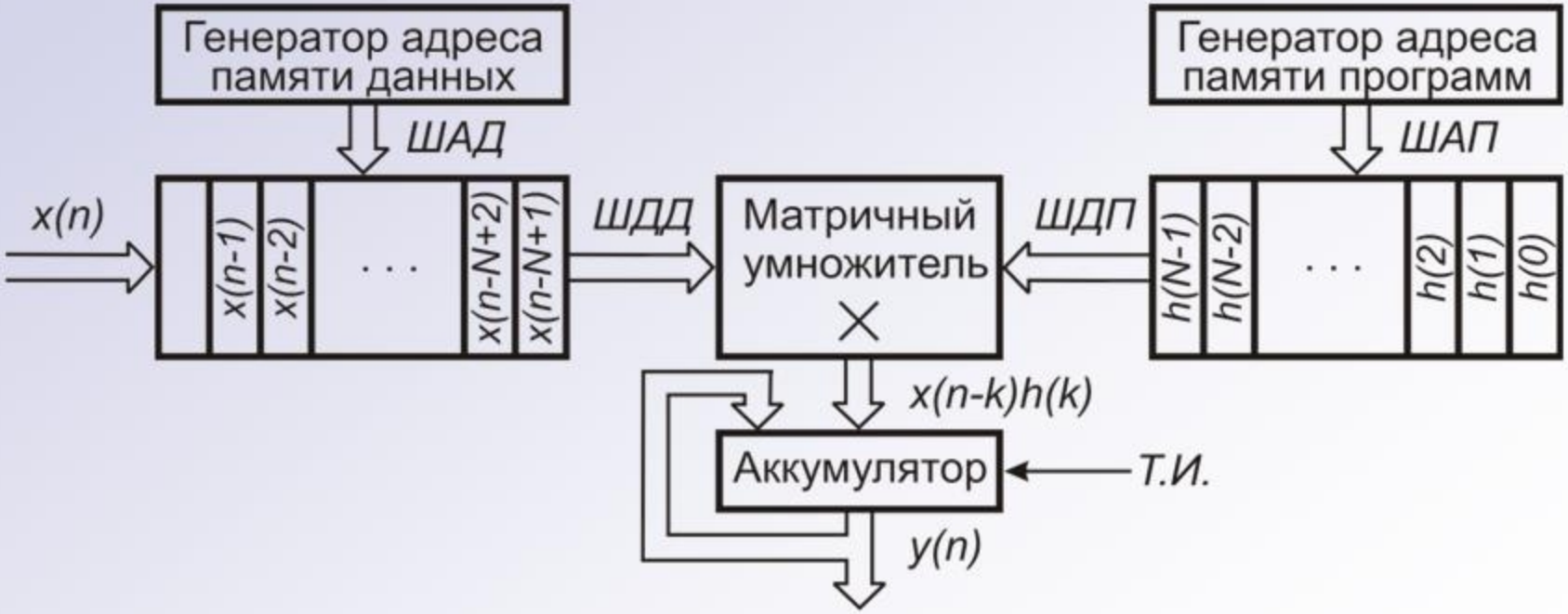


Рис. 1. Общая структура ЦСП, реализующего КИХ-фильтр

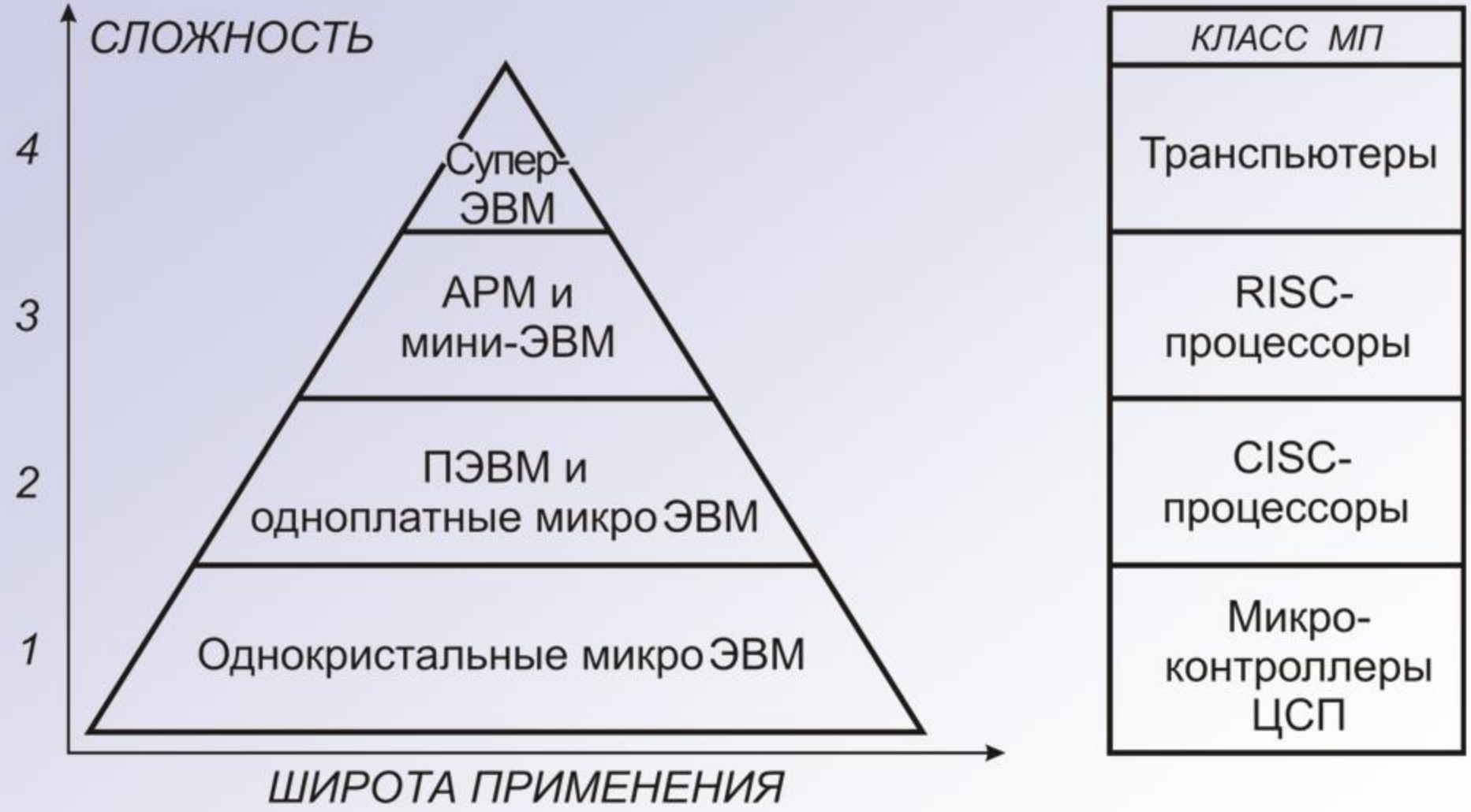


Рис. 2. Классификация вычислительных систем и устройств

Классификация ЦСП

1. ЦСП с фиксированной точкой – 16/24/32 - разрядные
2. ЦСП с плавающей точкой – 32/64 - разрядные
3. Сигнальные микроконтроллеры – 16 - разрядные с подсистемой аналогового ввода
4. Сигнальные транспьютеры – 32/64 - разрядные
5. Специализированные ЦСП

Ведущие фирмы *– разработчики и производители ЦСП*

1. Texas Instruments, США
2. Analog Devices, США
3. Motorola, США
4. НТЦ “МОДУЛЬ”, Россия
5. НТЦ “ЭЛВИС”, Россия

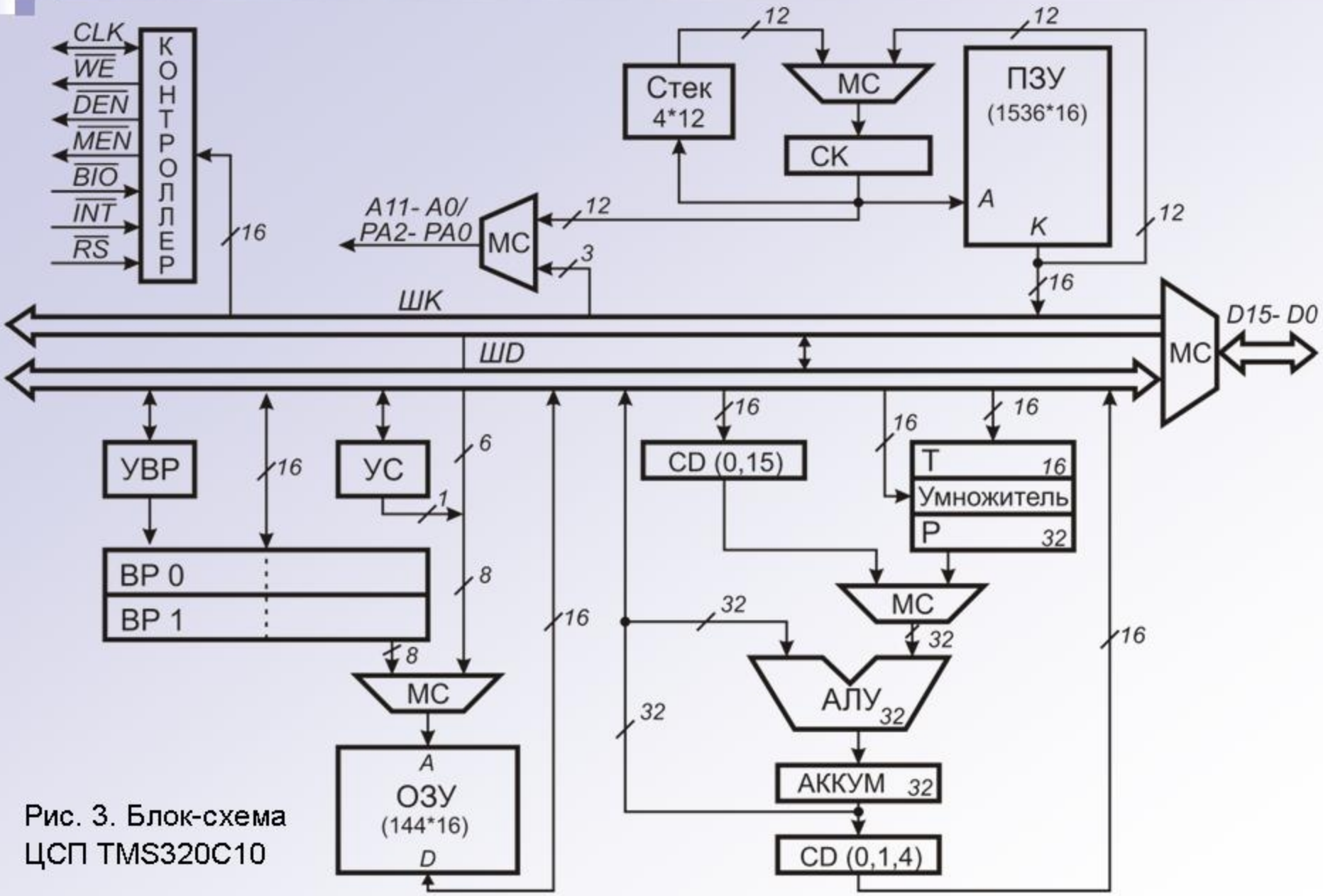


Рис. 3. Блок-схема
ЦСП TMS320C10

2. Цифровые сигнальные процессоры платформы C2000 фирмы Texas Instruments

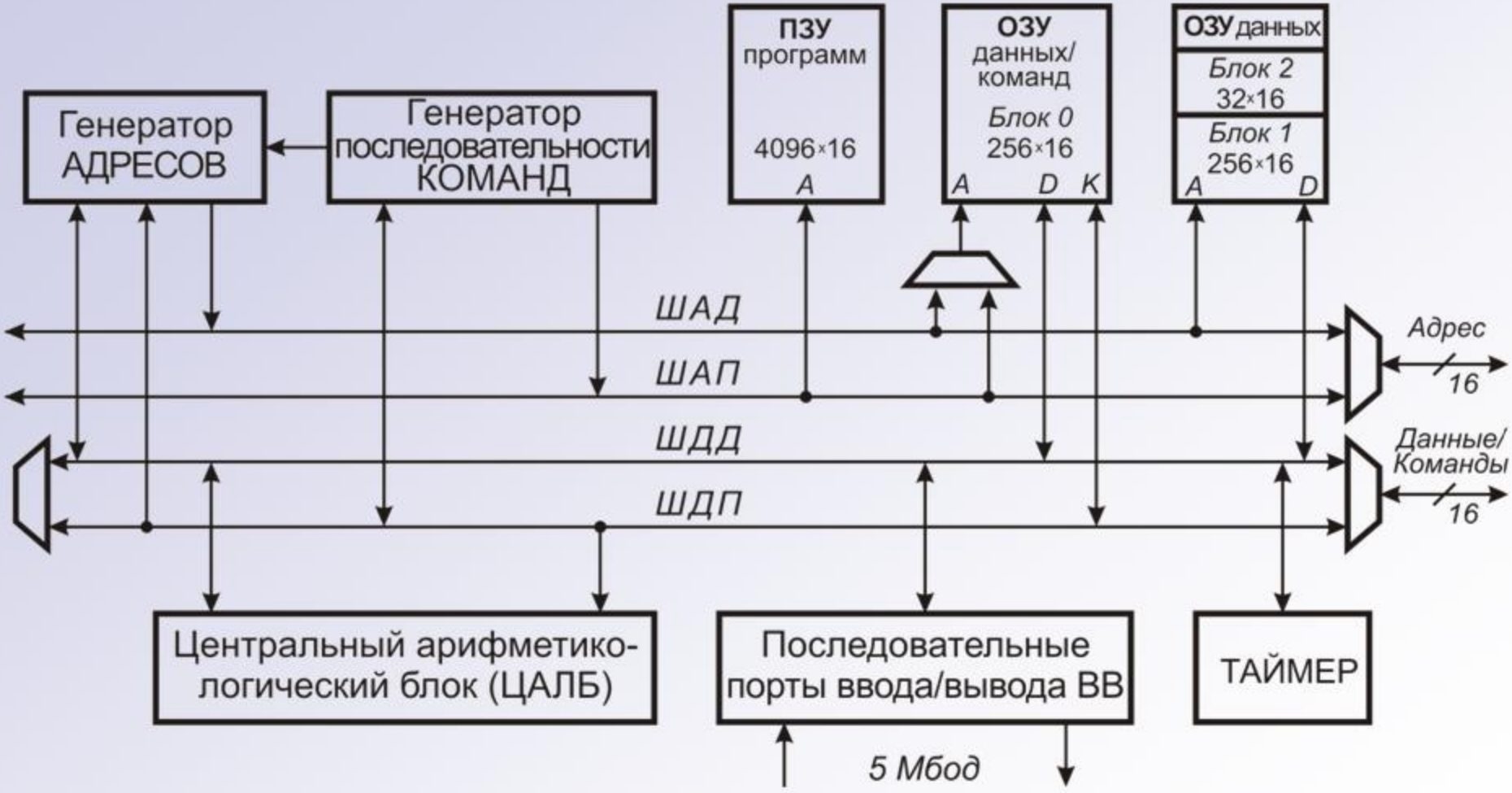


Рис. 4. Функциональная схема ЦСП TMS320C25

Организация памяти данных и программ

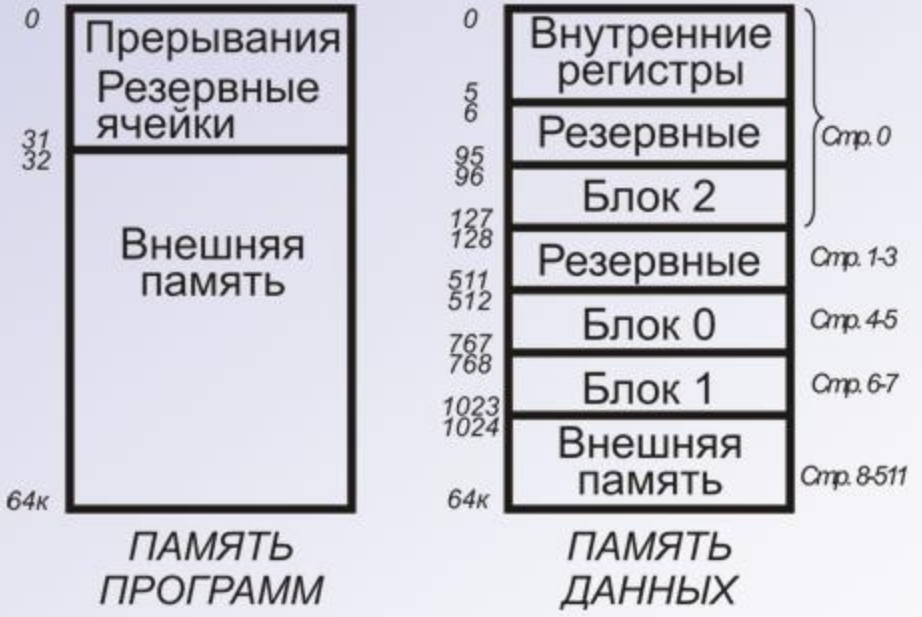
Внутренняя память:

Блок 0 – память данных / команд емкостью 256×16
Блок 1 – память данных емкостью 256×16
Блок 2 – память данных емкостью 32×16
CNFD – конфигурация Блока 0 как память данных
CNFP – конфигурация Блока 0 как память программ

Внешняя прямоадресуемая память: *64к ×16* – память данных
64к ×16 – память программ

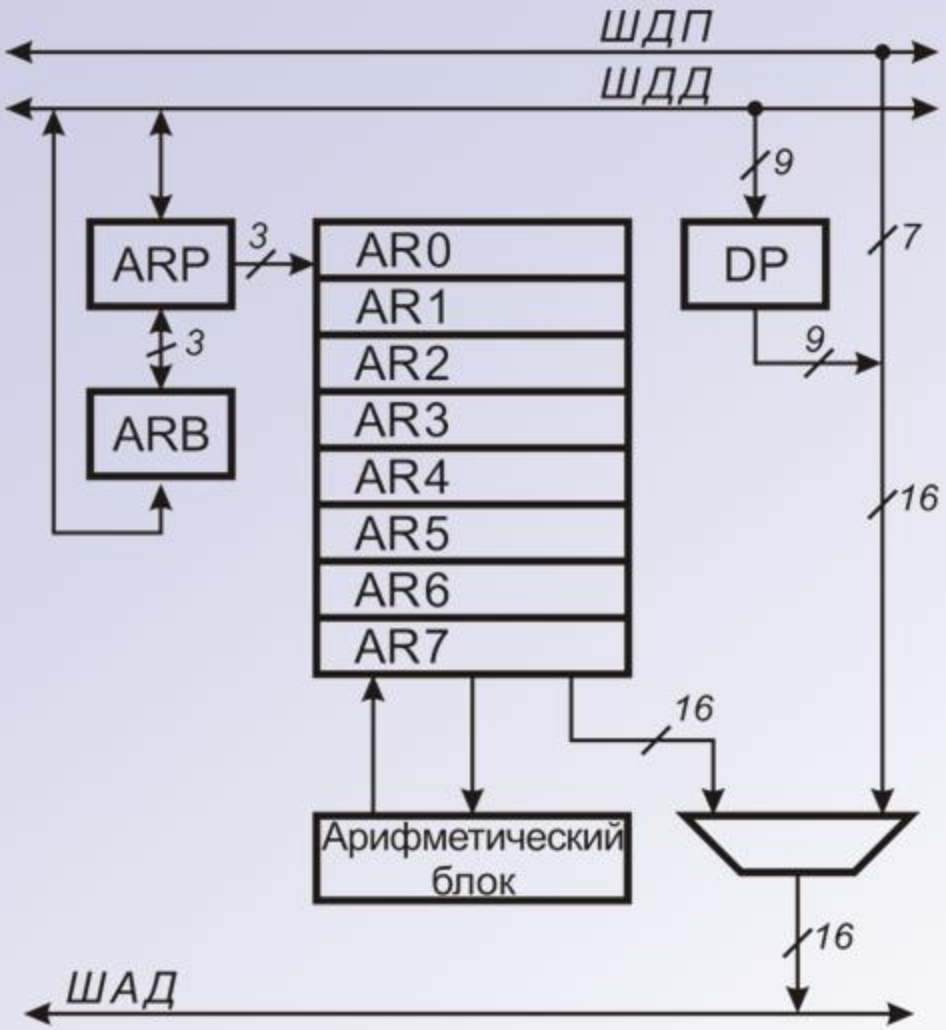
Карты распределения памяти процессора:

Блок 0 – память данных

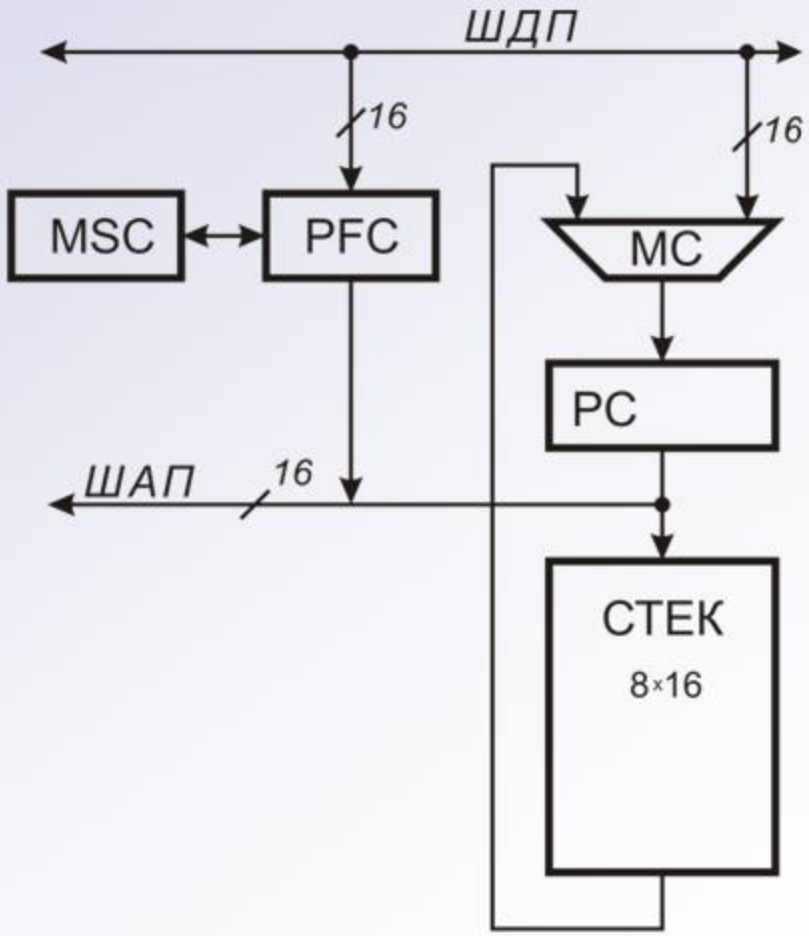


Блок 0 – память программ

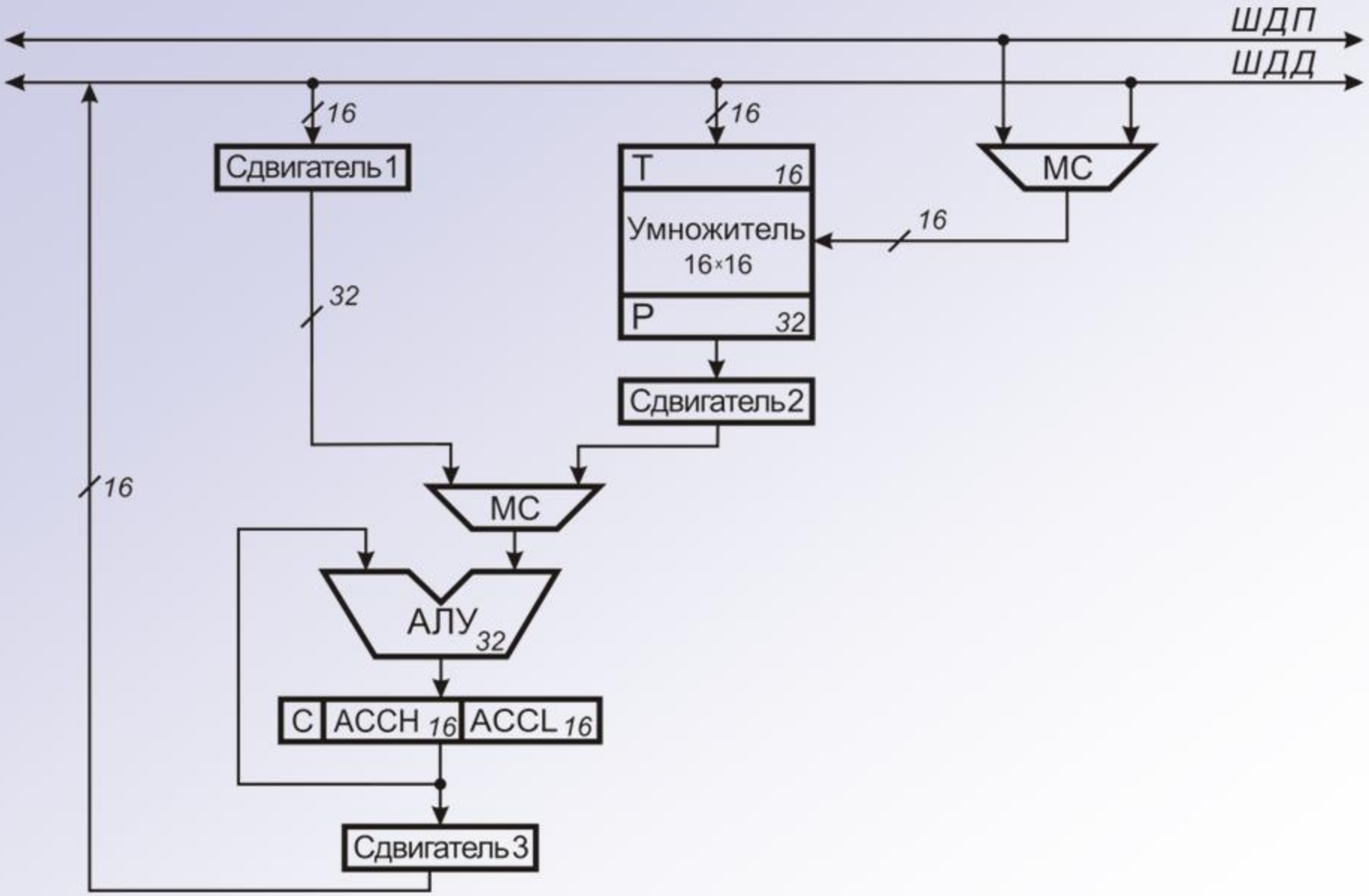
Генератор адреса памяти данных



Генератор последовательности команд



Центральный арифметико-логический блок



Управление системой, интерфейс ввода-вывода и конфигурации системы

В *управлении* работой системы участвуют:

1. Регистры состояния *ST0, ST1*
2. Таймерный регистр *TIM* и регистр периода *PRD*
3. Счетчик повторений RPTC
4. Контроллер прерываний

Интерфейс ввода-вывода:

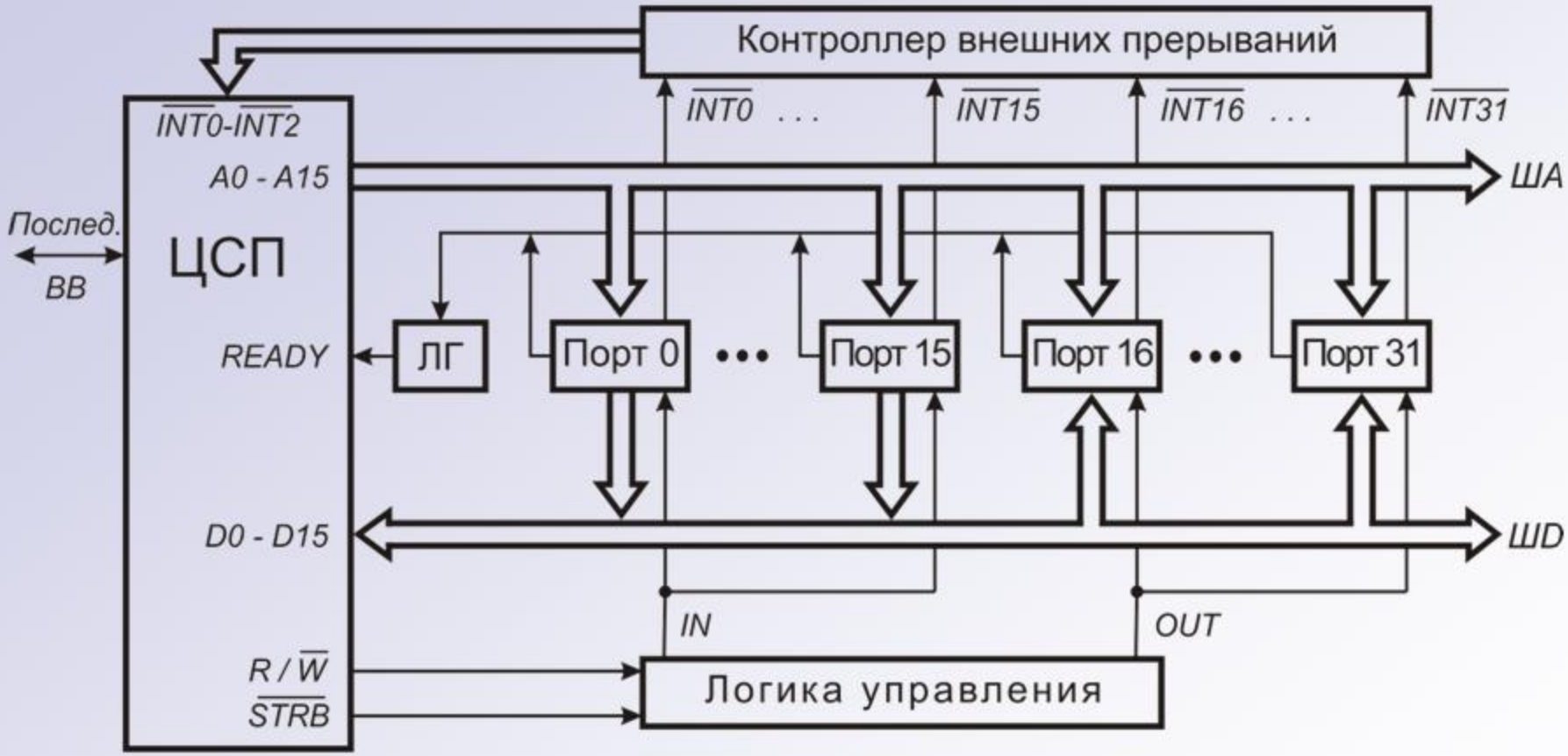
1. Параллельный ввод-вывод: 16-разрядным параллельным кодом со скоростью до 20 Мбайт/с.
2. Последовательный порт ВВ: со скоростью до 5 Мбит/с.

Обслуживание прерываний:

1. Внутренние прерывания:
   - TINT – от таймера
   - TRAP – командное
   - RINT – прием портом ВВ
   - XINT – передача через порт ВВ
2. Внешние прерывания:
   - INT2, INT1, INT0 – внешние маскируемые прерывания
   - RST – сброс

Конфигурации системы

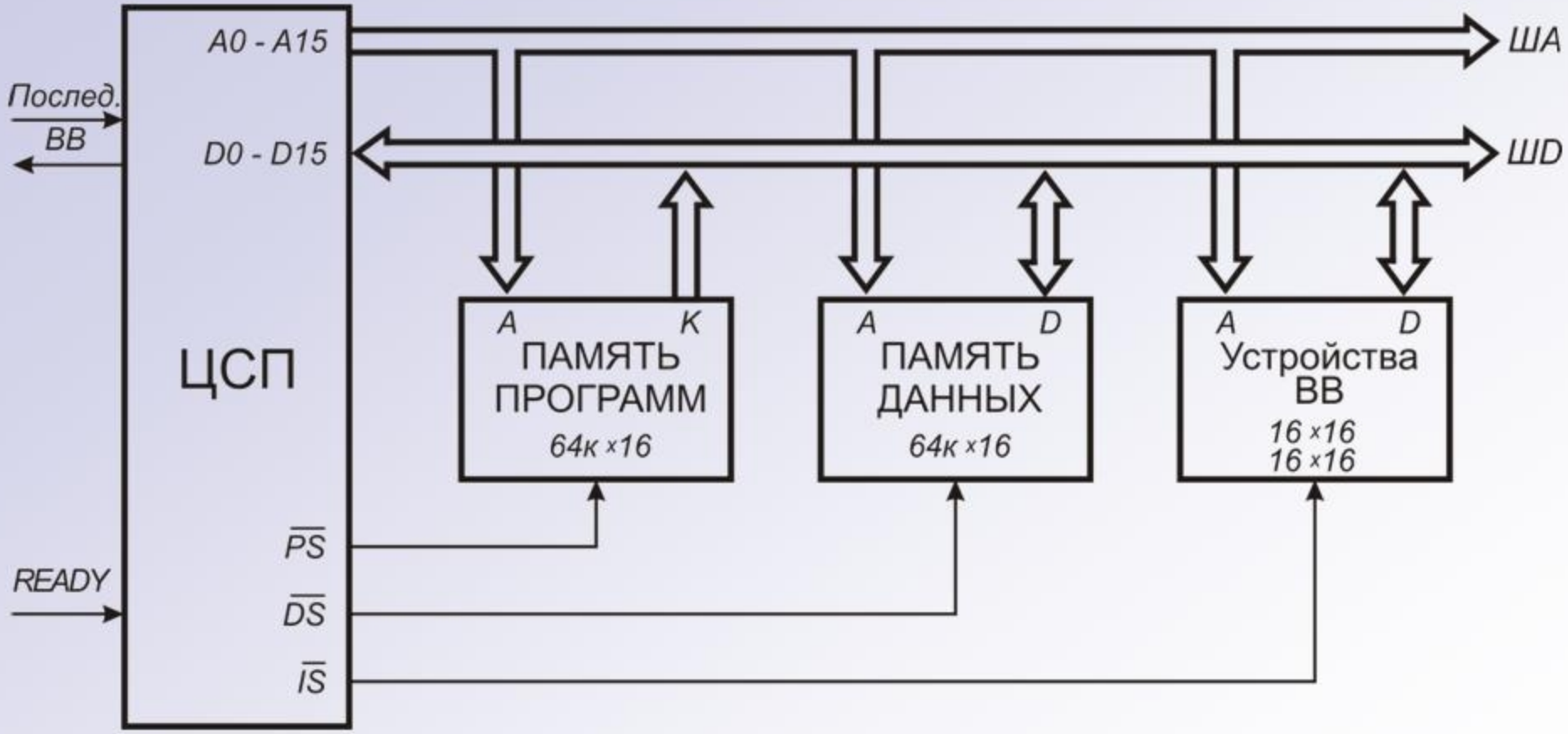
1. *Автономная система*



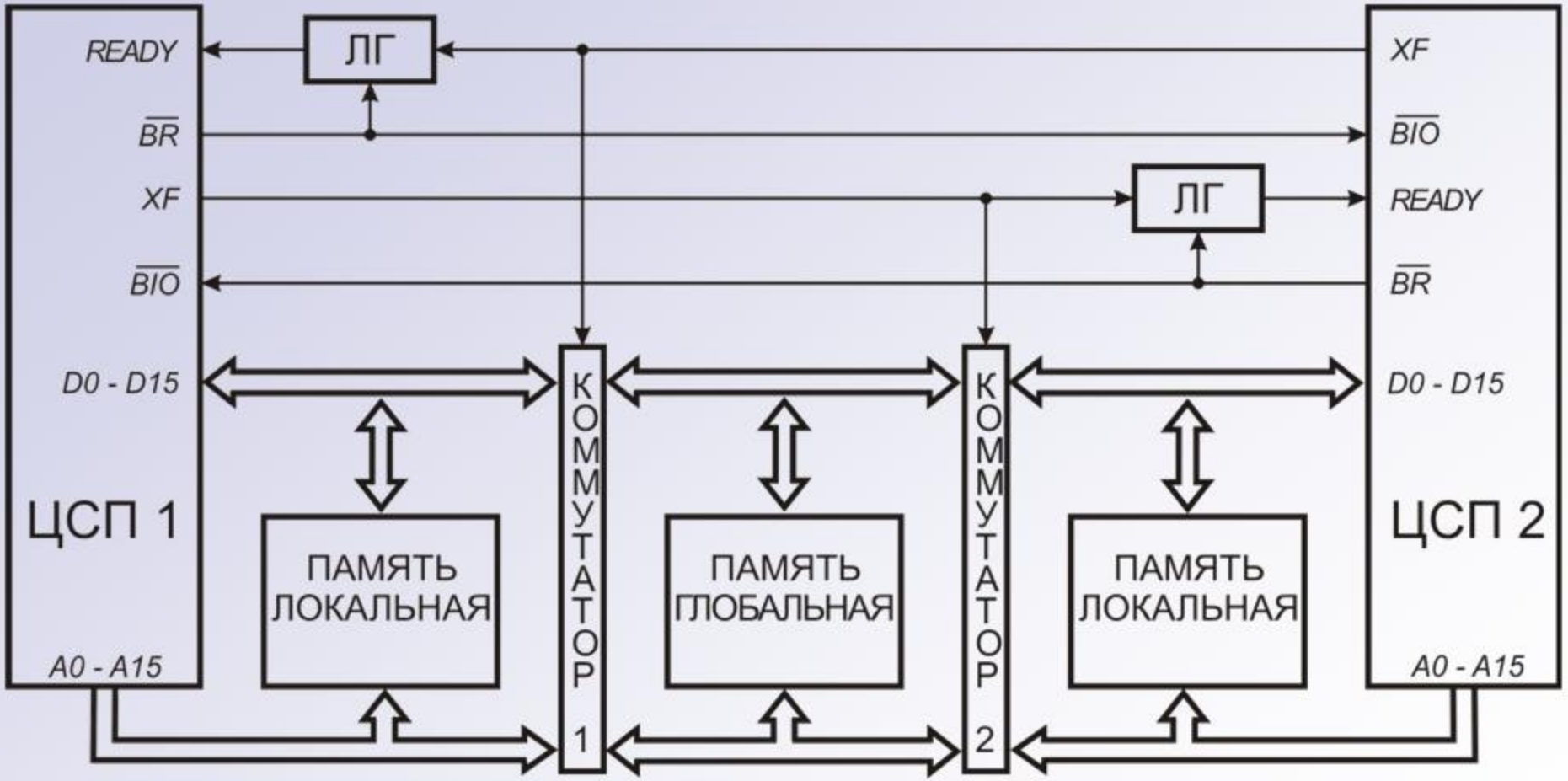
Способы обмена с ВУ:

1. Синхронный ввод-вывод
2. Асинхронный ввод-вывод
3. Обмен по прерыванию

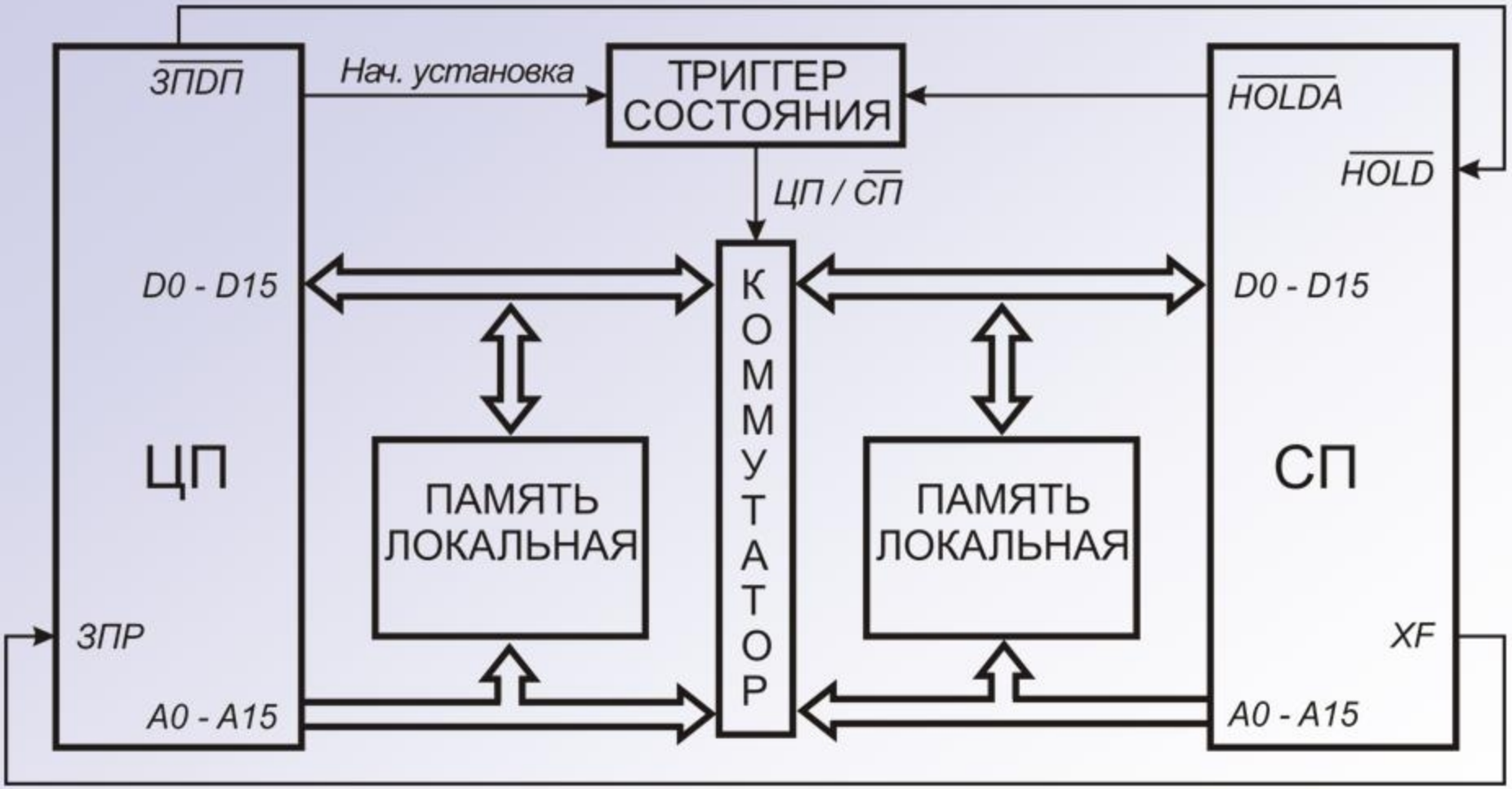
2. Минимальная конфигурация



3. Максимальная конфигурация



4. Двухуровневая иерархическая конфигурация



СПОСОБЫ АДРЕСАЦИИ ЦПОС TMS320C25

1. Прямая адресация.

< *instr* > ␣ < *dma* >, < *shift* >

Пример: ADD 27,3

2. Косвенная адресация

< *instr* > ␣ < *indirect* >, < *shift* >, < *ARX* >

Модификации < indirect >:

1. * – простая косвенная
2. **+* – косвенная с инкрементом текущего *ARX*
3. **-* – косвенная с декрементом текущего *ARX*
4. **0+* – косвенная с инкрементом текущего *ARX* числом, содержащемся в AR0
5. **0-* – косвенная с декрементом текущего *ARX* числом, содержащемся в AR0

Пример: ADD *+,8,3

3. Непосредственная адресация

< *instr* > ␣ < *const* >

Пример: ADDK 25

КОМАНДЫ РАБОТЫ С АККУМУЛЯТОРОМ

1. *ABS* – абсолютное значение АКК
2. *ADD* – прибавить к АКК со сдвигом
3. *ADDC* – прибавить к АКК с переносом
4. *ADDK* – прибавить к АКК короткую константу (8 бит)
5. *ADLK* – прибавить к АКК длинную константу (16 бит) со сдвигом
6. *AND* – логическое умножение с младшим словом АКК
7. *ANDK* – логическое умножение АКК и длинной константы со сдвигом
8. *LAC* – загрузка АКК со сдвигом
9. *LACK* – загрузка АКК короткой константой
10. *LALK* – загрузка АКК длинной константой со сдвигом
11. *NEG* – отрицание АКК: $-(АКК) \rightarrow АКК$
12. *OR* – логическое ИЛИ с младшим словом АКК
13. *ORK* – логическое ИЛИ АКК и длинной константы со сдвигом
14. *ROL* – циклический сдвиг АКК влево $(АКК\ 31 \rightarrow C;\ C \rightarrow АКК\ 0)$
15. *ROR* – циклический сдвиг АКК вправо $(C \rightarrow АКК\ 31;\ АКК\ 0 \rightarrow C)$

КОМАНДЫ РАБОТЫ С АККУМУЛЯТОРОМ

16. *SACH* – сохранить старшее слово АКК со сдвигом
17. *SACL* – сохранить младшее слово АКК со сдвигом
18. *SBLK* – вычесть из АКК длинную константу со сдвигом
19. *SFL* – сдвиг АКК влево: *0 → АКК 0*
20. *SFR* – сдвиг АКК вправо: *АКК 31 → АКК 31*
21. *SUB* – вычесть из АКК со сдвигом
22. *SUBB* – вычесть из АКК с заемом
23. *SUBH* – вычесть из старшего слова АКК
24. *SUBK* – вычесть из АКК короткую константу
25. *XOR* – “исключающее ИЛИ” с младшим словом АКК
26. *XORK* – “исключающее ИЛИ” АКК и длинной константы со сдвигом
27. *ZAC* – установка в нуль АКК
28. *ZALH* – обнуление младшего слова АКК и загрузка старшего слова АКК
29. *ZALR* – то же с округлением
30. *ZALS* – обнуление АКК и загрузка младшего слова АКК без учета знака

КОМАНДЫ РАБОТЫ СО ВСПОМОГАТЕЛЬНЫМИ РЕГИСТРАМИ

1. *ADRK* – прибавить к ARX короткую константу
2. *LAR* – загрузить ARX
3. *LARK* – загрузить ARX короткой константой
4. *LARP* – загрузить указатель ARX
5. *LDP* – загрузить указатель страниц
6. *LDPK* – загрузить указатель страниц константой (9 бит)
7. *LRLK* – загрузить ARX длинной константой
8. *MAR* – модификация ARX
9. *SAR* – сохранить ARX
10. *SBRK* – вычесть из ARX короткую константу (8 бит)

КОМАНДЫ УМНОЖЕНИЯ И РАБОТЫ с T и P – регистрами

1. *APAC* – прибавить P-регистр к АКК
2. *LT* – загрузить T-регистр
3. *LTA* – загрузить T-регистр и прибавить P-регистр к АКК
4. *LTD* – загрузить T-регистр, прибавить P-регистр к АКК и копировать память данных
5. *LTP* – загрузить T-регистр и запомнить P-регистр в АКК
6. *LTS* – загрузить T-регистр и вычесть P-регистр из АКК
7. *MAC* – умножить и просуммировать нарастающим итогом
8. *MACD* – умножить, просуммировать нарастающим итогом и копировать память данных
9. *MPY* – умножить T-регистр на переменную
10. *MPYK* – умножить T-регистр на константу (13 бит)
11. *PAC* – загрузить P-регистр в АКК
12. *SPAC* – вычесть P-регистр из АКК
13. *SPH* – сохранить старшее слово P-регистра
14. *SPL* – сохранить младшее слово P-регистра
15. *SORA* – возвести в квадрат и просуммировать нарастающим итогом
16. *SORS* – возвести в квадрат и вычесть P-регистр из АКК

КОМАНДЫ ПЕРЕХОДА И ВЕТВЛЕНИЯ

1. *B* – безусловный переход *(pma → PC)*
2. *BACC* – безусловный переход по адресу из АКК
3. *BANZ* – условный переход по ненулевому состоянию ARX
4. *BC* – условный переход по переносу *(C=1)*
5. *BGEZ* – условный переход, если *АКК ≥ 0*
6. *BGZ* – условный переход, если *АКК > 0*
7. *BIOZ* – условный переход, если $\overline{BIO}$*=0*
8. *BLEZ* – условный переход, если *АКК ≤ 0*
9. *BLZ* – условный переход, если *АКК < 0*
10. *BNC* – условный переход, если перенос *C=0*
11. *BNZ* – условный переход, если *АКК ≠ 0*
12. *BZ* – условный переход по нулевому состоянию АКК
13. *CALL* – вызов подпрограммы
14. *RET* – возврат из подпрограммы

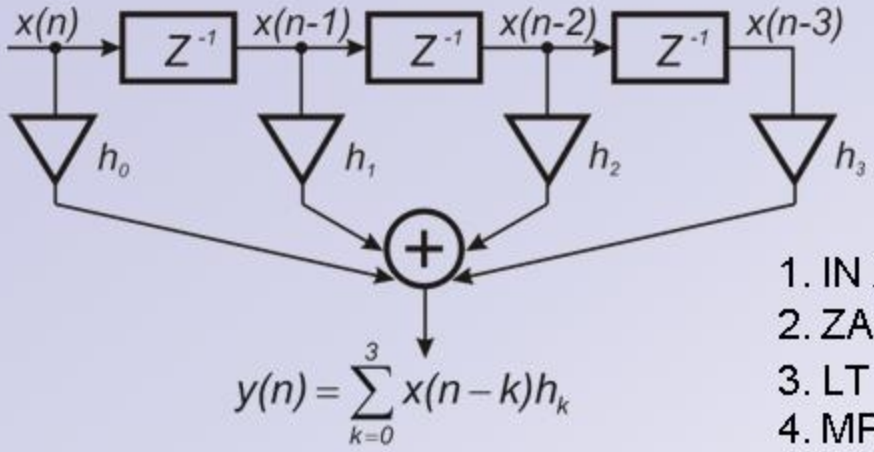
КОМАНДЫ ВВОДА-ВЫВОДА И РАБОТЫ С ПАМЯТЬЮ

1. *BLKD* – блоковое перемещение внутри памяти данных
2. *BLKP* – блоковое перемещение из памяти программ в память данных
3. *DMOV* – перемещение (копирование) данных в памяти данных
4. *IN* – ввод данных из порта
5. *OUT* – вывод данных в порт
6. *RXF* – сброс внешнего флага XF
7. *SXF* – установка внешнего флага XF
8. *TBLR* – чтение таблицы
9. *TBLW* – запись в таблицу

КОМАНДЫ УПРАВЛЕНИЯ

1. *CNFD* – конфигурация блока 0 – память данных
2. *CNFP* – конфигурация блока 0 – память программ
3. *DINT* – блокирование прерывания
4. *EINT* – разрешение прерывания
5. *IOLE* – ожидание до прерывания
6. *LST* – загрузка регистра состояния ST0
7. *LST1* – загрузка регистра состояния ST1
8. *NOP* – нет операции
9. *POP* – выталкивание из стека в младшее слово АКК
10. *POPD* – выталкивание из стека в память данных
11. *PSHD* – проталкивание в стек ячейки памяти данных
12. *PUSH* – проталкивание в стек младшего слова АКК
13. *RC* – сброс бита переноса
14. *RPTK* – повторение следующей команды число раз, указанное непосредственной константой
15. *SC* – установка бита переноса
16. *TRAP* – программное прерывание

ПРОГРАММИРОВАНИЕ ЦПОС TMS320C25



x(n) ↔ M:<XN>

x(n-1) ↔ M:<XN1>

x(n-2) ↔ M:<XN2>

x(n-3) ↔ M:<XN3>

y(n) ↔ M:<YN>

PA0 → x(n)

y(n) → PA1

1. IN XN,PA0	$x(n) \rightarrow M:<XN>$
2. ZAC	$0 \rightarrow АКК$
3. LT XN3	$x(n-3) \rightarrow$ Т-регистр
4. MPYK H3	$x(n-3)\times h_3 \rightarrow$ Р-регистр
5. LTD XN2	$x(n-2) \rightarrow$ Т-регистр $x(n-3)h_3 \rightarrow АКК$ $x(n-2) \rightarrow x(n-3)$
6. MPYK H2	$x(n-2)\times h_2 \rightarrow$ Р-регистр
7. LTD XN1	$x(n-1) \rightarrow$ Т-регистр $x(n-3)h_3 + x(n-2)h_2 \rightarrow АКК$ $x(n-1) \rightarrow x(n-2)$
8. MPYK H1	$x(n-1)\times h_1 \rightarrow$ Р-регистр
9. LTD XN	$x(n) \rightarrow$ Т-регистр $x(n-3)h_3 + x(n-2)h_2 + x(n-1)h_1 \rightarrow АКК$ $x(n) \rightarrow x(n-1)$
10. MPYK H0	$x(n)\times h_0 \rightarrow$ Р-регистр
11. APAC	$x(n-3)h_3 + x(n-2)h_2 + x(n-1)h_1 + x(n)h_0 \rightarrow АКК$
12. SACH YN,1	$АКК \rightarrow M:<YN>$
13. OUT YN,PA1	$y(n) \rightarrow PA1$