Київський національний університет імені Тараса Шевченка

Колєнов С. О.

ЦИФРОВИЙ ЗВ'ЯЗОК

Методичний посібник до лабораторного практикуму
для студентів радіофізичного факультету

Київ 2013

УДК 681.3

Цифровий зв’язок: Методичний посібник до лабораторного практикуму для студентів радіофізичного факультету / Колєнов С. О. – Київ: Радіофізичний факультет Київського національного університету імені Тараса Шевченка, 2013. – 76 с.

Рецензент: доцент кафедри нанофізики та наноелектроніки Київського національного університету імені Тараса Шевченка, кандидат фізико-математичних наук Коваленко А. В.

Методичний посібник до лабораторного практикуму складений у відповідності з навчальною програмою дисципліни "Цифровий зв’язок", що викладається на кафедрі квантової радіофізики радіофізичного факультету. В описі лабораторних робіт вказані мета роботи, короткі теоретичні відомості, схема експериментальної установки, завдання до роботи, методичні вказівки щодо проведення досліджень та контрольні питання для самостійної роботи студентів. Посібник містить 3 лабораторні роботи, в яких вивчаються пряме цифрове перетворення частоти, ефективне завадостійке кодування інформації в цифрових каналах зв’язку, кореляційний прийом цифрових сигналів в каналі з шумом.

Ухвалено Вченою радою радіофізичного факультету
Київського національного університету імені Тараса Шевченка
(протокол №___ від ________)

Вступ

Даний лабораторний практикум виконується в рамках дисципліни "Цифровий зв’язок", що викладається студентам кафедри квантової радіофізики радіофізичного факультету. Метою практикуму є ознайомлення студентів з основними методами та технічною реалізацією систем цифрового зв’язку, що включає пряме цифрове перетворення частоти, вивчення методів завадостійкого та ефективного кодування інформації, а також кореляційний прийом цифрових сигналів в каналі з шумом.

В ході підготовки до роботи студенти повинні ознайомитись з описом лабораторної роботи, при наявності вимірювального стенду ознайомитись з його схемою та приладами, що необхідні для виконання роботи, а також з’ясувати призначення окремих елементів стенду та порядок їх увімкнення, скласти необхідні функціональні та принципові схеми, виконати передбачені описом попередні розрахунки. Перед виконанням роботи кожен студент зобов'язаний представити заготовлену форму звіту, що містить необхідні розрахунки, функціональні або принципові схеми. Виконанню роботи передує перевірка готовності студента до роботи, в ході якої студент отримує питання за темою роботи. При задовільних відповідях студент допускається до виконання роботи.

Звіт по роботі оформлюється за встановленим зразком і повинен містити:

1. Опис явища або приладу, що досліджується у роботі (основні властивості, параметри, характеристики, області застосування).
2. Схему та опис вимірювальної установки (якщо присутня в роботі).
3. Хід роботи та коротке викладення суті вимірювань, що проводяться.
4. Результати вимірювань та розрахунків у вигляді таблиць та графіків, а також формули з прикладами розрахунків.
5. Висновки по роботі та критичні оцінки отриманих результатів, виходячи з співставлення їх з теоретичними розрахунками, довідковими даними чи лекційним матеріалом.

Студент отримує залік по роботі після представлення оформленого звіту та пояснення отриманих результатів. Студент, який не отримав залік по зробленій роботі, до наступної роботи не допускається.

Лабораторна робота № 1

ПРЯМЕ ЦИФРОВЕ ПЕРЕТВОРЕННЯ ЧАСТОТИ

Мета роботи: аналіз спектрів амплітудно-модульованих сигналів з використанням прямого цифрового перетворенням частоти.

Література:

1. Гоноровский И. С. Радиотехнические цепи и сигналы. – М.: Радио, 1986.
2. Харкевич А. А. Спектры и анализ. – М.: URSS, 2009, 236 с.
3. Робинер Л., Гоулд Б. Теория и применение цифровой обработки сигналов. – М.: Мир, 1978.
4. Гольденберг Л.М. и др. Цифровая обработка сигналов: Учебное пособие для вузов.- М.: Радио и связь, 1990.- 256 с.

Теоретичні відомості

За своєю природою всі сигнали є аналоговими, чи то є сигнал постійного або змінного струму, цифровий або імпульсний. Тим не менш прийнято розрізняти аналогові та цифрові сигнали. Це виражається у тому, що в природі всі фізичні величини, які є вимірюваними, вважаються аналоговими сигналами. З іншого боку існують сигнали, які називають цифровими, де сигнал є певним чином оброблений, перетворений у цифри. Внаслідок такого перетворення цифровий сигнал приймає логічний формат двійкових нулів та одиниць і з метою передачі проходить етап імпульсної модуляції, внаслідок чого утворюються низькочастотні імпульсні сигнали, або *відеоімпульси*. Такі сигнали прийнято називати відеосигналами. Взагалі, термін “відеосигнал” визначає сигнал, спектр якого починається від (або біля) постійної складової і закінчується деяким кінцевим значенням, зазвичай не більше декількох мегагерц. Для імпульсної передачі повідомлень по реальним лініям зв’язку використовується додаткова модуляція відеоімпульсами високочастотної гармонічної несучої, внаслідок чого гармонічні коливання набувають вигляду короткочасних *радіоімпульсів*, характеристики яких визначаються формою модулюючого відеоімпульсу.

Головною метою перетворення та обробки фізичних сигналів є необхідність отримання інформації, що в них міститься. Ще одна причина обробки сигналів полягає у стисненні смуги частот сигналу (без суттєвих втрат інформації) для її узгодження зі смугою пропускання каналу зв'язку. У випадку цифрової обробки сигналу аналоговий сигнал перетворюється у двійкову форму за допомогою аналого-цифрового перетворювача (АЦП). На виході АЦП отримується двійкове представлення аналогового сигналу, яке потім обробляється цифровим сигнальним процесором (ЦСП). Після обробки інформація, що міститься у сигналі, може бути перетворена назад в аналогову форму за допомогою цифро-аналогового перетворювача (ЦАП).

ДИСКРЕТИЗАЦІЯ АНАЛОГОВИХ СИГНАЛІВ

Аналоговий сигнал та його цифрова версія пов'язані процесом, який називається *дискретизацією*. В ході цього процесу з неперервного сигналу формується послідовність вибірок. Вибірка неперервних аналогових даних повинна здійснюватись через інтервал дискретизації $\Delta t = 1/\nu_d$, який необхідно ретельно підбирати для точного представлення первинного аналогового сигналу. Зрозуміло, що чим більша кількість відліків (більш високі частоти дискретизації), тим точнішим буде представлення сигналу в цифровому вигляді, тоді як у випадку малого числа відліків (низькі частоти дискретизації) може бути досягнуто критичне значення частоти дискретизації, при якому втрачається інформація про сигнал. Це випливає з відомого критерію Найквіста, який вимагає, щоб частота дискретизації була принаймні вдвічі більшою за смугу сигналу, інакше інформація про сигнал буде втрачена. Якщо частота дискретизації менше подвійної смуги аналогового сигналу, виникає ефект, відомий як *накладання спектрів* (*aliasing*).

Для розуміння накладання спектрів, як у часовій, так і у частотній областях, спочатку розглянемо випадок представлення у часовій області вибірки одного тонального сигналу синусоїдальної форми, що показано на рисунку 1. В цьому прикладі частота дискретизації ν_d лише трохи більше за частоту аналогового вхідного сигналу ν_s, що не задовольняє критерію Найквіста. З рисунку 1 добре видно, що зроблена вибірка відповідає сигналу, частота якого є набагато меншою за частоту вхідного сигналу і насправді дорівнює різниці частоти дискретизації та частоти вхідного сигналу $\nu_d - \nu_s$. Відповідне представлення цього прикладу в частотній області показано на рисунку 2В.

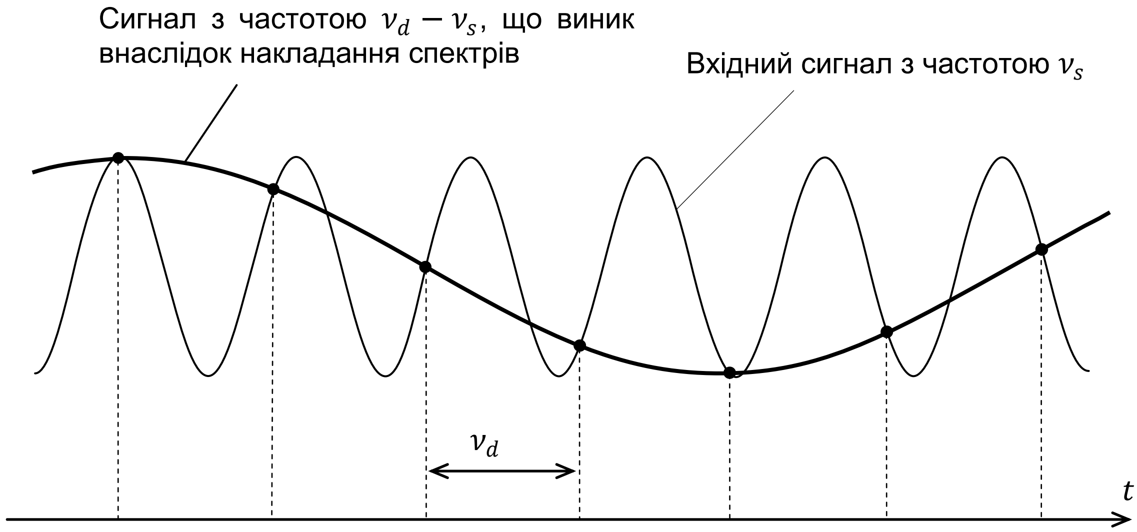


Рис. 1. Ефект накладання спектрів у часовій області.

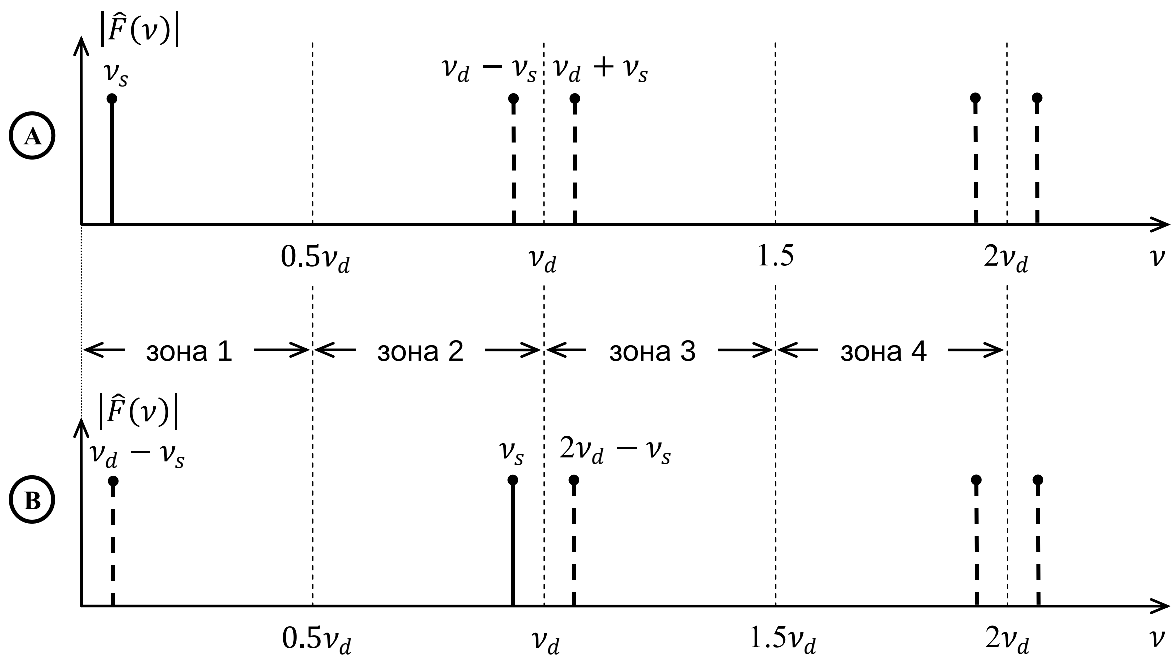


Рис. 2. Спектри сигналу з частотою ν_s після дискретизації з частотою ν_d.

Якщо ж розглянути випадок, коли $\nu_d > 2\nu_s$, то після дискретизації в частотному спектрі будуть спостерігатися гармоніки (*aliases* або *images*) початкового аналогового сигналу, які повторюються з частотою ν_d, тобто на частотах, що дорівнюють $|\pm\nu_s \pm k\nu_d|$, де $k = 1, 2, 3, \ldots$ (рис. 2А).

Частотна зона Найквіста визначається як смуга спектру від 0 до $\nu_d/2$. Частотний спектр розділений на нескінчену кількість зон Найквіста, кожна по $0.5\nu_d$. В реальних цифрових пристроях, що дискретизують вхідний аналоговий сигнал і розраховують його спектр, ідеальний дискретизатор замінюється на АЦП, який використовується разом з процесором швидкого перетворення Фур'є (ШПФ). ШПФ-процесор забезпечує присутність на виході тільки компонент сигналів, частоти яких потрапляють в першу зону Найквіста, тобто в смугу від 0 до $\nu_d/2$.

Давайте знову розглянемо рисунок 2В. Зверніть увагу, не дивлячись на те, що сигнал знаходиться за межами першої зони Найквіста, його складова $\nu_d - \nu_s$ потрапляє всередину зони. Звідси, повертаючись до рисунку 2А, легко зробити висновок, що якщо небажаний сигнал з'явиться в області будь-якої з гармонік частоти ν_s, він також з'явиться і в області самої частоти ν_s, призводячи до появи побічної частотної компоненти в першій зоні Найквіста.

Такий процес подібний до роботи змішувача, що використовується для детектування аналогових сигналів. Тому перед дискретизацією сигналу здійснюється його фільтрація, що пригнічує компоненти, частоти яких знаходяться поза смугою Найквіста і після дискретизації можуть потрапити в її межі. Виходячи з цього, на практиці майже з усіма АЦП використовується ФНЧ для пригнічення небажаних сигналів.

На рисунку 3 аналоговий сигнал з максимальною частотою в спектрі, що дорівнює ν_m, попередньо фільтрується так, що нова максимальна частота ν_m' зменшується до $\nu_d/2$ або навіть сильніше. Таким чином, оскільки $\nu_d > 2\nu_m'$, на рис. 3*б* вже відсутні компоненти, що перекриваються. Потрібно відмітити, що такий метод фільтрації, при якій видаляється частина спектру, де присутнє накладання, призведе до втрати деякої частини інформації. З цієї причини частота дискретизації, ширина смуги зрізу і тип фільтру, що обираються для конкретного сигналу, не є незалежними параметрами.

Фільтри, що реалізуються на практиці, мають ненульову ширину смуги для переходу між смугою пропускання та областю згасання. Ця область зветься *смугою переходу*. Для мінімізації частоти дискретизації системи бажано було б, щоб фільтри захисту від накладання спектрів мали вузьку смугу переходу. В той же час при звуження смуги переходу різко збільшується складність фільтрів та їх вартість. З іншого боку недостатня крутизна спаду фільтра може бути скомпенсована більшою частотою дискретизації АЦП. Обравши вищу частоту дискретизації (надлишкову

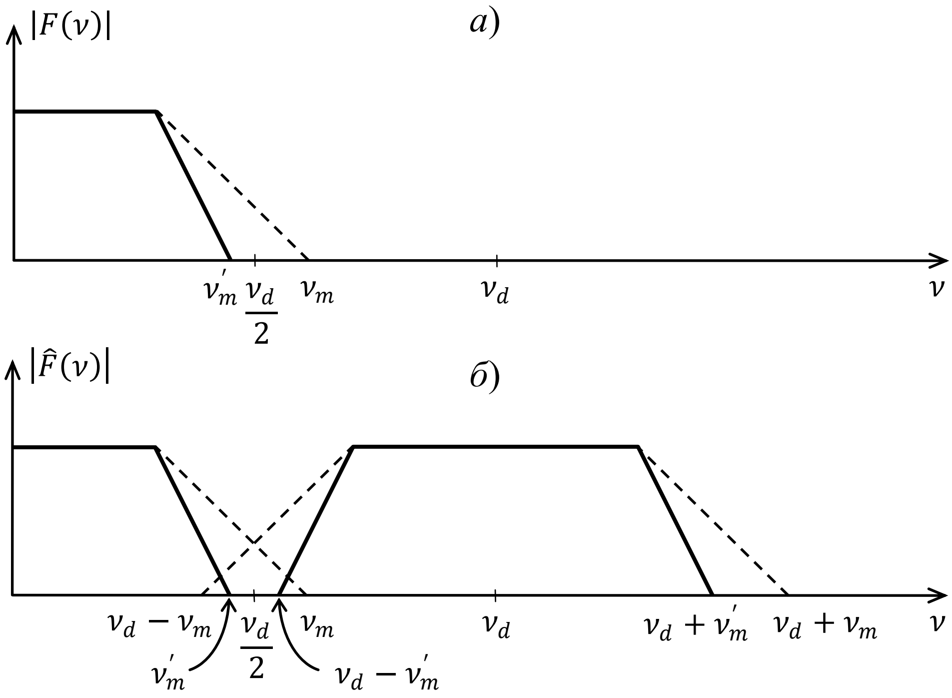


Рис. 3. Фільтрація аналогового сигналу перед дискретизацією запобігає перекриванню спектрів: *а*) спектр неперервного аналогового сигналу; *б*) спектр дискретного сигналу.

дискретизацію), ми зменшуємо вимоги до крутизни спаду фільтра і, відповідно, складність фільтру за рахунок використання більш швидкого АЦП з більшою швидкістю обробки даних. Зазвичай процес проектування фільтру починається з вибору початкової частоти дискретизації від $2.5\nu_m$ до $4\nu_m$.

СУБДИСКРЕТИЗАЦІЯ.
ДИСКРЕТИЗАЦІЯ МОДУЛЬОВАНИХ СИГНАЛІВ

До цього часу ми розглядали випадок дискретизації низькочастотних сигналів (відеосигналів або огинаючих), коли усі сигнали, що нас цікавлять, лежать у першій зоні Найквіста. На рисунку 4А показаний випадок, коли смуга сигналів, що підлягають дискретизації обмежена першою зоною Найквіста, а в інших зонах Найквіста містяться тільки дзеркальні частотні компоненти.

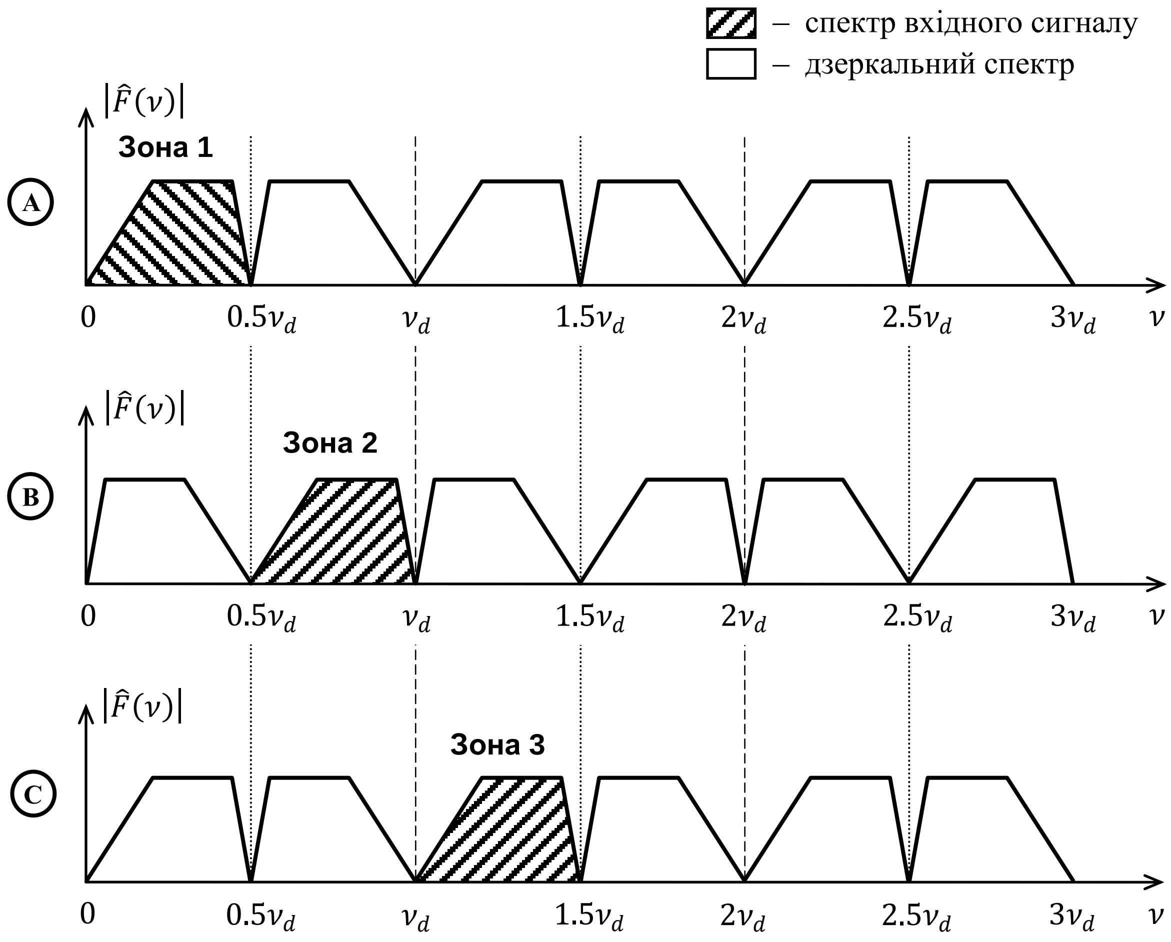


Рис. 4. Дискретизація на гармоніках.

На рисунку 4B представлений випадок, коли смуга сигналу, що підлягає дискретизації, повністю знаходиться у другій зоні Найквіста. Часто процес дискретизації сигналу, що знаходиться поза першою зоною Найквіста, називають *субдискретизацією* (*undesampling*) або *гармонічною* *дискретизацією*. Зверніть увагу, що бічна смуга в першій зоні Найквіста містить всю інформацію про первісний сигнал, проте порядок частотних компонентів в спектрі зворотній, що легко можна скоректувати шляхом перевпорядкування спектральних компонентів на виході ШПФ.

На рисунку 4С показаний варіант сигналу, що підлягає дискретизації, який обмежений третьою зоною Найквіста. Зверніть увагу, що при цьому в першій зоні Найквіста порядок спектральних компонентів такий самий, як і в спектрі вхідного сигналу.

Таким чином, фактично, частоти сигналів, що підлягають дискретизації, можуть знаходитись у будь-якій унікальній зоні Найквіста і смуга в першій зоні Найквіста буде точним представленням сигналу (за виключенням обернення частоти, яке відбувається, коли сигнали розташовані у парних зонах Найквіста). Тут ми можемо знову ясно сформулювати критерій Найквіста:

Для збереження інформації про сигнал частота дискретизації повинна бути рівною або більшою, ніж подвійна ширина його смуги.

Зауважимо, що у цьому формулюванні немає жодної згадки про абсолютне місцезнаходження у частотному спектрі смуги сигналів, що дискретизуються, відносно частоти дискретизації. Єдине обмеження полягає у тому, що смуга сигналів, які підлягають дискретизації, повинна бути обмежена однією зоною Найквіста.

Дискретизація сигналів, що знаходяться віще першої зони Найквіста, є досить популярною задачею, пов’язаною з телекомунікаціями, оскільки цей процес еквівалентний аналоговій демодуляції. Звичайною практикою є дискретизація сигналів проміжної частоти з подальшим використанням цифрових методів для обробки сигналу з усуненням у такий спосіб потреби в демодуляторі проміжної частоти.

Експериментальна установка

Схема експериментальної установки зображена на рисунку 5. Вона складається з генератора стандартних сигналів (ГСС), генератора модульованих сигналів (ГМС), аналого-цифрового перетворювача (АЦП), цифро-аналогового перетворювача (ЦАП), перемикача, аналізатора спектрів (АС). та цифрового осцилографа (ЦО) з можливістю відображення спектрів вхідних сигналів. Генератор модульованих сигналів використовує генератор стандартних сигналів як джерело зовнішньої модуляції сигналу несучої. Сигнал з виходу ГМС подається на перемикач, два положення якого забезпечують або безпосереднє дослідження вихідного аналогового сигналу ГМС, або дослідження цього ж сигналу після проходження процедури оцифровки та відновлення аналогового сигналу з цифрового коду за допомогою АЦП-ЦАП. Дослідження вихідного сигналу здійснюється за допомогою АС та ЦО. АЦП-ЦАП тактуються внутрішнім генератором, що забезпечує певну частоту дискретизації вхідного сигналу.

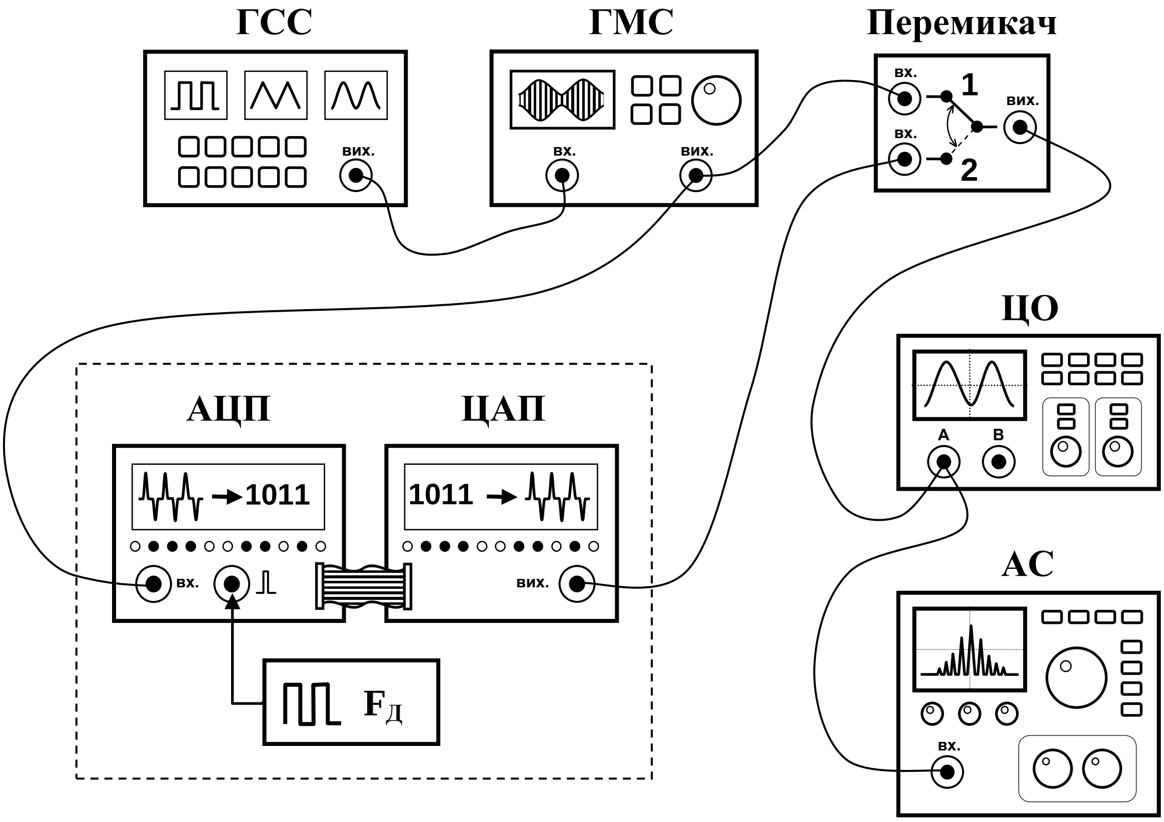


Рис. 5. Схема експериментальної установки.

Завдання

1. Вивчити експериментальну установку для проведення досліджень.
2. Освоїти роботу аналогового аналізатора спектрів, а також методику отримання спектрів за допомогою цифрового осцилографа.
3. Зібрати експериментальну установку у відповідності зі схемою, зображеною на рисунку 5.
4. Використовуючи генератор модульованих сигналів, генератор спеціальних сигналів та аналізатор спектрів, отримати та зарисувати форму та спектр радіосигналу синусоїдальної форми.
5. Здійснити перетворення амплітудно-модульованого сигналу за схемою АНАЛОГ-КОД-АНАЛОГ, використовуючи систему АЦП-ЦАП.
6. Отримати спектри сигналу з виходу ЦАП за допомогою аналізатора спектрів та цифрового осцилографа та зарисувати їх. Пояснити, в чому полягає відмінність цих спектрів від спектру радіосигналу, отриманого в пункті 4.
7. Дослідити зміну спектру амплітудно-модульованого сигналу з виходу ЦАП в залежності від частоти несучої.
8. За допомогою зміни частоти несучої визначити частоту дискретизації вхідного сигналу в АЦП.
9. Визначити в якій зоні Найквіста знаходиться основний спектр радіосигналу при заданій частоті дискретизації.
10. Зробити аналіз отриманого спектру з визначенням параметрів модулюючого сигналу та обґрунтувати вибір частоти несучої та частоти дискретизації достатньої для правильного відтворення форми модулюючого сигналу.
11. Здійснити перетворення радіосигналу прямокутної форми за схемою АНАЛОГ-КОД-АНАЛОГ, отримати спектр сигналу з виходу ЦАП та визначити мінімальну тривалість прямокутних імпульсів, яка можлива при даній частоті дискретизації.
12. Зробити висновки по роботі та оформити звіт.

Контрольні питання

1. У чому полягає відмінність між відео- та радіосигналами?
2. Який ефект виникає при порушенні критерію Найквіста в процесі дискретизації?
3. Що таке частотна зона Найквіста?
4. Чим відрізняються спектри сигналу у сусідніх зонах Найквіста?
5. Чому на практиці при аналого-цифровому перетворенні попередньо сигнал пропускають через ФНЧ?
6. Що таке субдискретизація? В якому випадку процес дискретизації сигналу називають субдискретизацією?
7. В якому випадку доцільно застосовувати субдискретизацію аналогових сигналів?
8. З якою мінімальною частотою можна дискретизувати амплітудно-модульований сигнал для достовірного та однозначного відтворення інформації, що закладена у модулюючому сигналі?

Лабораторна робота № 2

ЕФЕКТИВНЕ ЗАВАДОСТІЙКЕ КОДУВАННЯ ІНФОРМАЦІЇ В ЦИФРОВИХ КАНАЛАХ ЗВ'ЯЗКУ

Мета роботи: Вивчити методи ефективного кодування інформації. Вивчити можливості кодів щодо корекції помилок. Отримати практичні навички роботи з ефективними та завадостійкими кодами.

Литература:

1. Аршинов Н. М., Садовский Л. Е. Коды и математика. – М.: Наука, 1983.
2. Золоторев В. В. Помехоустойчивое кодирование. – М.: Горячая линия, 2004.
3. Дж. Кларк. Кодирование с исправлением ошибок в системах цифровой связи. – М.: Радио и связь, 1987.
4. Скляр Б. Цифровая связь. Теоретические основы и практическое применение: Пер. с англ. – М.: Издательский дом «Вильямс», 2003.
5. Прокис Дж. Цифровая связь. Пер. с англ. / Под ред. Д. Д. Кловского. – М.: Радио и связь, 2000.

Теоретичні відомості

МЕТОДИ ЕФЕКТИВНОГО КОДУВАННЯ

НЕРІВНОМІРНІ ЕФЕКТИВНІ КОДИ. ВЛАСТИВІСТЬ ПРЕФІКСНОСТІ

Нехай деяке джерело повідомлень має алфавіт $(a_1, a_2, \dots, a_k)$ з ймовірностями $P(a_1), P(a_2), \dots, P(a_k)$. Кожне повідомлення джерела повинно бути представлене кодовим словом, яке складається з послідовності символів, що належать заданому кодовому алфавіту. Позначимо через D кількість різних символів в кодовому алфавіті

($D = \{0, 1\}$ для двійкового алфавіту), а через n_k – кількість символів у кодовому слові, що відповідає a_k. Тоді середня кількість $\bar{n}$ кодових символів на одне повідомлення джерела буде дорівнювати:

$$\bar{n} = \sum_{k=1}^{K} P(a_k) n_k \,, \tag{1}$$

де K – кількість повідомлень.

Згідно закону великих чисел, якщо кодується дуже довга послідовність повідомлень джерела за допомогою описаної процедури кодування, то кількість кодових символів на одне повідомлення джерела буде з великою ймовірністю близьким до $\bar{n}$.

Для того, щоб вивчити питання про те, на скільки малим може бути $\bar{n}$ при однозначному декодуванні, розглянемо обмеження на нерівномірний код. Нехай алфавіт джерела містить шість повідомлень $(a_1, a_2, a_3, a_4, a_5, a_6)$, які передаються незалежно одне від одного з ймовірностями $P(a_1) = 0.4$, $P(a_2) = 0.3$, $P(a_3) = 0.1$, $P(a_4) = 0.08$, $P(a_5) = 0.07$, $P(a_6) = 0.05$. Сума цих ймовірностей дорівнює 1. Ентропія цього джерела дорівнює:

$$H = \sum_{i=1}^{6} P(a_i) log_2 \frac{1}{P(a_i)} \approx 2.16\,. \tag{2}$$

Щоб закодувати ці повідомлення рівномірним двійковим кодом, потрібно на кожне повідомлення використати 3 символи. У відповідності з теоремою кодування та виразом (2) для джерела ці повідомлення можна закодувати двійковими символами таким чином, що в середньому на кожне повідомлення буде витрачатися 2.16 двійкових символів. В таблиці 1 представлений один з можливих варіантів кодування К1, де найвірогіднішим повідомленням присвоюються найкоротші кодові слова.

Таблиця 1. Варіант кодування повідомлень К1

Повідомлення	$\boldsymbol{a_1}$	$\boldsymbol{a_2}$	$\boldsymbol{a_3}$	$\boldsymbol{a_4}$	$\boldsymbol{a_5}$	$\boldsymbol{a_6}$
Код	0	1	00	01	10	11

Таким чином, для передачі повідомлень a_1 та a_2, що мають сумарну ймовірність 0.7, використовується один символ, а для передачі інших чотирьох повідомлень, що мають сумарну ймовірність 0.3, – два символи. Отже середнє число символів на повідомлення:

$$\bar{n} = 0.7 \cdot 1 + 0.3 \cdot 2 = 1.3 \text{ символа.}$$

Тобто повідомлення вийшли закодованими ще економічніше, ніж це дозволяє теорема кодування. Але при цьому не забезпечується однозначність декодування. Отже, обраний код не придатний для передачі повідомлень. Дійсно, якщо прийнята послідовність символів $L = (100110100011110\ldots)$, то її у відповідності з кодом таблиці 1 можна декодувати як $(a_1, a_1, a_2, a_2, a_1, a_2, \ldots)$ або $(a_3, a_6, a_4, a_3, a_1, a_6, \ldots)$ і багатьма іншими способами.

Однозначність декодування при коді К1 можна забезпечити, якщо після кожного повідомлення передавати деякий символ, що розділяє повідомлення. В цьому випадку вже буде не двійковий код, а трійковий. Це використовується в коді Морзе, де окрім точки та тире, використовується також третій символ – "пробіл". Вочевидь, що введення розділювального символу знижує ефективність кодування.

Проте, однозначність декодування можна забезпечити, не вводячи розділювального символу, якщо будувати код таким чином, щоб він задовольняв умові, відомій під назвою "властивисть префіксу". Вона полягає у тому, що жодне кодове слово не повинно співпадати з початком ("префіксом") іншого кодового слова, наприклад, 1, 00, 011, 0101, 0100. Коди, що задовольняють цій умові, називають *префіксними кодами*. Ця властивість не виконується у коді К1.

Існує декілька алгоритмів побудови префіксних кодів. Серед них коди Шеннона-Фано та Хаффмана ближче за всіх дозволяють наблизитися до межі ефективності, що визначається ентропією.

КОДУВАННЯ МЕТОДОМ ХАФФМАНА

Існує лема, на якій базується принцип методики Хаффмана. Дана лема стверджує, що для будь-якого заданого джерела з повідомленнями $a_1, a_2, \ldots, a_k$, які мають ймовірності $P(a_1), P(a_2), \ldots, P(a_k)$, такі, що

$P(a_1) \geq P(a_2) \geq \ldots \geq P(a_k)$, існує оптимальний двійковий код, що утворює множину $(x_1, x_2, \ldots, x_k)$ двійкових кодових слів, в якій два найменш ймовірних кодових слова x_k та x_{k-1} мають одну і ту саму довжину та відрізняються лише останнім символом (наприклад, x_k закінчується на 1, а x_{k-1} – на 0).

За допомогою цієї леми задача побудови оптимального коду зводиться до задачі побудови кодових слів $x_1, x_2, \ldots, x_{k-2}$ та пошуку перших $n_k - 1$ символів кодового слова x_k. Визначимо зменшений ансамбль A, який складається з повідомлень $a_1, a_2, \ldots, a_k$, як ансамбль A^*, що складається з повідомлень $a_1^{'}, a_2^{'}, \ldots, a_{k-1}^{'}$, які мають ймовірності:

$$P(a_m^{'}) = \begin{cases} P(a_m), & m \leq k-2, \\ P(a_{k-1}) + P(a_k), & m = k-1. \end{cases} \quad (3)$$

Таким чином, будь-який префікс ний код для A^* можна перетворити у відповідний код для A додаванням кінцевого символу "0" до x'_{k-1} для отримання x_{k-1} та додавання кінцевого символу "1" до x'_k для отримання x_k. Звідси випливає, що задача пошуку оптимального коду зводиться до задачі пошуку оптимального коду для зменшеного ансамблю, що має на одне повідомлення менше. Даний ансамбль може мати власні два найменш ймовірні повідомлення, що можуть бути згруповані разом, і, отже, можна буде створити новий зменшений ансамбль. Продовжуючи далі, таким чином, можна в результаті отримати ансамбль, що буде складатися лише з двох повідомлень, і тоді оптимальний код утворюється присвоєнням символу "1" одному повідомленню та символу "0" іншому.

Найзручніше та наочніше операції кодування по алгоритму Хаффмана можна проводити з використанням кодового дерева. Розглянемо приклад (таблиця 2). Для знаходження кодової комбінації, що відповідає і-му знаку, необхідно прослідкувати шлях переходу знака по рядкам та стовбцям таблиці. З точки, що відповідає ймовірності 1, спрямовуються 2 гілки, причому гілці з більшою ймовірністю присвоюється символ "1", а з меншою – "0". Таке розгалуження продовжується до тих пір, доки не дійдемо до ймовірності кожної букви.

Кодове дерево для алфавіту повідомлень табл. 2 наведено на рис. 1. Рухаючись вздовж кодового дерева, можна записати для кожного повідомлення кодові комбінації, що йому відповідають (табл. 3).

Таблиця 2. Кодування повідомлень методом Хаффмана

Повідомлення	**Ймовірності**	**Допоміжні стовбці**						
		1	**2**	**3**	**4**	**5**	**6**	**7**
A1	0.22	0.22	0.22	0.26	0.32	0.42	0.58	1.00
A2	0.20	0.20	0.20	0.22	0.26	0.32	0.42	
A3	0.16	0.16	0.16	0.20	0.22	0.26		
A4	0.16	0.16	0.16	0.16	0.20			
A5	0.10	0.10	0.16	0.16				
A6	0.10	0.10	0.10					
A7	0.04	0.06						
A8	0.02							

Таблиця 3. Кодові комбінації коду Хаффмана

Повідомлення	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
Код	01	00	111	110	100	1011	10101	10100

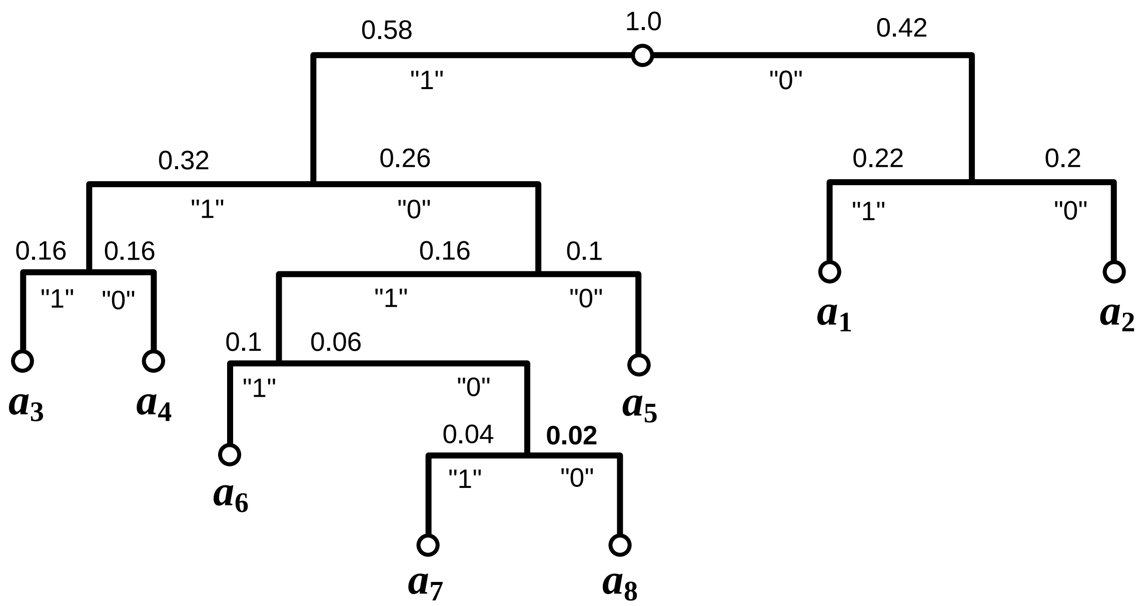


Рис. 1. Кодове дерево Хаффмана

АРИФМЕТИЧНЕ КОДУВАННЯ

У 70-ті роки у алгоритму Хаффмана з’явилася альтернатива, яка виправляє усі основні недоліки цього алгоритму, – арифметичне кодування. Цей метод оснований на ідеї перетворення вхідного потоку в одне число з плаваючою крапкою. Очевидно, що чим довшим буде повідомлення, тим довшим буде число, яке отримується у результаті кодування. Таким чином, на виході арифметичного компресора буде число, яке є меншим за 1 та є більшим або дорівнює 0. З цього числа можна однозначно відновити послідовність символів, з яких воно було побудовано.

В загальному вигляді алгоритм арифметичного кодування може бути описаний наступним чином. Початковий ймовірнісний інтервал знаходиться в межах від 0 до 1. Усі можливі символи займають місця в цьому ймовірнісному інтервалі згідно таблиці, що заздалегідь задає інтервали та їх межі відносно ймовірнісного відрізку одиничної довжини. Відрізок чергового символу, що кодується, буде новим інтервалом, в межах якого буде проведено ділення на підінтервали згідно із заданою на початку таблицею. Так буде продовжуватися до тих пір, доки не буде досягнутий кінець вхідного потоку даних.

Розглянемо загальні етапи узагальненого алгоритму арифметичного кодування:

1. Розрахунок ймовірностей усіх символів, що використовуються.
2. Розподіл інтервалів ймовірностей.
3. Розрахунок нижньої та верхньої межі кожного наступного символу.
4. Розрахунок інтервалів для наступних символів базується на розрахунку його меж в рамках інтервалу, що отриманий відносно попереднього символу, а потім отримання числа в рамках усього ймовірнісного інтервалу.

Розглянемо приклад. Спробуємо за допомогою алгоритму арифметичного кодування закодувати слово "КАСКА". Алфавіт символів цього слова складається з трьох елементів: "А", "К", "С". Ймовірності появи символів розрахуємо як частоту появи їх у послідовності, що кодується. Символ "С" з’являється у слові "КАСКА" 1 раз, у той час як символи "А" і "К" з’явилися по 2 рази. Таким чином, ймовірності появи символів "А", "К" та "С" будуть дорівнювати 0.4, 0.4 та 0.2 відповідно.

Тепер необхідно розподілити ймовірнісні інтервали символів на інтервалі від 0 до 1, заповнюючи тим самим допоміжну таблицю арифметичного кодування (див. табл. 4). Запишемо алфавіт символів в довільному порядку у перший стовпчик таблиці (нехай це буде порядок, в якому ці літери з’являються у українському алфавіті: "А", потім "К", потім "С"). Нижньою межею символу "А" буде нижня межа загального інтервалу, тобто 0. Символ "А" займає 40% (0.4) від загального інтервалу, тому його верхньою межею буде число 0.4. Символ "К" займає наступні 40% інтервалу (від 0.4 до 0.8) і залишок інтервалу буде визначений для символу "С". Таким чином, допоміжна таблиця буде повністю заповнена.

Таблиця 4. Допоміжна таблиця арифметичного кодування

Символ	**Ймовірність**	**Нижня межа**	**Верхня межа**
А	0.4	0	0.4
К	0.4	0.4	0.8
С	0.2	0.8	1

Допоміжна таблиця в даному прикладі показує те, що кожен раз інтервал символу "А" у вхідному потоці буде займати відрізок від 0 до 40% ***всередині попереднього інтервалу***. Символ "К" буде займати відрізок від 40% до 80% у попередньому інтервалі, а символ "С" – останні 20% попереднього інтервалу.

Тепер будемо заповнювати основну таблицю арифметичного кодування. Для цього будемо вказувати інтервал та його межі для кожного символу з потоку, що надходить. Першим у потоці йде символ "К". Його підінтервал та межі беруться в рамках початкового відрізку (0, 1), а отже будуть співпадати з підінтервалом у початковій таблиці – 0.4 – 0.8. Наступним з вхідного потоку зчитується символ "А". Він буде займати перші 40% попереднього відрізку, а саме 0.4 – 0.56. Наступний символ "С" буде займати останні 20% попереднього відрізку, а саме 0.528 – 0.56. Наступний символ "К" займатиме від 40% до 80% попереднього інтервалу, що у числовому вигляді буде дорівнювати 0.5408 – 0.5536 (число 0.5408 – це 40-відсоткова межа інтервалу 0.528 – 0.56, а число 0.5536 – це 80-відсоткова межа того ж інтервалу) і так далі (див. таблицю 5). Розглянутий процес кодування можна представити графічно (рис. 2).

Таблиця 5. Таблиця арифметичного кодування

Символ	Нижня межа	Верхня межа	Інтервал
К	0.4	0.8	0.4
А	0.4	0.56	0.16
С	0.528	0.56	0.032
К	0.5408	0.5536	0.0128
А	0.5408	0.54592	0.00512

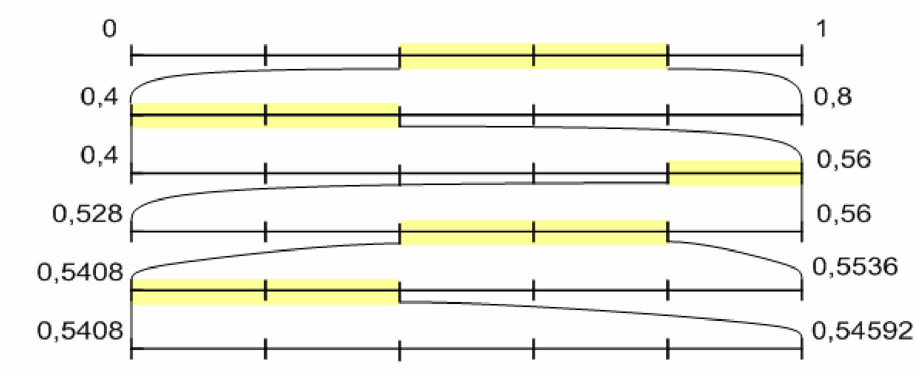


Рис. 2. Пояснення процесу арифметичного кодування

Таким чином, число з інтервалу (0.5408, 0.54592) буде однозначно відповідати слову "КАСКА". Кінцевий результат залежить від розміру алфавіту, що використовується, та від результатів ймовірнісних оцінок його складових.

Процес арифметичного декодування здійснюється в зворотному порядку. Для прикладу розглянемо алфавіт, який складається з чотирьох символів: "плюс", "мінус", знак "рівно" та знак кінця вхідного потоку. Припустимо, що число, яке потрібно декодувати, дорівнює 0.538. Кодер та декодер повинні чітко знати таблицю розподілу ймовірностей кожного з символів алфавіту. Визначимо ймовірності для вказаних символів. Нехай, наприклад, у відсотковому співвідношенні вони будуть наступними:

- 60% – ймовірність символу "рівно";
- 20% – ймовірність символу "плюс";
- 10% – ймовірність символу "мінус";
- 10% – ймовірність символу кінця вхідного потоку.

Отже, можна скласти допоміжну таблицю 6.

Таблиця 6. Допоміжна таблиця арифметичного кодування

Символ	Ймовірність	Нижня межа	Верхня межа
=	0.6	0	0.6
+	0.2	0.6	0.8
–	0.1	0.8	0.9
Кінець потоку	0.1	0.9	1

При декодуванні згідно з таблицею 6 число 0.538, що декодується, потрапляє у перший інтервал, а значить першим символом, що поступає на вихід декодера, буде символ "рівно", який займає інтервал [0, 0.6). Далі продовжимо розбиття на інтервали вже на першому відрізку:

- інтервал символу "рівно": [0, 0.36) – перші 60% від [0, 0.6);
- інтервал символу "плюс": [0.36, 0.48) – наступні 20% від [0, 0.6);
- інтервал символу "мінус": [0.48, 0.54) – наступні 10% від [0, 0.6);
- інтервал символу кінця потоку: [0.54, 0.6) – останні 10% від [0, 0.6).

Звідси можна побачити, що число 0.538 потрапляє тепер у третій ймовірнісний підінтервал [0.48, 0.54). Отже наступним символом на виході декодера буде символ "мінус".

Чергове ділення на інтервали відбувається на третьому відрізку поточного інтервалу і буде виглядати наступним чином:

- інтервал символу "рівно": [0.48, 0.516) – перші 60% від [0.48, 0.54);
- інтервал символу "плюс": [0.516, 0.528) – наступні 20% від [0.48, 0.54);
- інтервал символу "мінус": [0.528, 0.534) – наступні 10% від [0.48, 0.54);
- інтервал символу кінця потоку: [0.534, 0.54) – останні 10% від [0.48, 0.54).

Звідси видно, що число 0.538 знаходиться в інтервалі символу кінця вхідного потоку, а, отже, цей символ буде останнім символом, що надійде на вихід декодера, для заданого числа.

Розглянутий процес арифметичного декодування можна також представити графічно (рис. 3).

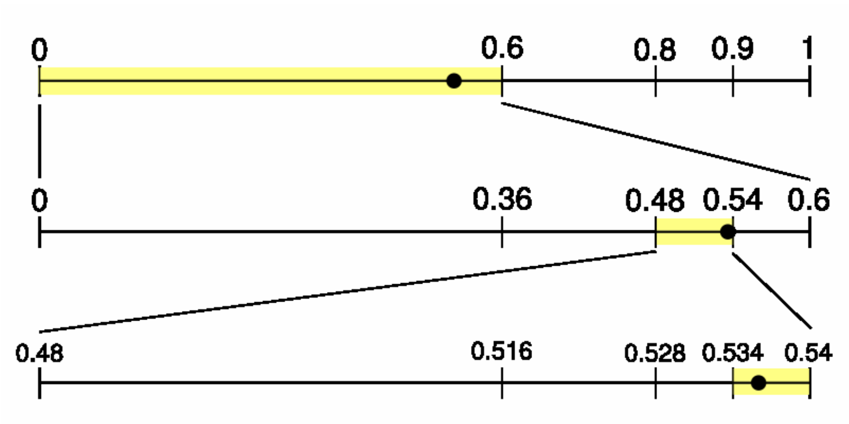


Рис. 3. Пояснення до процесу арифметичного декодування числа 0.538

КОРЕКТУЮЧІ КОДИ

ПРИНЦИПИ ЗАВАДОСТІЙКОГО КОДУВАННЯ

У реальних умовах прийом двійкових символів завжди відбувається з помилками, коли замість символу "1" приймається символ "0" і навпаки. Помилки можуть виникати через завади, що діють в каналі зв'язку (особливо завади імпульсного характеру), зміни за час передачі характеристик каналу (наприклад, завмирання), зниження рівня передачі, нестабільності амплітудно- і фазочастотних характеристик каналу і тому подібне.

Загальноприйнятим критерієм оцінки якості передачі в дискретних каналах є нормована на знак або символ допустима ймовірність помилки для даного виду повідомлень. Так, допустима ймовірність помилки при телеграфному зв'язку може складати 10^{-3} (на знак), а при передачі даних – не більше 10^{-6} (на символ). Для забезпечення таких значень ймовірності поліпшення лише якісних показників каналу зв'язку може виявитися недостатнім. Тому основною мірою є вживання спеціальних методів підвищення якості прийому інформації, що передається. Ці методи можна розбити на дві групи.

До першої групи відносяться методи збільшення завадостійкості прийому одиничних елементів (символів) дискретної інформації, пов'язані з вибором рівня сигналу, співвідношення сигнал-завада (енергетичні характеристики), ширини смуги каналу, методів прийому і так далі.

До другої групи відносяться методи виявлення та виправлення помилок, засновані на штучному введенні надлишковості в повідомлення, що передається. Збільшити надлишковість сигналу, що передається, можна різними способами. Оскільки об'єм сигналу

$$V = W \cdot \Delta F \cdot T, \tag{4}$$

де W – потужність сигналу, Вт; ΔF – ширина спектру сигналу, Гц; T – час передачі сигналу, сек, то його збільшення можливе за рахунок збільшення W, ΔF і T.

Практичні можливості збільшення надлишковості за рахунок потужності і ширини спектру сигналу в системах передачі дискретної інформації по стандартних каналах різко обмежені. Тому для підвищення якості прийому, як правило, йдуть по шляху збільшення часу передачі і використовують наступні основні способи:

1) багатократна передача кодових комбінацій (метод повторення);
2) одночасна передача кодової комбінації по декількох паралельно працюючих каналах;
3) завадостійке (що коректує) кодування, тобто використання кодів, що виправляють помилки.

Інколи застосовують комбінації цих способів.

Багатократне повторення (декілька разів) кодової комбінації є найпростішим способом підвищення достовірності прийому і легко реалізується, особливо в низькошвидкісних системах передачі для каналів із швидко змінними параметрами.

Способу багатократного повторення є аналогічним спосіб передачі однієї і тієї ж інформації по декількох паралельних каналах зв'язку. В цьому випадку необхідно мати не менше трьох каналів зв'язку (наприклад, з частотним рознесенням), несучі частоти яких потрібно вибирати так, щоб помилки в каналах були незалежні. Перевагою таких систем є надійність та малий час затримки в отриманні інформації. Основним недоліком багатоканальних систем так само, як і систем з повторенням, є нераціональне використання надлишковості.

Найдоцільніше надлишковість використовується при вживанні завадостійких (що коректують) кодів.

У звичайному рівномірному незавадостійкому коді кількість розрядів n в кодових комбінаціях визначається кількістю повідомлень і основою коду. Внесення надлишковості при використанні завадостійких кодів обов'язково пов'язане із збільшенням кількості розрядів n (довжини) кодової комбінації. При цьому завадостійкий код відрізняється від звичайного тим, що в канал передаються не всі кодові комбінації, які можна сформувати з наявного числа розрядів n, а лише їх частина, яка складає підмножину *дозволених комбінацій*, тобто таких, що володіють певними ознаками, водночас інша частина кодових комбінацій буде складати підмножину *заборонених комбінацій*, що не володіють цими ознаками.

Коди, в яких всі кодові комбінації є дозволеними до передачі, називаються простими або *рівнодоступними* і є повністю *безнадлишковими*. Безнадлишкові первинні коди мають велику "чутливість" до завад.

Саме наявність в коді дозволених та заборонених кодових комбінацій робить код завадостійким. Якщо при прийомі з'ясовується, що кодова комбінація належить до заборонених, то це свідчить про наявність помилок в комбінації, тобто таким чином вирішується завдання виявлення помилок. При цьому прийнята комбінація або не декодується (не приймається рішення про передане повідомлення), або аналізується з метою виправлення помилки, що виникла (якщо це передбачено властивостями коду). У зв'язку з цим завадостійкі коди ще називають кодами, що коректують. Властивості надлишкових кодів, що коректують, залежать від правила їх побудови, що визначає структуру коду, та параметрів коду (тривалості символів, кількості розрядів, надлишковості і тому подібне).

При завадостійкому кодуванні найчастіше вважають, що надлишковість джерела повідомлень на вході кодера дорівнює нулю. Це обумовлено тим, що дуже багато дискретних джерел (наприклад, цифрова інформація на виході ЕОМ) володіють малою надлишковістю. Якщо надмірність первинних джерел повідомлень істотна, то в цих випадках по можливості прагнуть її зменшити шляхом ефективного кодування, застосовуючи, наприклад, коди Шеннона-Фано або Хаффмана. Потім методами завадостійкого кодування можна внести таку надмірність в сигнал, яка дозволить досить простими засобами поліпшити якість прийому. Таким чином, ефективне кодування сповна може поєднуватися з завадостійким.

ОСНОВНІ ХАРАКТЕРИСТИКИ КОРЕКТУЮЧИХ КОДІВ

В даний час найбільша увага з точки зору технічних застосувань приділяється двійковим блоковим кодам, що коректують. При використанні блокових кодів цифрова інформація передається у вигляді окремих кодових комбінацій (блоків) рівної довжини. Кодування і декодування кожного блоку здійснюється незалежно один від одного.

Майже всі блокові коди відносяться до роздільних кодів, кодові комбінації яких складаються з двох частин: інформаційної та перевірочної. При загальній кількості n символів в блоці кількість інформаційних символів дорівнює k, а кількість перевірочних символів

$$r = n - k, \tag{5}$$

До основних характеристик кодів, що коректують, відносяться:

- кількість дозволених та заборонених кодових комбінацій;
- надлишковість коду; мінімальна кодова відстань;
- кількість помилок, що виявляються або виправляються;
- можливості коду щодо корекції.

Кількість дозволених та заборонених кодових комбінацій.

Для блочних кодів з кількістю символів в блоках, що дорівнює n, загальна кількість можливих комбінацій визначається значенням

$$N_0 = 2^n. \tag{6}$$

Кількість дозволених кодових комбінацій при наявності k інформаційних розрядів в первинному коді дорівнює

$$N_k = 2^k. \tag{7}$$

Вочевидь, що кількість заборонених комбінацій дорівнює

$$N_з = N_0 - N_k = 2^n - 2^k, \tag{8}$$

а з врахуванням (5) співвідношення буде:

$$N_0/N_k = 2^n/2^k = 2^{n-k} = 2^r. \tag{9}$$

Надлишковість коректуючого коду.

Надлишковістю коректуючого коду називають величину

$$\chi = \frac{r}{n} = \frac{n-k}{n} = 1 - \frac{n}{k}, \tag{10}$$

звідки випливає:

$$B_k = \frac{k}{n} = 1 - \chi\,. \tag{11}$$

Ця величина показує, яку частину загальної кількості символів кодової комбінації складають інформаційні символи. У теорії кодування величину B_k називають *відносною швидкістю коду*. Якщо продуктивність джерела інформації дорівнює H символів за секунду, то швидкість передачі після кодування цієї інформації виявиться рівною

$$B = H \cdot \frac{k}{n}, \tag{12}$$

оскільки в закодованій послідовності з кожних n символів лише тільки k символів є інформаційними.

Якщо кількість помилок, які потрібно виявити або виправити, є значною, то необхідно мати код з великою кількістю перевірочних символів. Щоб при цьому швидкість передачі залишалася досить високою, необхідно в кожному кодовому блоці одночасно збільшувати як загальну кількість символів, так і кількість інформаційних символів. При цьому тривалість кодових блоків істотно зростатиме, що призведе до затримки інформації при передачі та прийомі. Чим складніше кодування, тим тривалішою буде часова затримка інформації.

Мінімальна кодова відстань.

Для того, щоб можна було виявляти та виправляти помилки, дозволена комбінація повинна якомога більше відрізнятися від забороненої. Якщо помилки в каналі зв'язку діють незалежно, то ймовірність перетворення однієї кодової комбінації в іншу буде тим меншою, чим більшою кількістю символів вони різняться.

Якщо інтерпретувати кодові комбінації як крапки в просторі, то відмінність виражається в близькості цих крапок, тобто у відстані між ними.

Кількість розрядів (символів), якими відрізняються дві кодові комбінації, можна прийняти за кодову відстань між ними. Для визначення цієї відстані потрібно скласти дві кодові комбінації по модулю 2 і підрахувати кількість одиниць в отриманій сумі. Відмітимо також, що кількість одиниць в кодовій комбінації визначає *вагу* W комбінації або, інакше кажучи, вага комбінації це кодова відстань $d(x_i, x_0)$ між комбінацією x_i та нульовою комбінацією $x_0 = 00 \dots 0$. Наприклад, дві кодові комбінації $x_i = 01011$ та $x_j = 10010$ мають відстань $d(x_i, x_j)$, що дорівнює 3, оскільки

$$\begin{array}{rl} x_i = 01011 & \rightarrow\ W = 3 \\ \oplus & \\ x_j = 10010 & \rightarrow\ W = 2 \\ \hline x_i \oplus x_j = 11001 & \rightarrow\ d(x_i, x_j) = 3 \end{array}$$

(Тут під операцією "⊕" розуміється додавання по модулю 2).

Відстані між різними комбінаціями деякого конкретного коду можуть істотно відрізнятися. Так, зокрема, в безнадлишковому первинному натуральному коді $(n = k)$ ця відстань для різних комбінацій може змінюватися від одиниці до величини n. Особливу важливість для характеристики властивостей коду щодо корекції має *мінімальна кодова відстань* d_{min}, що визначається при попарному порівнянні всіх кодових комбінацій, яку називають відстанню Хеммінга.

У безнадлишковому коді всі комбінації є дозволеними, а, отже, його мінімальна кодова відстань дорівнює одиниці – $d_{min} = 1$. Тому досить спотворитися одному символу, щоб замість переданої комбінації була прийнята інша дозволена комбінація. Для того, щоб код володів властивостями виявлення помилок, необхідно ввести в нього деяку надлишковість, яка забезпечувала б мінімальну відстань між будь-якими двома дозволеними комбінаціями не менше двох – $d_{min} \geq 2$.

Мінімальна кодова відстань є найважливішою характеристикою завадостійких кодів, що вказує на гарантовану кількість помилок, що виявляються або виправляються заданим кодом.

Кількість помилок, що виявляється або виправляється.

При застосуванні двійкових кодів враховують лише дискретні спотворення, при яких одиниця переходить в нуль ($1 \to 0$) або нуль переходить в одиницю ($0 \to 1$). Перехід $1 \to 0$ або $0 \to 1$ лише в одному елементі кодової комбінації називають *одиничною помилкою* (одиничним спотворенням). У загальному випадку під *кратністю помилки* мають на увазі кількість позицій кодової комбінації, на яких під дією завади одні символи виявилися заміненими на інші. Можливі двократні ($g = 2$) та багатократні ($g > 2$) спотворення елементів в кодовій комбінації в межах $0 < g < n$.

Мінімальна кодова відстань є основним параметром, що характеризує здатність даного коду, до корекції. Якщо код використовується лише для виявлення помилок кратністю g_0, то необхідно і достатньо, щоб мінімальна кодова відстань задовольняла умові

$$d_{min} \geq g_0 + 1. \tag{13}$$

В цьому випадку жодна комбінація з g_0 помилок не здатна перевести одну дозволену кодову комбінацію в іншу дозволену. Таким чином, умова виявлення усіх помилок кратністю g_0 може бути записана у вигляді:

$$g_0 \leq d_{min} - 1. \tag{14}$$

Для того, щоб було можливим виправляти усі помилки кратністю g_u та менше, необхідно мати мінімальну відстань, що задовольняє умові:

$$d_{min} \geq 2g_u + 1. \tag{15}$$

В цьому випадку будь-яка кодова комбінація з кількістю помилок g_u відрізняється від кожної дозволеної комбінації не менше ніж у $g_u + 1$ позиціях. Якщо умова (15) не виконана, можливий випадок, коли помилки кратності g спотворять передану комбінацію так, що вона стане ближча до однієї з інших дозволених комбінацій, ніж до переданої комбінації, або навіть перейде в іншу дозволену комбінацію. Відповідно до цього, умова виправлення всіх помилок кратністю не більш g_u може бути записана у вигляді:

$$g_u \le \frac{d_{min} - 1}{2}. \tag{16}$$

З (13) та (15) випливає, що якщо код виправляє усі помилки кратністю g_u, то кількість помилок, які він може виявити, дорівнює $g_0 = 2g_u$. Потрібно зазначити, що ці співвідношення встановлюють лише гарантовану мінімальну кількість помилок, що виявляються або виправляються при заданому d_{min}, і не обмежують можливість виявлення помилок більшої кратності. Наприклад, найпростіший код з перевіркою на парність з $d_{min} = 2$ дозволяє виявляти не тільки поодинокі помилки, але і будь-яку непарну кількість помилок в межах $g_0 < n$.

Можливості кодів щодо корекції.

Питання про мінімально необхідну надлишковість, при якій код володіє потрібними властивостями щодо корекції, є одним з найважливіших в теорії кодування. Це питання до цих пір не отримало повної відповіді. В даний час отримана лише низка верхніх та нижніх оцінок (границь), які встановлюють зв'язок між максимально можливою мінімальною відстанню коду, що коректує, та його надлишковістю.

Так, границя Плоткіна дає верхню межу кодової відстані d_{min} при заданій кількості розрядів n в кодовій комбінації та кількості інформаційних розрядів k, і для двійкових кодів вона визначається співвідношенням:

$$d_{min} \le \frac{n \cdot 2^{k-1}}{2^k - 1} \tag{17}$$

або $r \ge 2 \cdot (d_{min} - 1) - \log_2 d_{min}$ при $n \ge 2 \cdot d_{min} - 1$.

Верхня границя Хеммінга встановлює максимально можливу кількість дозволених кодових комбінацій (2^k) будь-якого завадостійкого коду при заданих значеннях n та d_{min}:

$$2^k \leq 2^n \Bigg/ \sum_{i=0}^{\frac{d_{min}-1}{2}} C_n^i \, , \tag{18}$$

де C_n^i – кількість сполучень з n елементів по i елементам. Звідси можна отримати вираз для оцінки кількості перевірочних символів:

$$r \geq \log_2\left(\sum_{i=0}^{\frac{d_{min}-1}{2}} C_n^i \right). \tag{19}$$

Для значень $d_{min}/n < 0.3$ різниця між границею Хеммінга та границею Плоткіна порівняно невелика.

Границя Варшамова-Гільберта для великих значень n визначає нижню межу для кількості перевірочних розрядів, що необхідна для забезпечення заданої кодової відстані:

$$r \geq \log_2\left(\sum_{i=0}^{d_{min}-2} C_{n-1}^i \right). \tag{20}$$

Потрібно відзначити, що для деяких частинних випадків Хеммінг отримав прості співвідношення, що дозволяють визначити необхідну кількість перевірочних символів:

$$r \geq \log_2(n+1) \quad \text{для} \quad d_{min} = 3,$$
$$r \geq \log_2(2 \cdot n) \quad \text{для} \quad d_{min} = 4.$$

Блочні коди з $d_{min} = 3$ та $d_{min} = 4$ в літературі зазвичай називають кодами Хеммінга.

Всі наведені вище оцінки дають уявлення про верхню межу числа d_{min} при фіксованих значеннях n та k або оцінку знизу кількості перевірочних символів r при заданих k та d_{min}.

Існуючі методи побудови надлишкових кодів вирішують в основному задачу знаходження такого алгоритму кодування й декодування, який дозволяв би найпростіше побудувати та реалізувати код із заданим значенням d_{min}. Тому різні коди, що коректують, при однакових d_{min} порівнюються по складності кодуючого та декодуючого пристроїв. Цей критерій у ряді випадків є визначним при виборі того або іншого коду.

КОРЕКТУЮЧІ КОДИ ХЕММІНГА

Побудова коду Хеммінга базується на принципі перевірки на парність ваги W в інформаційній групі кодового блоку. Розглянемо ідею перевірки на парність на прикладі простого коду, що коректує, який так і називається *кодом з перевіркою на парність* або *кодом з перевіркою по паритету* (рівності).

У такому коді до кодових комбінацій безнадлишкового первинного двійкового k-розрядного коду додається один додатковий розряд (символ перевірки на парність, що називається перевірочним або контрольним). Якщо кількість символів "1" початкової кодової комбінації є парним числом, то в додатковому розряді формують контрольний символ "0", а якщо кількість символів "1" є непарним числом, то в додатковому розряді формують символ "1". В результаті загальна кількість символів "1" в будь-якій кодовій комбінації, що передається, завжди буде парною.

Таким чином, правило формування перевірочного символу зводиться до наступного:

$$r_1 \geq i_1 \oplus i_2 \oplus \ldots \oplus i_k \, , \tag{21}$$

де i – відповідний інформаційний символ ("0" або "1"), k – загальна їх кількість. Очевидно, що додавання додаткового розряду збільшує загальну кількість можливих комбінацій удвічі в порівнянні з кількістю комбінацій початкового первинного коду, а умова парності розділяє усі комбінації на дозволені та недозволені. Код з перевіркою на парність дозволяє виявляти поодиноку помилку при прийомі кодової комбінації, оскільки така помилка порушує умову парності, переводячи дозволену комбінацію в заборонену.

Критерієм правильності прийнятої комбінації є рівність нулю результату S підсумовування по модулю 2 всіх n символів коду, включаючи перевірочний символ r_1:

$$\underbrace{S = r_1 \oplus i_1 \oplus i_2 \oplus \ldots \oplus i_k}_{n} = \begin{cases} 0 \text{ – помилки немає} \\ 1 \text{ – однократна помилка} \end{cases}$$

Результат $S = 1$ буде свідчити про наявність поодинокої помилки.

Цей код є $(k+1, k)$ кодом або $(n, n-1)$ кодом. Мінімальна відстань коду дорівнює двом ($d_{min} = 2$), і, як наслідок, ніякі помилки не можуть бути виправлені. Простий код з перевіркою на парність може використовуватися тільки для виявлення (але не для виправлення) однократних помилок.

Збільшуючи кількість додаткових перевірочних розрядів і формуючи по певних правилах перевірочні символи r, що дорівнюють 0 або 1, можна підсилити властивості коду щодо корекції, так, щоб він дозволяв не лише виявляти, але й виправляти помилки. На цьому і заснована побудова кодів Хеммінга.

Розглянемо коди Хеммінга, що дозволяють виправляти одиночну помилку, за допомогою безпосереднього опису. Для кожної кількості перевірочних символів $r = 3, 4, 5...$ існує класичний код Хеммінга з маркуванням

$$(n, k) = (2^r - 1, 2^r - 1 - r),$$

тобто – (7, 4), (15, 11), (31, 26)…

При інших значеннях кількості інформаційних символів k отримуються, так звані, усічені (укорочені) коди Хеммінга. Так, для міжнародного телеграфного коду МТК-2, що має 5 інформаційних символів, знадобиться використання коду (9,5), що коректує, який є усіченим від класичного коду Хеммінга (15,11), оскільки кількість символів в цьому коді зменшується (коротшає) на 6.

Для прикладу розглянемо класичний код Хеммінга (7,4), який можна сформувати і описати за допомогою кодера, представленого на рисунку 4. У найпростішому варіанті при заданих чотирьох ($k = 4$) інформаційних символах (i_1, i_2, i_3, i_4) вважатимемо, що вони згруповані спочатку кодового слова, хоча це і не є обов'язковим. Доповнимо ці інформаційні символи трьома перевірочними символами ($r = 3$), задаючи їх наступними рівностями перевірки на парність:

$$
\begin{aligned}
r_1 &= i_1 \oplus i_2 \oplus i_3; \\
r_2 &= i_2 \oplus i_3 \oplus i_4; \qquad (22) \\
r_3 &= i_1 \oplus i_2 \oplus i_4.
\end{aligned}
$$

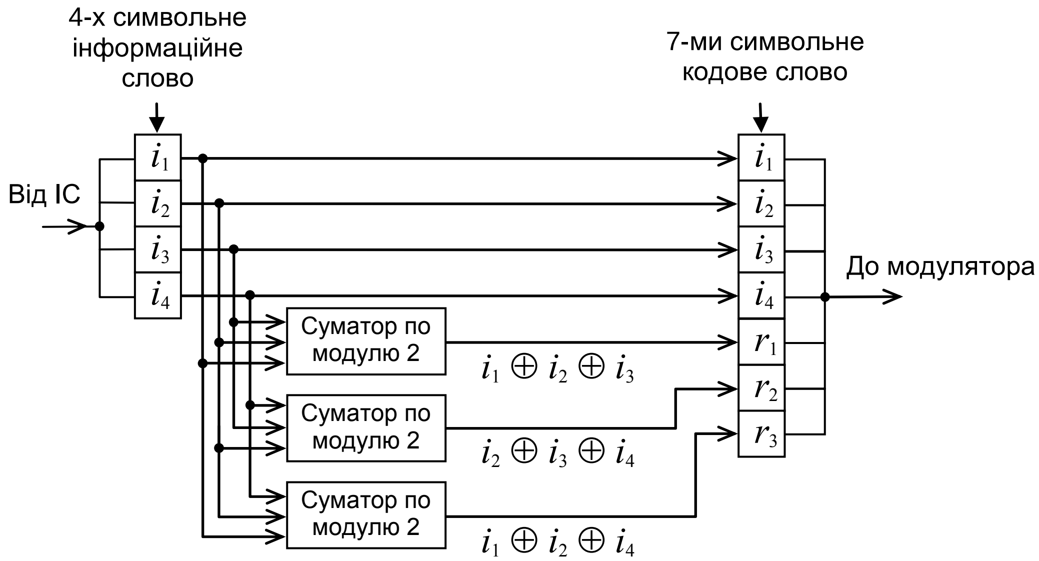


Рис. 4. Кодер для простого (7, 4) коду Хеммінга

Таблиця 7. Усі кодові слова (7, 4) коду Хеммінга

$k = 4$				$r = 3$		
i_1	i_2	i_3	i_4	r_1	r_2	r_3
0	0	0	0	0	0	0
0	0	0	1	0	1	1
0	0	1	0	1	1	0
0	0	1	1	1	0	1
0	1	0	0	1	1	1
0	1	0	1	1	0	0
0	1	1	0	0	0	1
0	1	1	1	0	1	0
1	0	0	0	1	0	1
1	0	0	1	1	1	0
1	0	1	0	0	1	1
1	0	1	1	0	0	0
1	1	0	0	0	1	0
1	1	0	1	0	0	1
1	1	1	0	1	0	0
1	1	1	1	1	1	1

У відповідності з алгоритмом визначення значень перевірочних символів r_i за допомогою (22) в таблиці 7 виписані усі можливі 16 кодових слів (7, 4) коду Хеммінга.

На рисунку 5 наведена схема декодера для (7, 4) коду Хеммінга, на вхід якого потрапляє кодове слово

$$V = (i_1^{'}, i_2^{'}, i_3^{'}, i_4^{'}, r_1^{'}, r_2^{'}, r_3^{'}).$$

Апостроф означає, що будь-який символ може бути спотворений завадою в каналі передачі.

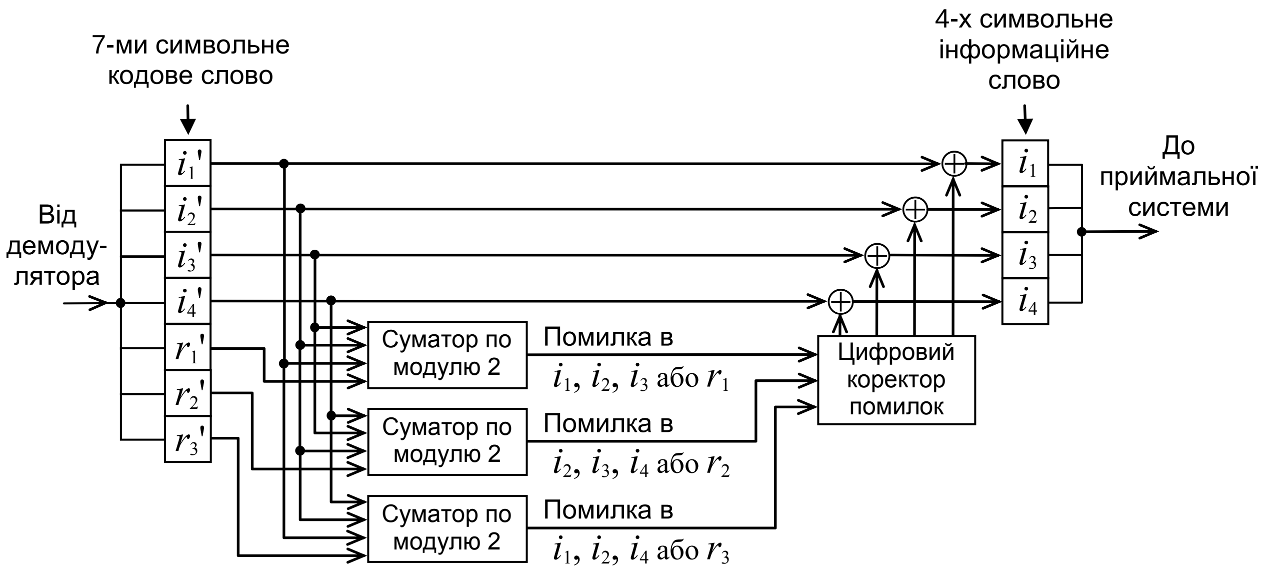


Рис. 5. Декодер для простого (7, 4) коду Хеммінга

В декодері в режимі виправлення помилок будується послідовність:

$$\begin{aligned} s_1 &= r_1^{'} \oplus i_1^{'} \oplus i_2^{'} \oplus i_3^{'}; \\ s_2 &= r_2^{'} \oplus i_2^{'} \oplus i_3^{'} \oplus i_4^{'}; \\ s_3 &= r_3^{'} \oplus i_1^{'} \oplus i_2^{'} \oplus i_4^{'}. \end{aligned} \tag{23}$$

Трьохсимвольна послідовність (s_1, s_2, s_3) називається *синдромом*. Термін "синдром" використовується і в медицині, де він позначає сполучення ознак, характерних для певного захворювання. В даному випадку синдром $S = (s_1, s_2, s_3)$ є сполучення результатів перевірки на парність відповідних символів кодової групи та характеризує певну конфігурацію помилок (шумовий вектор). Кількість можливих синдромів визначається виразом

$$N_S = 2^r. \tag{24}$$

При кількості перевірочних символів $r = 3$ маємо вісім можливих синдромів ($2^3 = 8$). Нульовий синдром (000) вказує на те, що помилки при прийомі відсутні або не виявлені. Будь-якому ненульовому синдрому відповідає певна конфігурація помилок, яка і виправляється. Класичні коди Хеммінга $(n, k) = (2^r - 1,\ 2^r - 1 - r)$, що мають таку кількість синдромів, яка точно дорівнює їх необхідній кількості, дозволяють виправляти усі однократні помилки у будь-якому інформаційному чи перевірочному символах та містять один нульовий синдром. Такі коди називають *щільноупакованими*.

Усічені коди не є щільноупакованими, оскільки кількість синдромів в них перевищує необхідну кількість. Так у коді (9, 5) при чотирьох перевірочних символах кількість синдромів буде дорівнювати $2^4 = 16$, у той час як необхідно лише 10. Зайві 6 синдромів свідчать про неповне упакування коду (9, 5).

Для коду (7, 4), що розглядається, в таблиці 8 представлені ненульові синдроми та відповідні їм конфігурації помилок.

Таблиця 8. Синдроми та відповідні їм конфігурації помилок для (7, 4) коду Хеммінга

Синдром	001	010	011	100	101	110	111
Конфігурація помилок	0000001	0000010	0001000	0000100	1000000	0010000	0100000
Помилка у символі	r_3	r_2	i_4	r_1	i_1	i_3	i_2

Таким чином, код (7, 4) дозволяє виправити всі поодинокі помилки. Проста перевірка показує, що кожна з помилок має свій єдиний синдром. При цьому можливе створення такого цифрового коректора помилок (дешифратора синдрому), який по відповідному синдрому виправляє відповідний символ в прийнятій кодовій групі. Після внесення виправлень перевірочні символи r_i можна на вихід декодера не виводити. Дві або більше помилок перевищують можливості коду Хеммінга щодо корекції і декодер помилятиметься. Це означає, що він вноситиме неправильні виправлення і видаватиме спотворені інформаційні символи.

Ідея побудови подібного коду, що коректує, зрозуміло, не змінюється при перестановці позицій символів в кодових словах. Всі такі варіанти також будуть називатися (7, 4) кодами Хеммінга.

РОЗШИРЕНИЙ КОД ХЕММІНГА

Додамо до кодових слів коду Хеммінга довжини 7 ще один перевірочний символ r_4, а до перевірочних співвідношень (23) ще одне (загальну перевірку на парність):

$$s_4 = i_1' \oplus i_2' \oplus i_3' \oplus i_4' \oplus r_1' \oplus r_2' \oplus r_3' \oplus r_4'. \qquad (25)$$

Новий код також міститиме 16 кодових слів, тому що, як і раніше, символи (i_1, i_2, i_3, i_4) можуть бути узяті якими завгодно. По ним із співвідношень (22) визначаються символи (r_1, r_2, r_3), а з рівності (25) символ r_4. В разі одиночної помилки співвідношення (25) порушується, а номер положення помилки визначається із співвідношень (23). Якщо ж сталася подвійна помилка, то співвідношення (25) буде виконано, а хоча б одна з рівностей (23) буде порушена. Це і дозволяє виявити будь-яку подвійну помилку. Таким чином, для виправлення одиночних помилок і виявлення подвійних помилок до чотирьох інформаційних символів досить додати чотири перевірочні символи.

КОДИ, ЩО ВИПРАВЛЯЮТЬ НЕСИМЕТРИЧНІ ПОМИЛКИ

На практиці часто зустрічаються канали, що відрізняються асиметричним характером помилок, наприклад, такі, в яких переважають заміщення вигляду $0 \to 1$ (тобто замість нуля приймається одиниця), а заміщення $1 \to 0$ украй рідкі. Звичайно, і в цьому випадку можна використовувати коди, призначені для виправлення заміщень обох видів, зокрема, розглянутий вище код Хеммінга. Але це буде дуже марнотратно, оскільки тоді здатність коду коректувати помилки витрачалася б наполовину даремно. Тому були придумані коди, пристосовані спеціально для виправлення несиметричних помилок.

Вихідне міркування тут дуже просте: якщо в каналі можливі лише помилки вигляду $1 \to 0$, то в прийнятому двійковому слові немає потреби перевіряти позиції з нульовими символами – вони напевно передані без помилок. Тому будемо робити перевірку так, щоб її результат залежав лише від позицій з одиничними символами, точніше від номерів цих позицій. З цією метою для довільного двійкового слова $u = x_1 x_2 \dots x_n$ складемо суму

$$S(u) = \sum_{i=1}^{n} x_i \cdot i \,. \tag{26}$$

В сумі (26) ненульові доданки відповідають лише одиничним символам і кожне з них збігається з номером цього символу, тобто число $S(u)$ дорівнює сумі номерів одиничних позицій слова u.

Знайдемо просту умову, що виділяє кодові слова серед інших. Шукатимемо цю умову у вигляді залишку від ділення суми (26) на деяке число l. Припустимо, що число l вже вибране, і розглянемо код, що складається з усіх таких слів $v = x_1 x_2 \dots x_n$, для яких сума номерів одиничних позицій $S(v)$ ділиться на число l без залишку. Позначимо вказаний код через $V_{n,l}$. Так при, $n = 4$, $l = 5$ отримаємо наступну множену кодових слів:

$$V_{4,5} = \{0000,\ 1001,\ 0110,\ 1111\}.$$

Неважко переконатися, навіть перебором всіх можливих випадків, що даних код виправляє будь-які поодинокі помилки вигляду $0 \to 1$. Наприклад, помилка в третьому символі слова 1001 переводить його в слово 1011. При цьому жодне інше слово не могло перетворитися в результаті поодинокої помилки в слово 1011. Тому одержувач декодує слово 1011 однозначно, вважаючи, що було послане слово 1001. Аналогічне буде відбуватися і з іншими двійковими наборами, що містять поодинокі заміщення нуля на одиницю.

Подібні властивості зберігаються і для довільного коду $V_{n,l}$, якщо $l \geq n + 1$. Зокрема, такий код виправляє будь-які поодинокі помилки вигляду $0 \to 1$; при цьому алгоритм їх виправлення є надзвичайно простим. Дійсно, нехай в кодовому слові $v = x_1 x_2 \dots x_n$ сталося не більше однієї помилки і вийшло слово u. Якщо помилка сталася в j-му символі, то зрозуміло, що $S(u) = S(v) + j$. Оскільки $S(v)$ ділиться на l без залишку, то залишок від ділення $S(u)$ на число l дорівнює j, тобто номеру позиції, в якій сталася помилка. Якщо ж помилки не сталося, то $S(u)$ ділиться на l без залишку.

Проілюструємо виправлення поодинокої помилки на прикладі розглянутого вище коду $V_{4,5}$. Нехай прийнято слово $u = 0111$. Тоді $S(u) = 2 + 3 + 4 = 9$. При діленні на 5 числа 9 отримуємо залишок 4. Отже помилка сталася в 4 позиції, тобто передавалося кодове слово 0110.

Порядок виконання роботи та методичні вказівки

Дана лабораторна робота складається з двох частин.

Перша частина роботи відноситься до методів ефективного кодування інформації. Робота виконується на персональному комп'ютері з використанням спеціальної комп'ютерної програми "Ефективне кодування інформації", головне вікно якої показано на рисунку 6. В головному меню програми здійснюється вибір завдання для виконання. Порядок виконання завдань може бути довільним. На кожному етапі виконання завдання здійснюється проміжна перевірка отриманих результатів. Продовження виконання завдання можливе лише при наданні правильного проміжного результату.

Завдання "Префіксність" містить задачі, які дозволяють дослідити відмінності між префіксними та непрефіксними кодами. Завдання виконується в наступному порядку:

1. Оберіть з двох запропонованих варіантів той код, який задовольняє умові префіксності. Для цього необхідно зробити клацання мишею на вибраному вами варіанті коду.
2. Декодуйте за допомогою вказаного префіксного коду запропоновану числову послідовність. Відповідь запишіть в полі "Результат". Натисніть кнопку "Перевірка".

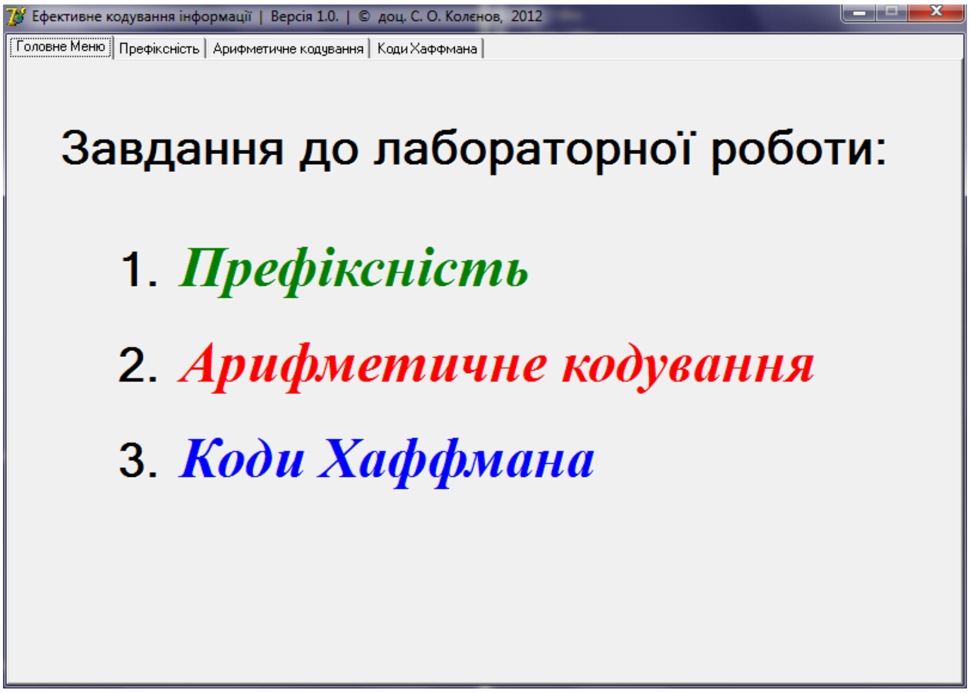


Рис. 6. Головне вікно програми "Ефективне кодування інформації"

3. Декодуйте за допомогою непрефіксного коду іншу запропоновану числову послідовність. Знайдіть два варіанти декодування. Відповіді запишіть у полях "Перший варіант" та "Другий варіант". Натисніть кнопку "Перевірка".

Завдання "Код Хаффмана" полягає у знаходженні кодів запропонованих символів, виходячи із заданої ймовірності появи цих символів. Для того, щоб знайти коди символів згідно із завданням, необхідно скласти повне дерево коду Хаффмана для усіх символів запропонованого алфавіту.

Зауваження! *На екрані буде відображена таблиця ймовірності появи символів заданого алфавіту. Натиснення на заголовок колонки "Ймовірність" відсортує символи в порядку зменшення їх ймовірності.*

Побудова кодового дерева починається знизу. В першу чергу об'єднуються два символи з найменшою ймовірністю. Нехай для прикладу це будуть символи E і G з ймовірністю 0.005 та 0.001 відповідно. Більшому ймовірністю символу відповідатиме одиниця, як останній символ, меншому – нуль. Далі, два розглянуті символи викреслюються з таблиці, і замість них в нову таблицю вписується новий сумарний символ (наприклад, EG), його ймовірність дорівнюватиме сумі ймовірностей цих двох символів ($P(EG) = P(E) + P(G) = 0.005 + 0.001 = 0.006$). Нова таблиця потім знову сортується, знаходяться і об'єднуються два символи з найменшою ймовірністю. Алгоритм повторюється до тих пір, поки в таблиці не залишиться лише два символи, що визначають верхівку дерева. Тому символу з них, що має більшу ймовірність, приписується одиниця, іншому – нуль.

Після складання кодового дерева код кожного з символів можна визначити, йдучи по дереву зверху вниз, від верхівки до цього символу, записуючи послідовно кодовий нуль або одиницю залежно від вибраних гілок дерева.

У порожню таблицю запишіть певні коди відповідних символів алфавіту. Переконайтеся, що в полях немає зайвих пробілів. Натисніть кнопку "Перевірка", щоб визначити правильність заповнення таблиці.

У завданні "Арифметичне кодування" потрібно буде виконати відповідне кодування та декодування запропонованих послідовностей. Для цього на початку потрібно буде для запропонованої послідовності символів заповнити допоміжну таблицю, вказавши ймовірності та інтервали, аналогічно прикладу в теоретичній частині. Заповнивши таблицю, натисніть кнопку "Перевірка таблиці" для перевірки. У випадку правильного заповнення усіх комірок таблиці програма запропонує перейти до наступної частини завдання.

Користуючись допоміжною таблицею, закодуйте запропоновану послідовність символів за допомогою арифметичного методу кодування. Натисніть кнопку "Перевірка". Користуючись тією ж допоміжною таблицею, декодуйте задану послідовність з 5 символів, закодовану методом арифметичного кодування. Натисніть кнопку "Перевірка".

Робота вважається виконаною при правильному виконанні усіх завдань. Хід роботи та отримані результати заносяться у звіт.

Зауваження! *Перезапуск програми "Ефективне кодування інформації" призведе до генерування нових завдань та втрати усіх отриманих результатів.*

Друга частина роботи відноситься до методів завадостійкого кодування. Робота виконується на персональному комп’ютері з використанням середовища програмування "Free Pascal". Даний персональний комп’ютер є частиною експериментальної установки для передачі та прийому інформаційних повідомлень, схема якої показана на рисунку 7.

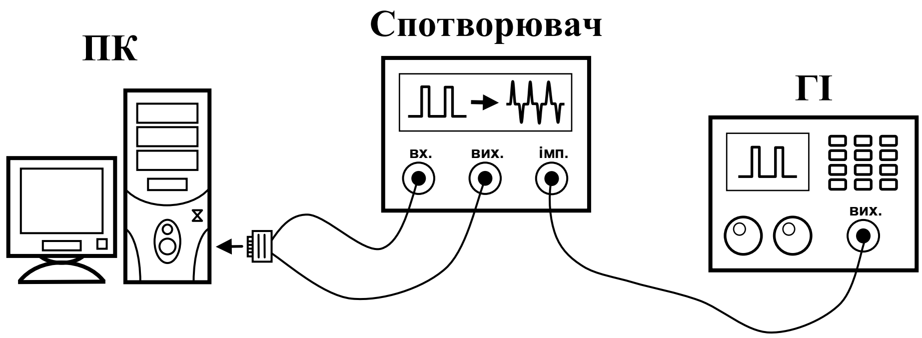


Рис. 7. Схема експериментальної установки для дослідження завадостійких кодів

В ході проведення досліджень завадостійких кодів персональний комп’ютер (ПК), використовуючи послідовний комунікаційний порт, передає задану кодовану послідовність даних у вигляді імпульсно-модульованих сигналів на вхід "Спотворювача". У "Спотворювачі" відбувається спотворення даних, що передаються, шляхом накладання на вхідний сигнал прямокутних імпульсів, що надходять з генератора імпульсів (ГІ), параметри яких (амплітуда, частота, тривалість) визначають ступінь спотворення та, відповідно, ймовірність виникнення однієї чи декількох помилок в кожному байті переданих даних. Таким чином, з виходу "Спотворювача" дані потрапляють назад до ПК з помилками, які можна виявити або виправити, користуючись можливостями корекції даних в завадостійкому коді, що досліджується. Відповідні алгоритми кодування та декодування даних, а також їх передачі та прийому реалізуються в середовищі програмування "Free Pascal" на ПК.

Робота виконується у відповідності із завданням. Хід роботи та отримані результати відображаються у звіті.

Завдання

До першої частини роботи:

1. Ознайомтеся з теоретичною частиною роботи, яка відноситься до методів ефективного кодування інформації та дайте відповіді на відповідні контрольні питання.
2. Запустіть на персональному комп'ютері програму "Ефективне кодування інформації" та дочекайтеся завершення формування нових завдань.
3. Виберіть в головному меню пункт завдання "Префіксність". Виконайте завдання згідно з вказівками у вікні програми.
4. Виберіть в головному меню пункт завдання "Коди Хаффмана". Знайдіть за методом Хаффмана коди усіх запропонованих символів, виходячи із заданої ймовірності їх появи, що відображена у відповідній таблиці.
5. Виберіть в головному меню пункт завдання "Арифметичне кодування". Виконайте відповідне кодування та декодування запропонованих послідовностей.
6. Зробіть висновки щодо розглянутих методів кодування відносно їх ефективності.

До другої частини роботи:

1. Ознайомтеся з теоретичною частиною роботи, яка відноситься до методів завадостійкого кодування інформації та дайте відповіді на відповідні контрольні питання.
2. Вивчіть експериментальну установку для передачі та прийому інформаційних повідомлень по послідовному протоколу через комунікаційний порт комп'ютера з використанням схеми, яка імітує лінію зв'язку, що вносить однократні чи двократні помилки в коди повідомлень, що передаються. Зауважте, що дані в лінію передаються по одному байту (по 8 біт).
3. Запустіть на персональному комп'ютері середовище програмування "Free Pascal" та створіть у ньому новий файл проекту. Збережіть цей файл на диску, вказавши своє прізвище як ім'я файлу.
4. Перенесіть у створений файл початковий програмний код з шаблону, лістинг якого наведено у Додатку. Цей програмний код реалізує алгоритм передачі та прийому інформаційних повідомлень через

комунікаційний порт комп'ютера.

5. На базі створеного файлу, розробити та застосувати наступні алгоритми передачі заданого викладачем інформаційного повідомлення:
   - алгоритм кодування вихідних повідомлень кодом Хеммінга з виправленням однократних помилок та виявленням двократних помилок у прийнятих повідомленнях (використовувати для кодування 4 інформаційних символи та 4 перевірочних символи);
   - алгоритм кодування з використанням несиметричних кодів, що коректують, для виправлення однократних помилок.
6. Реалізувати програмний код, що забезпечує вивід на екран інформації у вигляді таблиці з колонками:

№ символу	Прийнятий код	Виправлений код	Синдром	Кількість помилок	Позиція помилки

7. Користуючись кодом Хаффмана, зменшити надлишковість кодових послідовностей у повідомленні, що передається (спочатку закодувати початкове повідомлення кодом Хаффмана для зменшення середньої довжини кодової комбінації, а потім провести кодування отриманої послідовності коректуючим кодом Хеммінга).
8. Порівняти параметри усіх використаних методик кодування та заповнити наступну таблицю:

Параметр	Код Хеммінга	Несиметричний код	Код Хеммінга + код Хаффмана	Несиметричний код + код Хаффмана
Кількість дозволених/ заборонених код. комбінацій				
Ентропія (біт/симв.)				
Середня довжина код. комбінації (біт/симв.)				
Середня швидкість передачі повідомлень (симв./сек.)				

Контрольні питання

1. Чим відрізняються рівномірні коди від нерівномірних кодів?
2. Які коди називають префіксними? В чому їх особливість?
3. Який принцип покладений в основу ефективного кодування інформації?
4. В чому полягає суть ефективного кодування за методом Хаффмана?
5. Що таке кодове дерево? Для чого воно використовується?
6. Які недоліки методу Хаффмана не дозволяють досягти максимальної ефективності кодування?
7. В чому полягає перевага арифметичного кодування над кодуванням за методом Хаффмана?
8. Які властивості коду надають можливості виявляти та виправляти помилки?
9. Які основні характеристики мають коди, що коректують?
10. Що таке мінімальна кодова відстань і яке вона має практичне значення?
11. На якому принципі будується код Хеммінга? Якими властивостями він володіє? Які бувають види кодів Хеммінга?
12. Які коректуючі коди називають несиметричними? Як вони будуються та де використовуються?

Лабораторна робота № 3

КОРЕЛЯЦІЙНИЙ ПРИЙОМ ЦИФРОВИХ СИГНАЛІВ В КАНАЛІ З ШУМОМ

Мета роботи: Ознайомлення з принципами роботи систем зв’язку з шумоподібними сигналами. Дослідження властивостей псевдовипадкових (Баркера, М, Голда) та ортогональних (Уолша) послідовностей.

Литература:

1. Скляр Б. Цифровая связь. Теоретические основы и практическое применение: Пер. с англ. – М.: Издательский дом «Вильямс», 2003.
2. Волков Л. Н., Немировский М. С., Шинаков Ю. С. Системы цифровой радиосвязи. — М.: Эко-Трендз, 2005.
3. Алексеев А.И., Шереметьев А.Г., Тузов Г.И. Теория и применение псевдослучайных сигналов – М.: Наука, 1969. – 366 с.
4. Варакин Л. Е. Системы святи с шумоподобными сигналами. – М.: Радио и связь. – 1985, 384 с.

Теоретичні відомості

Системи зв'язку з шумоподібними сигналами (ШПС) відомі півстоліття. За цей час їх переваги стали очевидними, а багато недоліків було усунено. Системи зв'язку з ШПС займають особливе місце серед різних систем зв'язку, що пояснюється їх властивостями.

По-перше, вони мають високу перешкодозахищеність при дії потужних перешкод.

По-друге, забезпечують кодову адресацію великої кількості абонентів та їх кодове розділення при роботі в загальній смузі частот.

По-третє, вони забезпечують сумісність прийому інформації з високою достовірністю і вимірювання параметрів руху об'єкта з високою точністю і роздільною здатністю.

Всі ці властивості систем зв'язку з ШПС були відомі давно, але, оскільки потужності перешкод були відносно невисокі, а елементна база не дозволяла реалізувати пристрої формування та обробки в прийнятних габаритах, то довгий час системи зв'язку з ШПС використовувались тільки в спеціальних системах зв’язку.

У наш час мініатюризація та автономізація пристроїв зв’язку не дозволяє використовувати потужні передавачі, що суттєво обмежує відношення сигнал-шум. З іншої сторони розвиток мікроелектроніки дозволив розробити високоефективні та мініатюрні апаратні засоби обробки сигналів. Поштовхом до розвитку систем ШПС стала необхідність збільшення пропускної здатності частотних каналів, яке забезпечується кодовим розділенням абонентів.

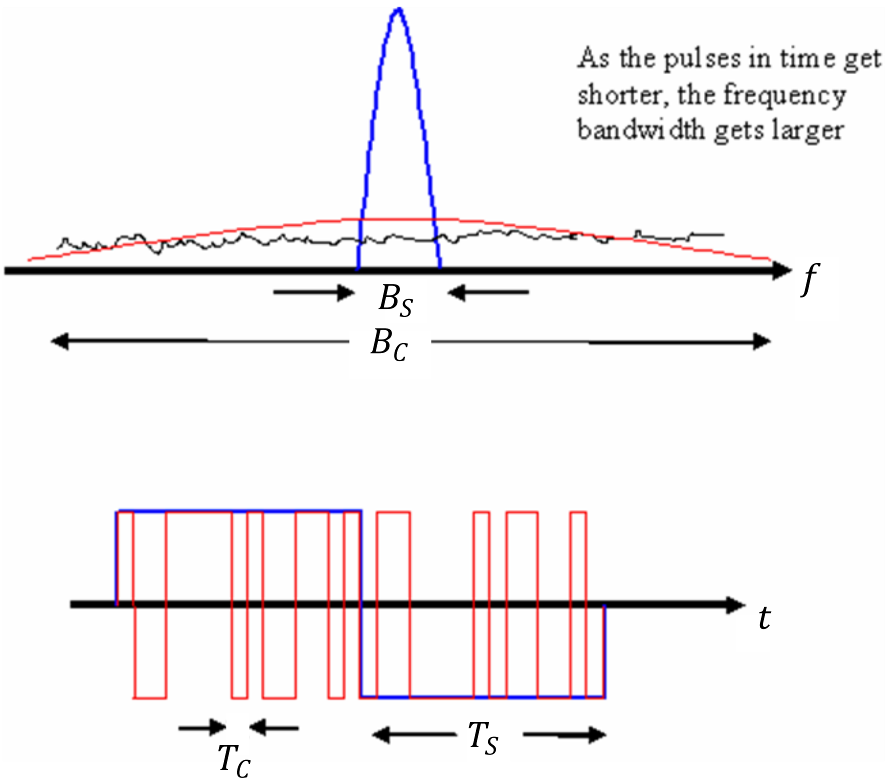


Рис. 1. Уширення спектру інформаційного сигналу (синій колір, T_S) шляхом кодування шумоподібним кодом (червоний колір, T_C)

Визначення

Шумоподібними сигналами (ШПС) називають такі сигнали в яких добуток ширини спектра ΔF на тривалість T набагато більше одиниці. Цей добуток називається базою сигналу і позначається $B = \Delta FT \gg 1$. Для звичайних сигналів $B = 1$.

У системах зв'язку з ШПС ширина спектра ШПС ΔF завжди набагато більше ширини спектру повідомлення, яке передається. У цифрових системах зв'язку, що передають інформацію у вигляді двійкових символів, тривалість ШПС та швидкість передачі інформації R пов'язані співвідношенням $T_S = 1/R$. Тому *база* ШПС $B = \Delta F/R$. База ШПС характеризує уширення спектру ШПС відносно спектру повідомлення (рис. 1). В аналогових системах зв'язку, в яких W є верхньою частотою повідомлення, а частота відліків $2W$,

$$B = \frac{\Delta F}{2W}.$$

І якщо $B \gg 1$, то $\Delta F \gg R$, та $\Delta F \gg 2W$. Саме тому системи зв'язку з ШПС у закордонній літературі отримали назву системи зв'язку з розширеним (або розподіленим) спектром (*Spread Spectrum*), а у вітчизняній літературі – *широкосмугові системи передачі*. Надалі термін "широкосмугові системи передачі" (ШСП) буде відповідати системам зв'язку з ШПС.

Розглянемо основні особливості систем з ШПС.

Завадостійкість

Відношення сигнал-шум q^2 на виході приймача (після детектування) і ρ^2 на його вході пов'язані співвідношенням

$$q^2 = 2B\rho^2 \qquad (1)$$

де $\rho^2 = P_c/P_ш$ (P_c, $P_ш$ – потужності ШПС та шуму відповідно), $q^2 = 2E/N_ш$, E – енергія ШПС, $N_ш$ – спектральна густина потужності шуму в смузі ШПС.

Як видно з (1) відношення сигнал-шум після детектування може бути рівним заданому, навіть якщо вхідне значення цього параметру менше одиниці (сигнал нижче рівня шумів), лише відповідно підвищивши базу сигналу. Тобто детектування ніби підсилює сигнал в 2В раз. Саме тому величину

$$K_{\text{ШПС}} = q^2 / \rho^2$$

називають *коефіцієнтом підсилення* ШПС при обробці або просто підсиленням обробки.

Наприклад (рис. 2), якщо необхідно мати $q^2 = 20\ \partial Б$, а на вході приймача $\rho^2 = -40\ \partial Б$, то необхідна база повинна бути рівною $60\ \partial Б$, тобто $B = 10^6$.

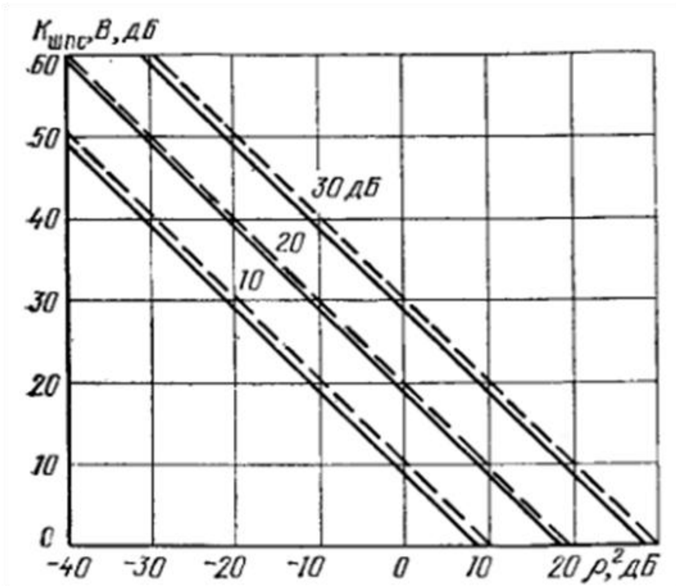


Рис. 2. Залежність підсилення обробки і бази ШПС від відношення сигнал-шум на виході приймача.

Рівняння (1) застосовне для завади у вигляді Гауссівського білого шуму з рівномірно розподіленою спектральною густиною потужності в смузі ШПС і являється фундаментальним в теорії систем зв'язку з ШПС. Але це ж співвідношення справедливе і для вузькосмугових завад.

Електромагнітна сумістність

ШПС добре протидіють вузькосмуговим перешкодам, якими є (для ШПС) більшість сигналів у звичайних системах зв'язку. З іншого боку інтенсивність ШПС зазвичай нижче рівня білого шуму в каналі, а тому вони вносять в канал звичайної системи зв'язку шуму не більше ніж там вже є. Можливість використання ШПС та звичайних сигналів в одній частотній смузі без взаємних перешкод і називається *електромагнітною сумістністю*.

<u>Прихованість</u>

Це здатність сигналу протидіяти спробам детектування та вимірювання параметрів. Загалом розрізняють багато ситуацій з різними прийомами забезпечення прихованості, що залежать від апріорних початкових знань небажаного користувача про сигнал. Зазвичай вважають, що відомо смугу передачі сигналу. Тоді час детектування буде збільшуватись по мірі зростання бази сигналу та ширини смуги.

<u>Боротьба з багатопроменевим прийомом</u>

Сигнал може відбиватись від атмосфери або наземного об'єкту і прийти на приймач з часовою затримкою відносно основного сигналу (рис. 3). При інтерференції цих сигналів прийняте повідомлення сильно спотворюється. Системи з ШПС успішно справляються з цим, оскільки після декодування можна чітко розділити основний та відбитий сигнали (рис. 3 знизу).

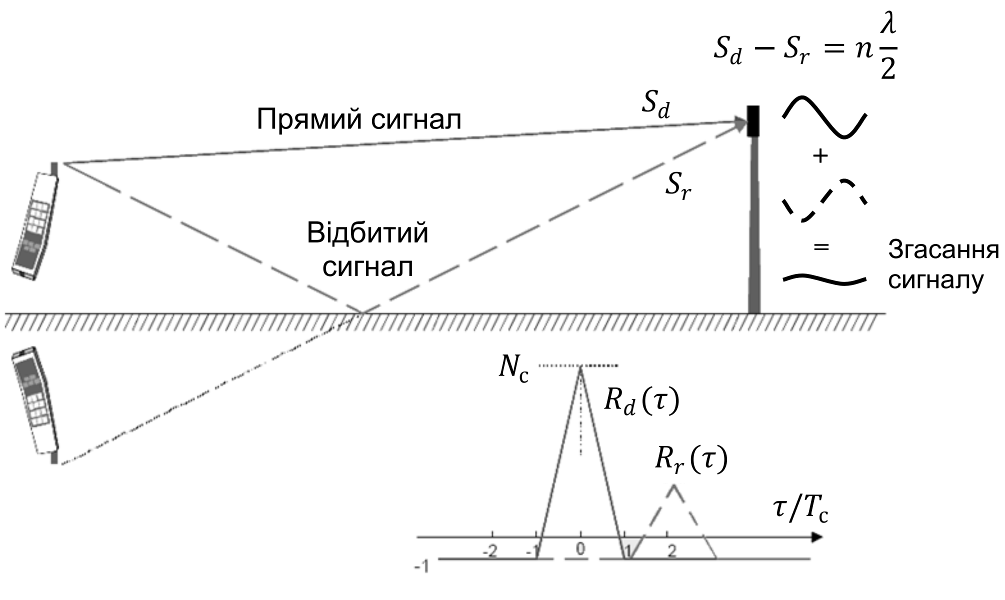


Рис. 3. Вплив відбитого сигналу на роботу приймача. Знизу: сигнал після детектування.

Визначення координат рухомих об'єктів

Визначити положення об'єкту можна, вимірявши затримку сигналу, точність якої пропорційна ширині частотної смуги. Швидкість вимірюють за допомогою доплерівського зсуву частоти, точність якого пропорційна тривалості сигналу. Саме тому система глобального позиціонування GPS використовує ШПС.

Кодове розділення абонентів

В системах зв'язку другого покоління (GSM) використовується частотне та часове розділення абонентів, а тому спектри сигналів намагаються зробити якнайвужчими. В системах третього покоління використання сигналів з великою базою дозволило передавати в одній частотній смузі багато сигналів і, чим ширша база, тим більше цих сигналів може бути. Це дозволило організувати кодове розділення абонентів (CDMA) (рис. 4). Всі абоненти можуть працювати в загальній смузі частот, а розділити їх можливо за рахунок відмінності ШПС за кодом.

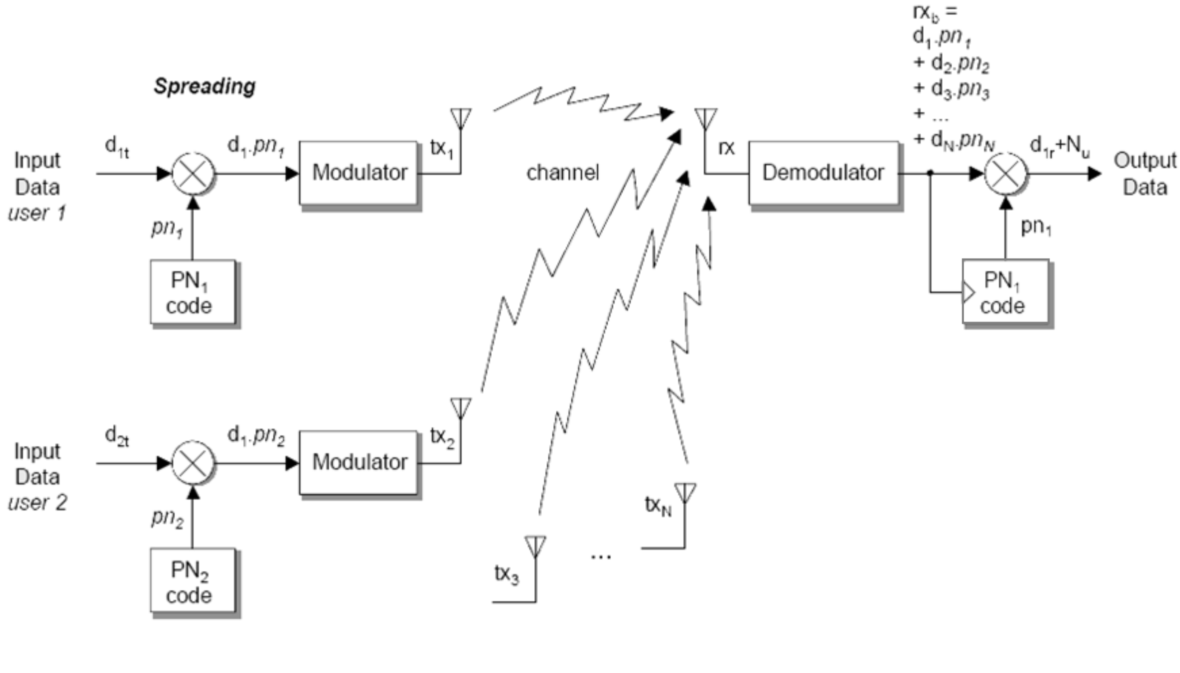


Рис. 4. Множинний доступ при кодовому розділенні каналів

При цьому ШПС є по суті справи адресою абонента і в цьому випадку принципово немає необхідності в примусовій тимчасової синхронізації абонентів. Тому подібні системи зв'язку отримали назву *асинхронних адресних систем зв'язку* (ААСС).

Недоліком таких систем є взаємні завади, що створюють інші абоненти в даній смузі частот (рис. 4). Таким чином ємність каналу обмежена базою сигналів. Тому розширюючи базу ми збільшуємо ємність. Це роблять шляхом збільшення довжини кодової послідовності (тобто зменшення швидкості передачі). З іншого боку взаємні завади сильно залежать від кроскореляційних і автокореляційних властивостей кодових послідовностей.

Автокореляція

Кореляційні властивості кодових послідовностей, що використовуються в системах з ШПС, залежать від типу кодової послідовності, її довжини, частоти проходження її символів і від її посимвольної структури.

Автокореляція – статистичний взаємозв'язок між випадковими величинами з одного ряду, але взятих зі зміщенням, наприклад, для випадкового процесу – зі зміщенням у часі.

У загальному вигляді автокореляційна функція (АКФ) для функції $f(t)$ визначається інтегралом

$$R(\tau) = \int_{-\infty}^{+\infty} f(t)f(t-\tau)dt$$

і показує зв'язок сигналу з копією самого себе, зміщеного в часі на величину τ. Вивчення АКФ грає важливу роль при виборі кодових послідовностей з точки зору найменшої ймовірності помилкової ідентифікації. Ідеальна АКФ – гострий пік і відсутність бічних пелюсток. АКФ для прямокутного імпульсу тривалістю T зображено на рис. 5. АКФ для псевдовипадкової неперіодичної послідовності з тривалістю елементу τ_0 та загальною тривалістю T зображено на рис. 6.

Як видно для ШПС пік АКФ в T/τ_0 раз вужчий ніж у прямокутного імпульсу такої ж тривалості, але з'являються бічні піки. Величина цих піків обернено пропорційна кореню квадратному з бази ШПС. Таким чином при $B \to \infty$ бічні піки зменшуються до нуля. Це має місце при ідеальній АКФ (рис. 7). При цьому також $\tau_0 \to 0$, тобто пік стає гостріший.

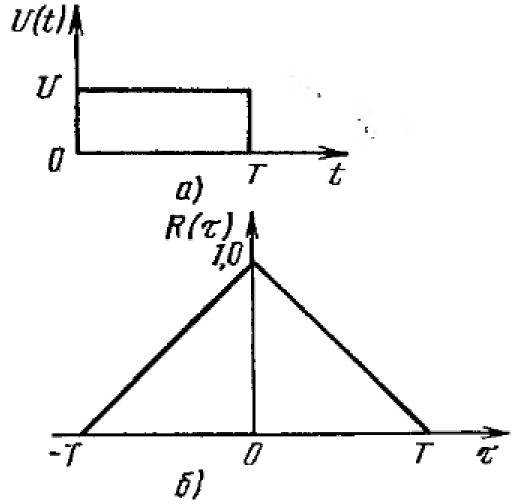


Рис. 5. Прямокутний імпульс (а) та його автокореляційна функція (б)

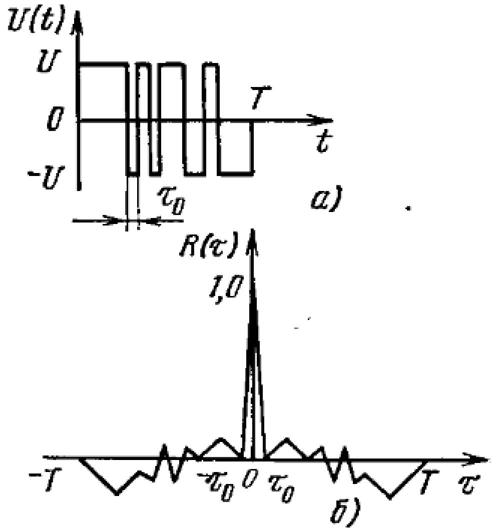


Рис. 6. Фазоманіпульований шумоподібний сигнал та його автокореляційна функція (б)

Виявляється, що найкращі автокореляційні властивості має білий шум – для нього автокореляційна функція – це нескінченно великий дельта-імпульс на початку координат, а поза ним значення функції нуль.

Кроскореляційна функція (взаємна кореляційна функція (ВКФ)) відрізняється від АКФ тим, що інтегрується добуток різних функцій:

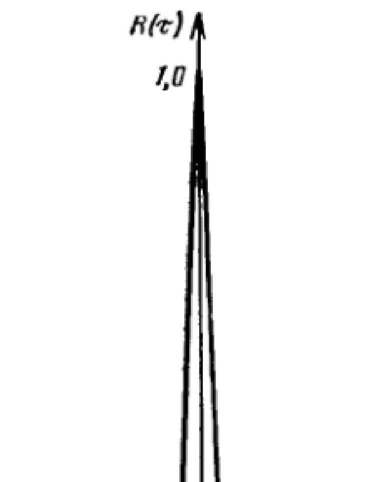


Рис. 7. Ідеальна АКФ

$$R(\tau) = \int_{-\infty}^{+\infty} f(t)g(t-\tau)dt.$$

ВКФ має значення в системах зв'язку з ШПС, в яких потрібно забезпечити множинний доступ абонентів (CDMA). Ідеальною ВКФ для таких систем є ВКФ з мінімальною амплітудою, а коди з такою ВКФ називають *ортогональними*.

<u>Білий шум</u>

Білим шумом називається випадковий процес, значення якого розподілені за Гауссівським законом:

$$p(x) = \frac{1}{\sqrt{2\pi\sigma}} exp\left[\frac{-(x-x_m)^2}{2\sigma^2}\right],$$

де x_m – математичне очікування, а σ^2 – дисперсія. Білий шум ще називають *адитивним білим гауссівським шумом* (АБГШ).

Для звичайного АБГШ, що наявний в каналі зв'язку, математичне очікування (середнє) дорівнює нулю, а дисперсія $\sigma = \frac{1}{2}N_0$, вона ж дорівнює спектральній густині потужності $\Phi_{\mathrm{n}}(f) = \frac{1}{2}N_0$. Оскільки АКФ являється оберненим Фур'є перетворенням спектральної густини потужності, то $R(\tau) = \frac{1}{2}N_0\delta(\tau)$, що відповідає ідеальній АКФ (рис. 7). З іншого боку тривалість білого шуму вважається нескінченною, тому база АГБШ прямує в нескінченність, а отже його АКФ має прямувати до ідеальної. Оскільки АГБШ має ідеальну АКФ, то псевдошумові послідовності намагаються робити якомога більш схожими на білий шум.

<u>Кодові послідовності</u>

Кодові послідовності класифікують на ортогональні та шумоподібні. Для перших характерна відсутність кроскореляції між різними користувачами в системах з множинним доступом. Шупоподібні послідовності намагаються робити якомога ближчими за властивостями до білого шуму. Вони характеризуються гарними автокореляційними властивостями, а перехресні завади від інших користувачів рівномірно розподіляються у часі та між користувачами.

Шумоподібні коди обираються так, щоб мати наступні властивості:

1) Кожен елемент послідовності (0 або 1) має однакову частоту появи (збалансованість).

2) Низький рівень бічних максимумів АКФ для зменшення вірогідності помилки при детектуванні.

3) Низька кроскореляція між різними кодами.

Останні два пункти є взаємовиключаючими: намагання зменшити кроскореляцію зменшує випадковість послідовності та збільшує побічні максимуми її АКФ. Перший пункт пов'язаний з нульовим середнім, яким характеризується Гауссовий розподіл випадкового процесу. Він також важливий для виключення постійної складової інформаційного сигналу.

Періодична АКФ на всьому періоді N k-ї послідовності $a^{(k)}$ розраховується наступним чином

$$R_{k,k}(n) = \frac{1}{2N}\sum_{i=0}^{N-1} a_i^{(k)}\, a_{i+n}^{*(k)} \,. \tag{2}$$

Аналогічна кроскореляційна функція

$$R_{k,m}(n) = \frac{1}{2N}\sum_{i=0}^{N-1} a_i^{(k)}\, a_{i+n}^{*(m)} \,. \tag{3}$$

Принцип розрахунку АКФ та ВКФ таким чином показує наступний приклад (рис. 8):

pn(0) = +1 +1 +1 -1 +1 -1 -1
pn(0) = +1 +1 +1 -1 +1 -1 -1
+1 +1 +1 +1 +1 +1 +1 $\rightarrow \Sigma = 7 = R(\tau=0)$

pn(0) = +1 +1 +1 -1 +1 -1 -1
pn(1) = +1 +1 -1 +1 -1 -1 +1
+1 +1 -1 -1 -1 +1 -1 $\rightarrow \Sigma = -1 = R(\tau=1)$

Тут pn(0) – задана періодична послідовність, pn(1) – ця ж послідовність зсунута на один символ.

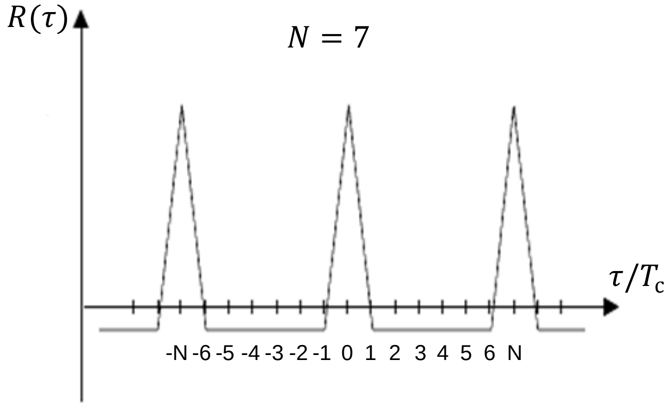


Рис. 8. Розрахунок АКФ за формулою (2)

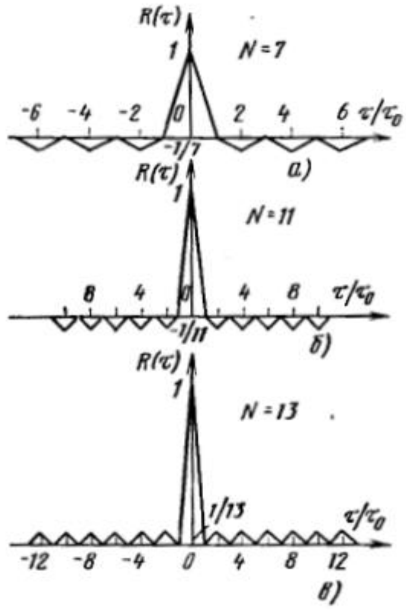


Рис. 9. АКФ для послідовності Баркера з $N = 7$ (а), $N = 11$ (б), $N = 13$ (в)

Послідовність Баркера – кодова послідовність, що складається з символів $a_i = \pm 1$, де $i = 0 \dots N$ і характеризується АКФ виду:

$$R(n) = \begin{cases} 1\,, & n = 0; \\ 0\,, & n = 2l + 1; \\ \pm 1/N\,, & n = 2l. \end{cases}$$

Коди Баркера характеризуються гарними автокореляційними та кроскореляційними властивостями (рис. 9). Але їх довжина обмежена. Коди Баркера відомі лише з $N \leq 13$. А тому вони не використовуються в системах зв’язку типу CDMA, а лише в спеціалізованих системах, системах бездротових мереж (WLAN, наприклад, Wi-Fi IEEE 802.11) або як синхроімпульси для тих самих CDMA. Таблиця 1 демонструє всі відомі послідовності Баркера.

Таблиця 1. Відомі послідовності Баркера

N	Кодова послідовність a_i													R_{21}
	1	2	3	4	5	6	7	8	9	10	11	12	13	
3	1	1	-1											$-1/3$
4	1	1	-1	1										$1/4$
5	1	1	1	-1	1									$1/5$
7	1	1	1	-1	-1	1	-1							$-1/7$
11	1	1	1	-1	-1	-1	1	-1	-1	1	-1			$-1/11$
13	1	1	1	1	1	-1	-1	1	1	-1	1	-1	1	$1/13$

М-послідовність

Псевдовипадкові послідовності можна генерувати за допомогою регістру зсуву зі зворотнім зв'язком для всіх ланок регістру на спільний вхід (рис. 10). Зротній зв'язок для лінійної послідовності відбувається через суматор по модулю 2, що працює за наступним правилом: $0 \oplus 0 = 1 \oplus 1 = 0$; $0 \oplus 1 = 1$. Інакше кажучи, якщо маємо парну кількість одиниць, то на виході суматора буде 0, якщо ж одиниць непарна кількість, то на виході буде 1.

Функції $f(x_1, x_2, \ldots)$ є многочленами порядку L, з коефіцієнтами зворотного зв'язку c_i. Максимально можлива довжина послідовності $N = 2^L - 1$. Такі послідовності називають послідовностями максимальної довжини або просто М-послідовності.

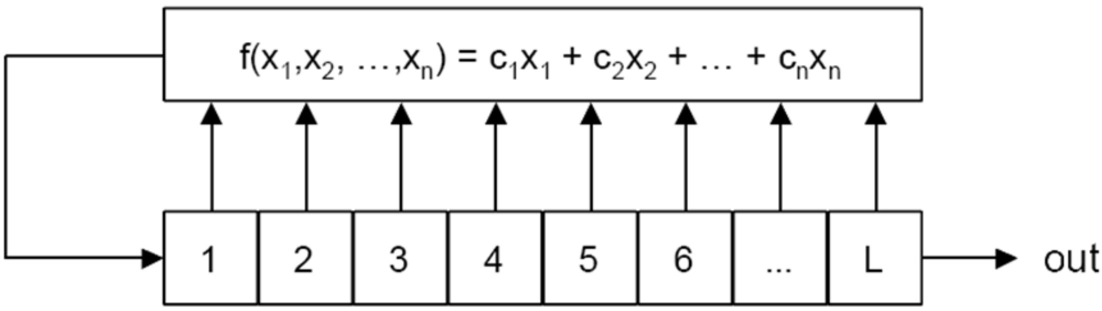


Рис. 10. Зсувний регістр з L розрядами для генерації М-послідовності

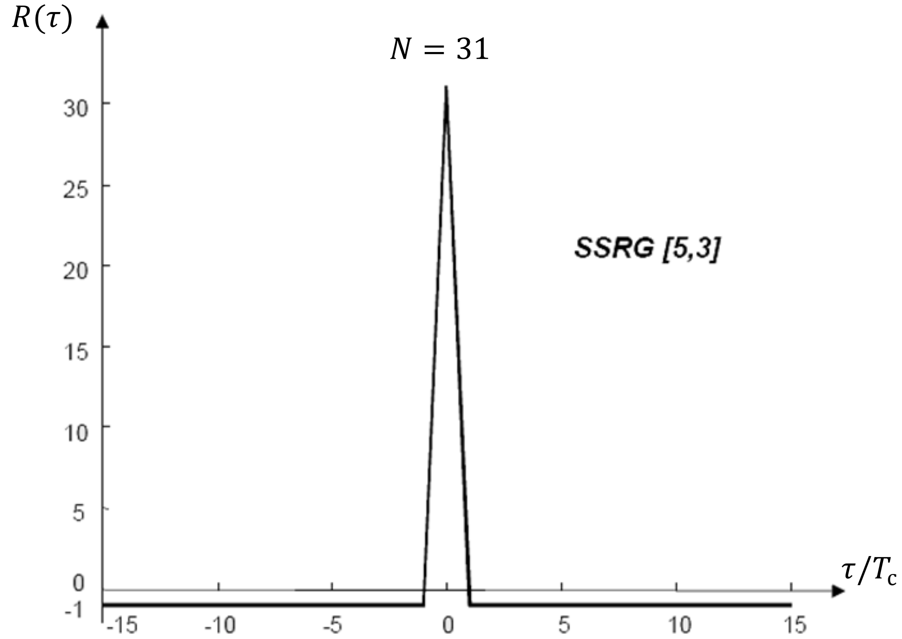


Рис. 11. Періодична АКФ для М-послідовності з $N = 31$

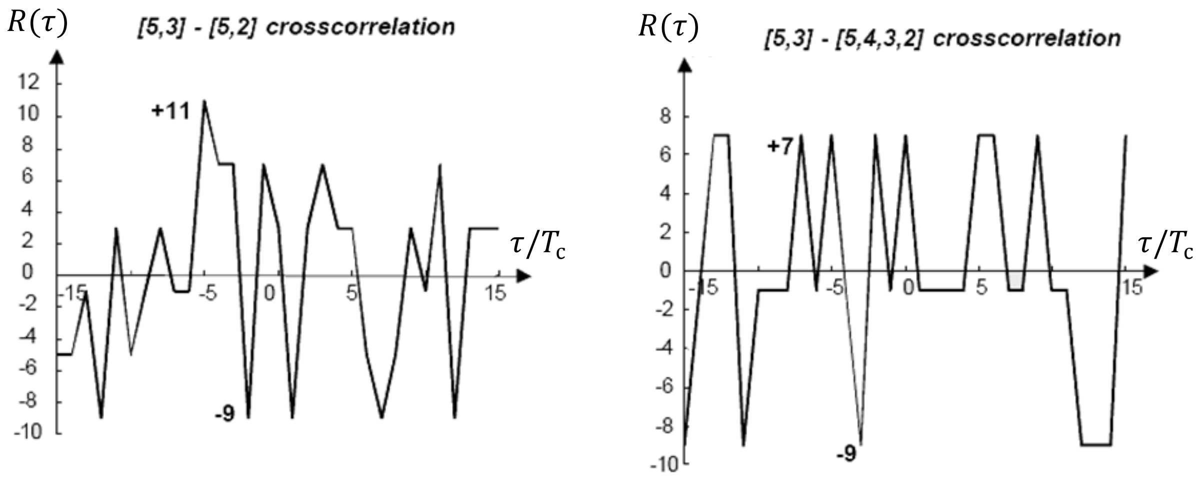


Рис. 12. ВКФ для різних пар М-послідовностей

Такі послідовності мають наступні основні властивості:

1. М-послідовність є періодичною з періодом, що складається з N імпульсів (символів).
2. Бічні піки періодичної автокореляційної функції сигналів, утворених М-послідовністю, рівні -1 для всіх величин зсуву τ окрім області $[-1;\ +1]$, в якій вона лінійно зростає до N (рис. 11). Якщо послідовність неперіодична, то бічні піки можуть не дорівнювати -1.

3. М-послідовність – збалансована, оскільки кількість одиниць на періоді послідовності 2^{L-1}, а нулів $2^{L-1}-1$. Нулів завжди на один менше, оскільки нульовий початковий стан зсувного регістру призводить до зациклювання його елементів в нульовому стані, а тому послідовностей з всіма нулями не існує.

Періодична АКФ для М-послідовностей найбільш близька до АКФ білого шуму (рис. 11). Кроскореляційні властивості дещо гірші (рис. 12), при чому вони не однакові для всіх можливих пар М-послідовностей. На рисунках 11 та 12 в квадратних дужках зазначено номери коефіцієнтів зворотного зв’язку, що не дорівнюють нулю.

Оскільки М-послідовності лінійні, то їх легко можна детектувати, знаючи початок послідовності.

Коди Голда та Касамі

М-послідовності, як ми вже бачили, мають погані кроскореляційні властивості. Принципово можна дещо пожертвувати автокореляційними властивостями псевдовипадкової послідовності, щоб покращити кроскореляційні властивості (для послідовностей однакової довжини). Якщо спеціально обрані М-послідовності однакової довжини, що циклічно зсуваються одна відносно одної, просумувати по модулю 2, то отримаємо, так звані, коди Голда (рис. 13). Якщо розрядність регістрів зсуву m, то окрім початкових М-послідовностей отримаємо ще 2^m-1 нових послідовностей з гарними кроскореляційними властивостями (рис. 14). І загалом отримуємо 2^m+1 послідовностей.

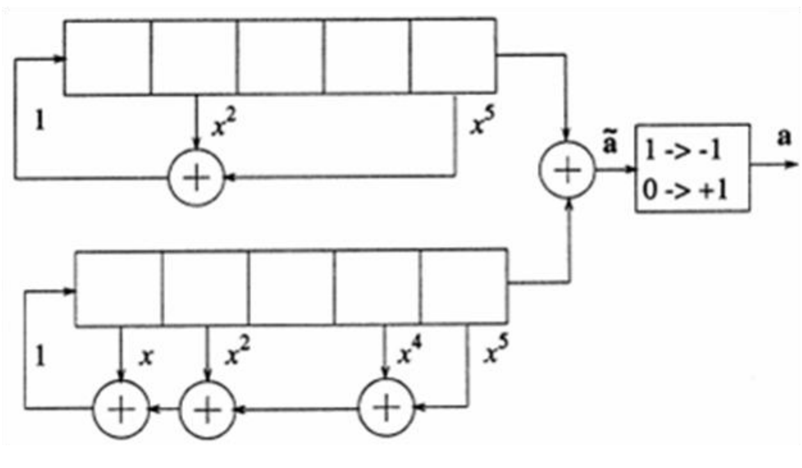


Рис. 13. Приклад схеми для отримання послідовностей Голда

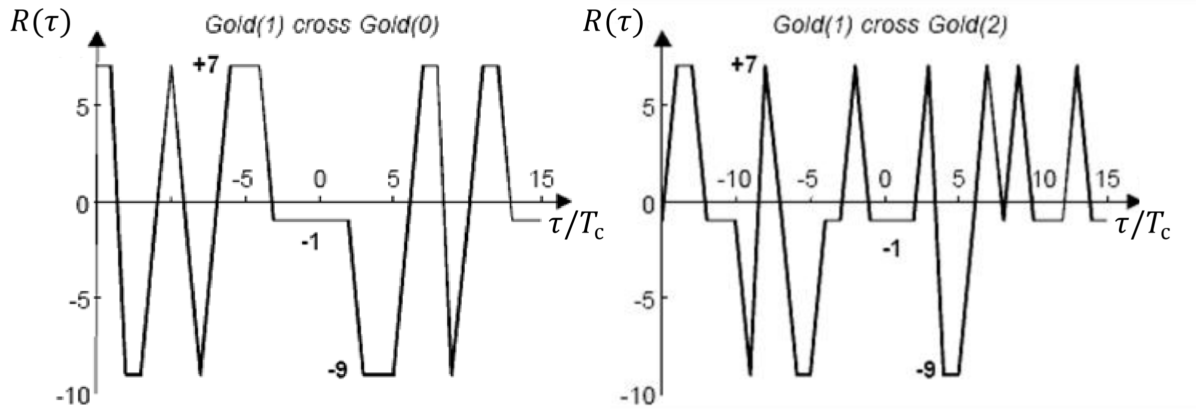


Рис. 14. ВКФ кодів Голда. В дужках вказано фазовий зсув між М-послідовностями [5,4,3,2] та [5,3]

Послідовності Касамі генеруються подібно до кодів Голда, але використовується одна коротка і одна довга М-послідовності. Коди Касамі мають на половину нижчий рівень піків ВКФ ніж коди Голда такої ж довжини, але їх кількість теж менша – одна довга М-послідовність та $2^{m/2} - 1$ нових послідовностей.

Коди Уолша та матриця Адамара

В системах з множинним доступом зазвичай використовують ортогональні коди, оскільки вони забезпечують мінімальні перехресні завади між користувачами. Такими кодами є система кодів Уолша. Для дослідження систем Уолша з точки зору їх кроскореляційних властивостей доцільно використовувати матриці Адамара, що визначаються через рекурентне співвідношення:

$$H_{2N} = \begin{vmatrix} H_N & H_N \\ H_N & -H_N \end{vmatrix}. \quad (4)$$

Як послідовності Уолша обирають рядки або стовпчики матриці Адамара. Приклад матриць Адамара:

$$H_2 = \begin{vmatrix} 1 & 1 \\ 1 & -1 \end{vmatrix}, \quad H_4 = \begin{vmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{vmatrix},$$

$$H_8 = \begin{vmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{vmatrix}.$$

Відстань між будь якими двома послідовностями (кількість неспівпадінь) складає $N/2$. Мається на увазі відстань між векторами у просторі сигналів, в яких роль координат відіграють символи послідовності. Ці вектори еквівалентні відповідним послідовностям.

Позначимо j-ту кодову послідовність Уолша як $\{W_j\}$ а її n-й символ через $W_j(n)$. Ортогональність кодових послідовностей Уолша записується через рівність:

$$\sum_{n=0}^{N-1} W_j(n) W_k(n) = \begin{cases} 0, & j \neq k, \\ N, & j = k. \end{cases}$$

Або в матричному записі:

$$H_N H_N^T = NI, \tag{5}$$

де I – одинична матриця. Оскільки послідовностями Уолша є стовпці або рядки матриці Адамара, то зрозуміло, що в (5) послідовності Уолша перемножуються між собою. Послідовності Уолша ортогональні, тому скалярний добуток векторів у просторі сигналів, що відповідають певним послідовностям ненульовий, якщо вектор множиться сам на себе (що ми і бачимо в (5)). Відносний зсув послідовностей за часом (що еквівалентно зсуву на певну кількість символів) еквівалентний відносній зміні положення векторів між собою. Таким чином вектори (сигнали) перестають бути ортогональними. Тому на практиці ортогональність забезпечується лише за умови часової синхронізації.

Властивості послідовностей

Послідовність	Властивості	Застосування
Barker	+ гарна автокореляція − кроскореляція незастосовна (різні послідовності мають різну довжину) − невелика довжина (3,4,5,7,11,13)	DSSS (прямі послідовності), прості радіомодеми
M-sequence	+ гарна автокореляція − погана кроскореляція + велика довжина	Кореляція (часова синхронізація)
Gold, Kasami	+ не така гарна автокореляція + але гарна кроскореляція + більша кількість послідовностей	Кореляція на одній несучій
Walsh	+ ортогональність − необхідна синхронізації у часі	CDMA

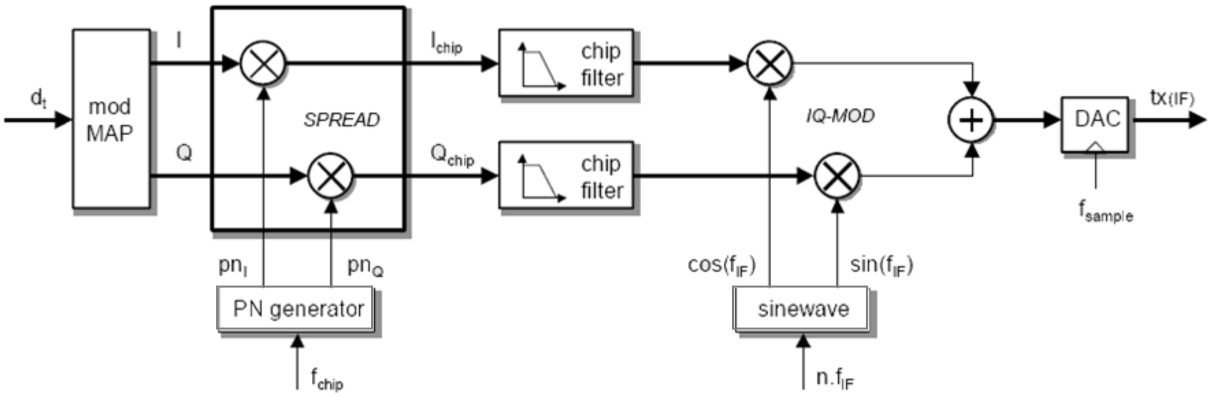
Схема системи зв'язку з ШПС

Структурна схема передавача та приймача для систем з прямим кодом (DSSS) зображена на рисунку 15.

Процес роботи передавача включає три основних етапи: перетворення інформаційних бітів у символи, уширення спектру та модуляція несучої. Варто зазначити, що уширення може відбуватись як на відеочастотах так і на несучій. В приймачі відповідно операції проходять в зворотному напрямку. Кодування інформаційних бітів відбувається власне послідовністю для одного символу та інвертованою послідовністю для іншого символу. Наприклад, ми хочемо закодувати інформаційні біти послідовністю Баркера з $N = 5$; якщо одиницю закодуємо послідовністю 11101, то нуль послідовністю 00010.

Звузити спектр (детектування) можна двома засобами: узгодженим фільтром або корелятором. Для правильної роботи обох потрібно знати кодові послідовності, що використовуються. Саме на детекторі відбувається підсилення обробки відповідно до формули (1), тобто максимізується відношення сигнал-шум.

Передавач



Приймач

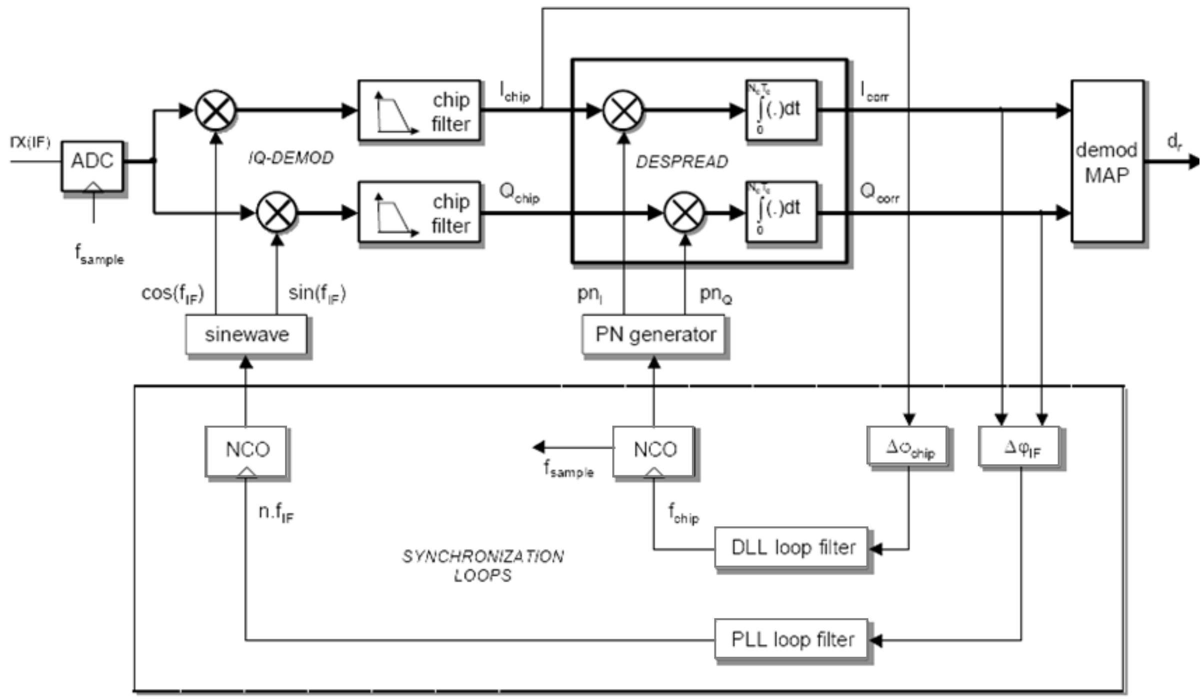


Рис. 15. Передавач та приймач DSSS систем

Оптимальний прийом сигналу (узгоджений фільтр та корелятор)

Узгодженим називається фільтр, імпульсний відгук якого є обернена у часі обвідна сигналу, з яким фільтр є, власне, узгоджений. Такий фільтр ще називають пасивним, оскільки його параметри не змінні у часі. Як наслідок, узгоджений фільтр не потребує фазової та часової синхронізації з прийнятим сигналом.

В кореляторі застосовується помножувач та інтегратор. Його параметри змінюються в залежності від поданого опорного сигналу на помножувач. Тому корелятор ще називають активним фільтром. Для нормальної роботи корелятора необхідна часова та фазова синхронізація. Схему приймача з корелятором зображена на рисунку 15. В кореляторі опорний сигнал зсувається відносно прийнятого на час $\tau_0/2$, аж поки сигнал після корелятора не досягне порогового значення.

Загалом обидва типи детекторів вираховують автокореляційну функцію сигналу. Але узгоджені фільтри за рахунок паралельної роботи мають більшу швидкодію, однак, корелятори потребують для своєї реалізації менше технічних засобів. Швидкодія сучасних кореляторів досить висока, тому вони частіше застосовуються в бездротових мережах зв’язку.

Схема експериментальної установки

Програмований пристрій запису-зчитування складних фазо або частотноманіпульованих сигналів побудований на основі цифрового синтезатора частоти прямого синтезу (DDS, Direct Digital Synthesizer) AD7008 фірми Analog Device, керованого від комп'ютера IBM PC. Характеристики та структурна схема показані у таблиці 2 та на рисунку 16.

Для обробки (декодування) прийнятого сигналу від пристрою запису-зчитування використовується прилад цифрового вводу сигналу кореляції. Його структурна схема зображена на рисунку 17.

Таблиця 2. Характеристики програмованого пристрою запису-зчитування

Діапазон частот	0 ÷ 16 МГц
Точність установки частоти	0.01 Гц
Діапазон регулювання фази	0° ÷ 360°
Точність установки фази	2’
Діапазон регулювання амплітуди	0 ÷ 100 %
Точність установки амплітуди	0.2 %
Нерівномірність частотної характеристики, не більше	3 дБ
Тривалість однієї події	0.32÷5.12 мкс
Максимальне число активних подій	512
Максимальне загальне число подій	512
Діапазон тривалостей сигналу	0.32÷2621.44 мкс

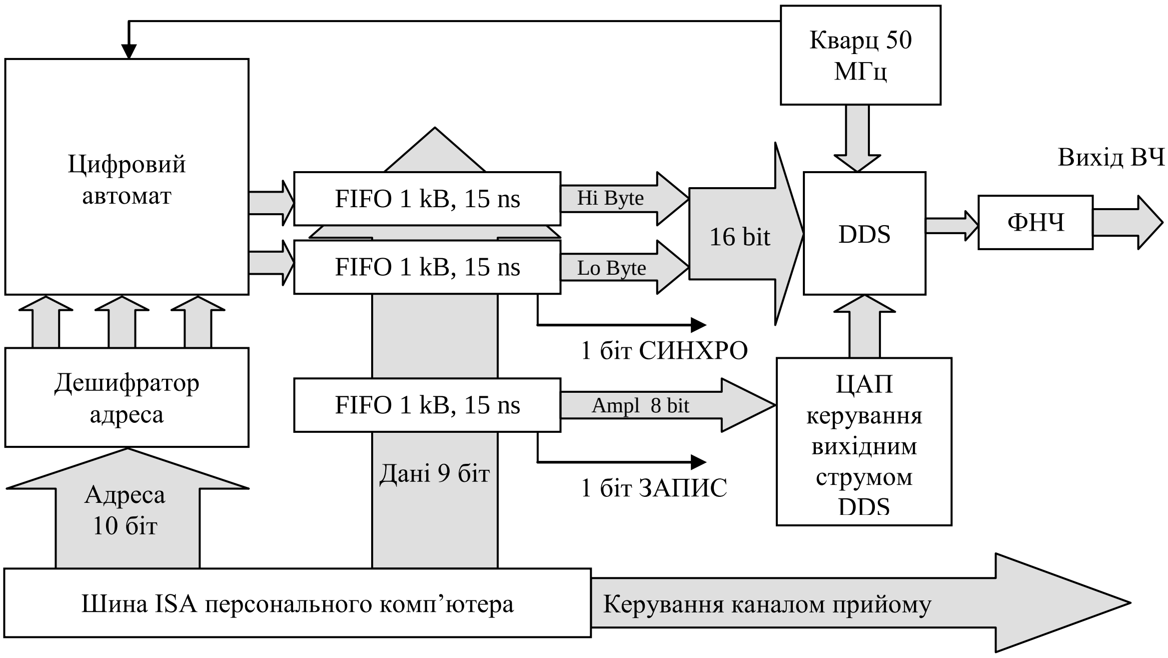


Рис.16. Структурна схема програмованого пристрою запису-зчитування

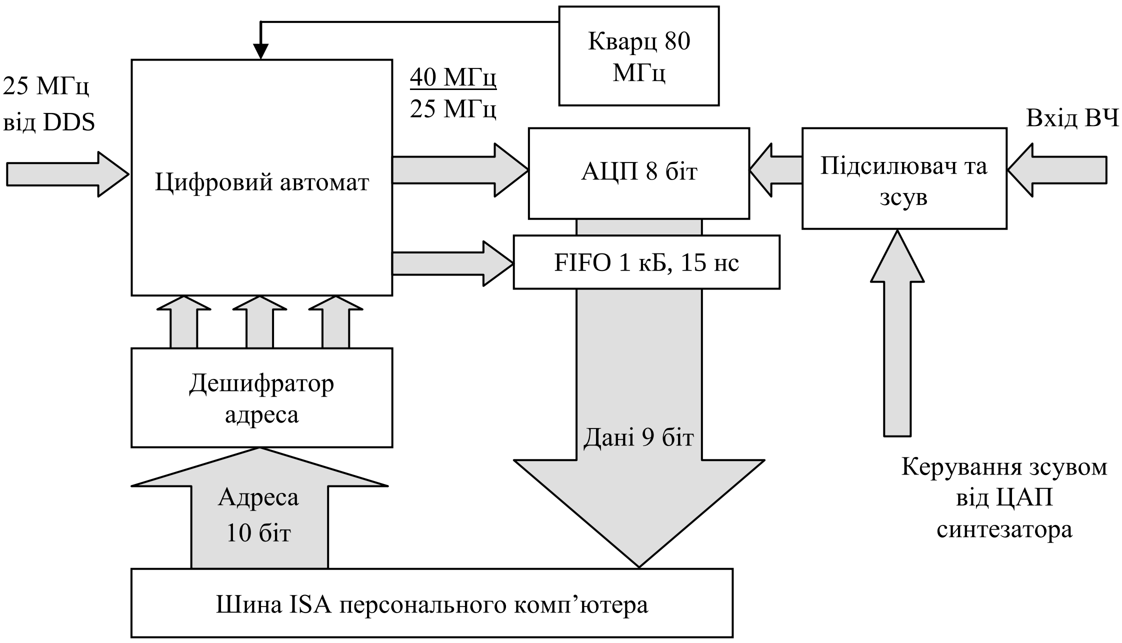


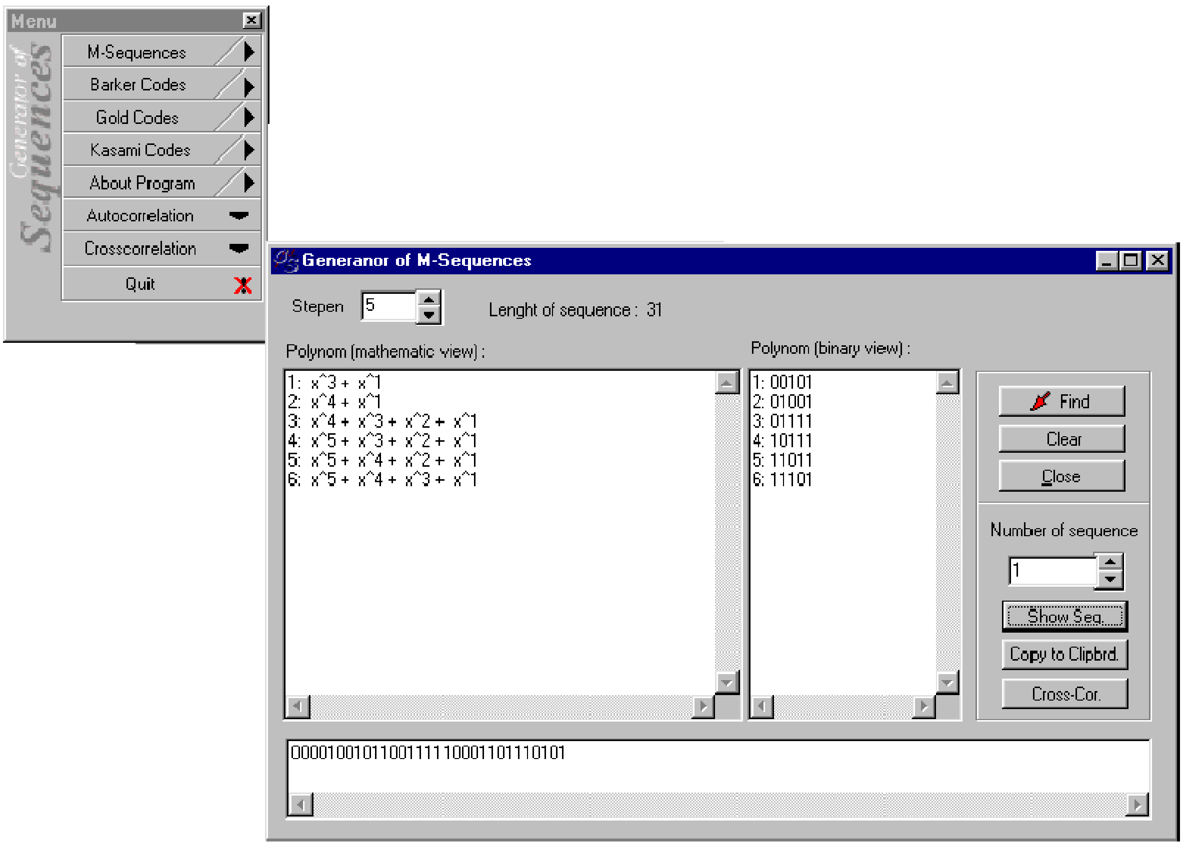
Рис. 17. Структурна схема приладу цифрового вводу сигналу кореляції

У пристрої реалізований асинхронний режим реєстрації з частотою дискретизації 40 *МГц* від власного тактового генератора.

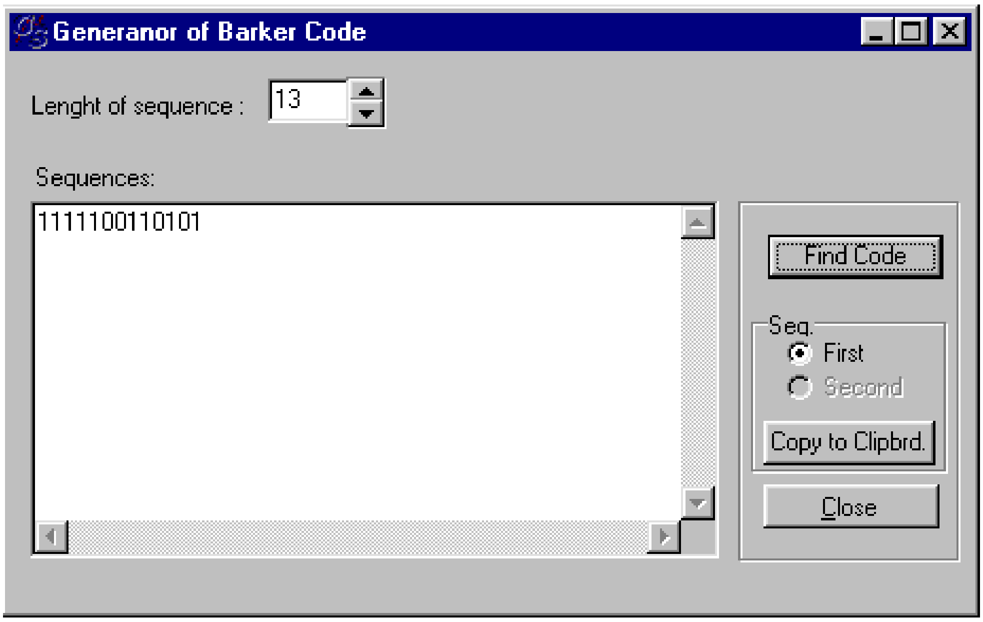
Враховуючи обсяг пам'яті 1 *кБайт* і частоту дискретизації 40 *МГц*, система дозволяє оцифрувати сигнал тривалістю 25 *мкс*, що як правило, достатньо для сигналу згортки.

Методичні вказівки до виконання лабораторної роботи

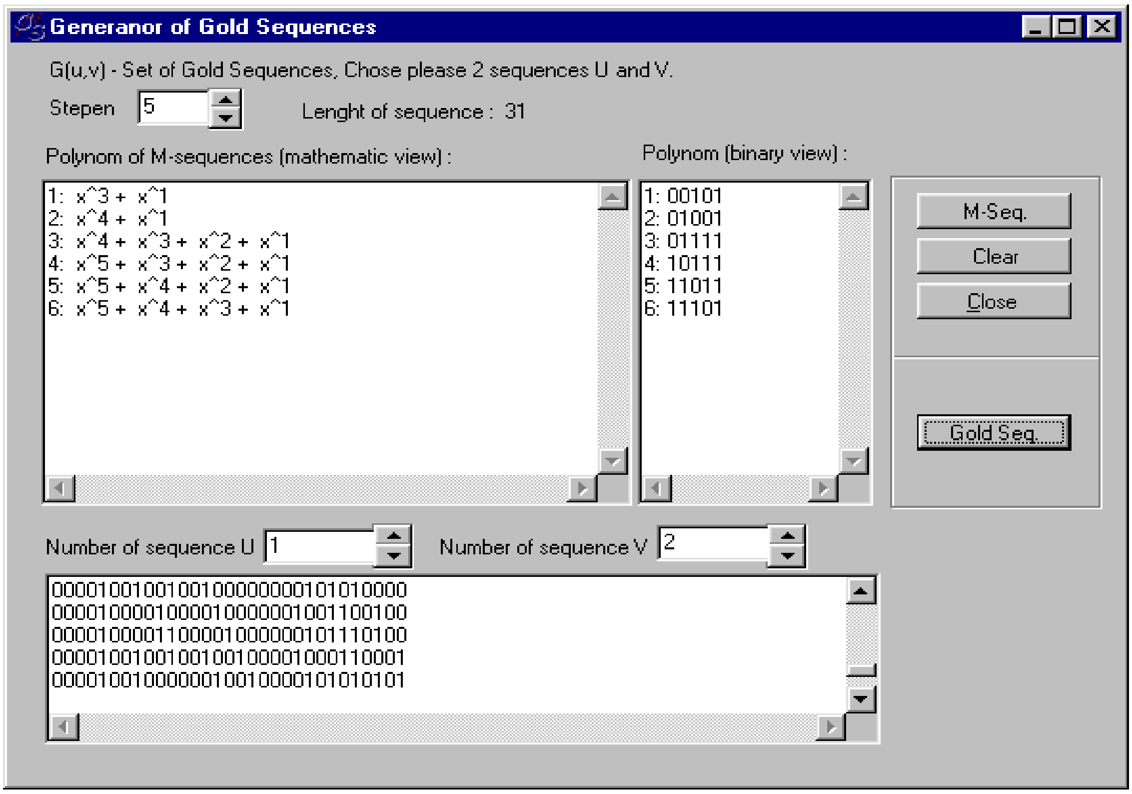
1) Знаходження М-послідовностей за допомогою програми за заданим ступенем поліномів, що породжують М-послідовності, та вигляд самих М-послідовностей.



2) Знаходження за допомогою програми кодів Баркера.

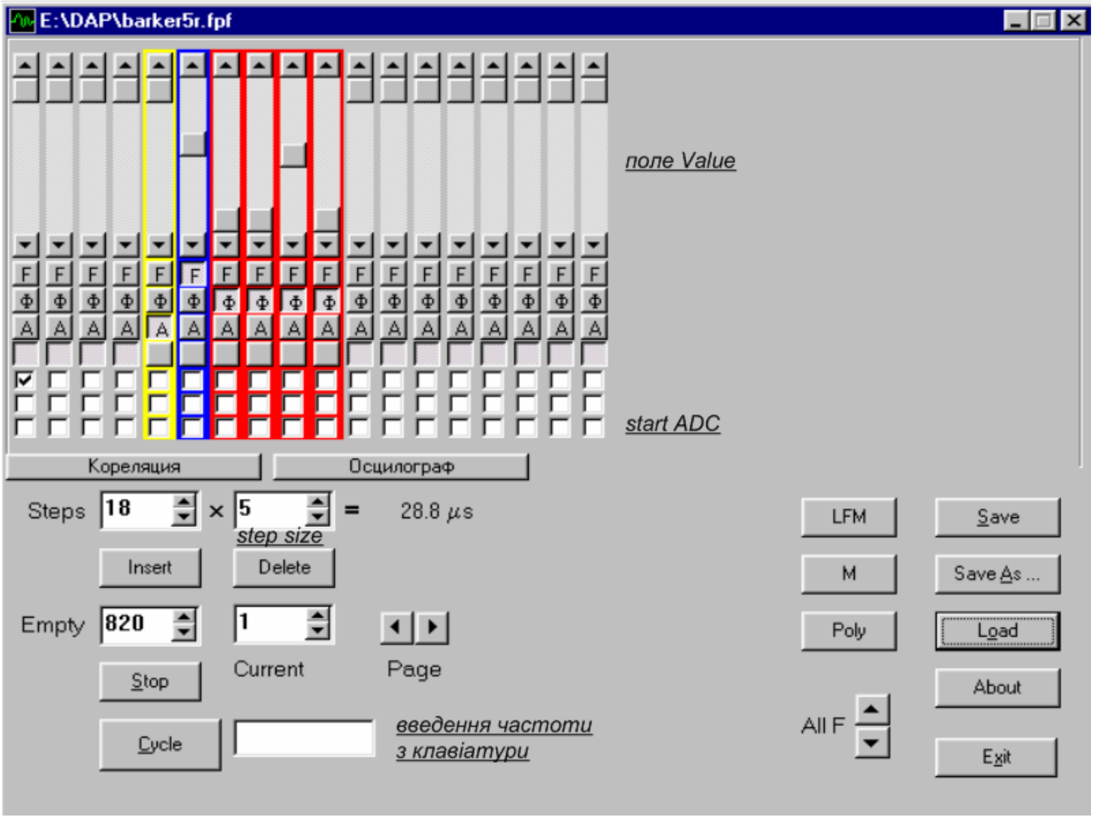


3) Знаходження множин послідовностей Голда та Касамі.



4) Генерація послідовностей.

Генерація послідовностей відбувається за допомогою програми керування синтезатором частоти. Головне вікно (вікно №1) виглядає наступним чином:



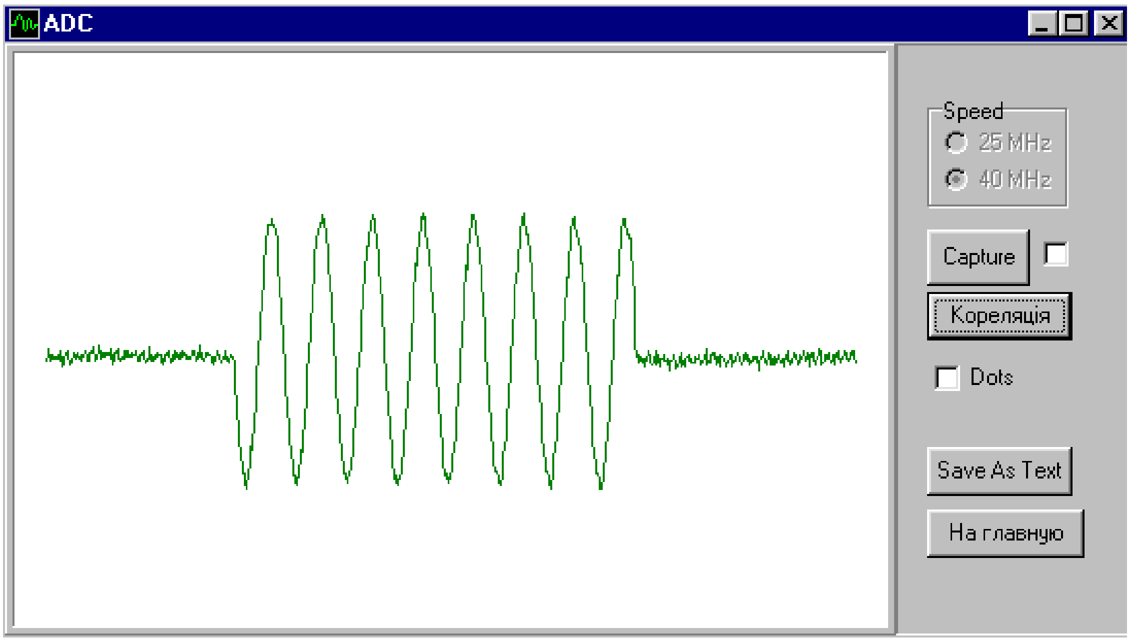
Кнопки «Insert» «Delete»	Додавання та віднімання кроку генерованій послідовності
Кнопки «Save» «Load»	Збереження та відкриття файлу з послідовністю
Поле «Steps»	Визначає кількість кроків
Поле «Step size»	Визначає довжину 1 кроку
Поле «Empty»	Визначає кількість порожніх кроків
Поле «Current»	Визначає номер поточного кроку
Кнопка «Cycle»	Початок циклічної генерації послідовності Перехід до вікна №2
Кнопка «M»	Ввід бінарної послідовності
Поле «Value» Вертикальний регулятор для кожного кроку	Значення (фази/ампулітуди/частоти) для поточного кроку
Перемикач(F/Ф/A) для кожного кроку	Вибирає величину, яка задається поточним кроком
Прапорець «Start ADC»	Початок роботи АЦП
Кнопка «Осцилограф»	Перехід до вікна №2
Кнопка «Кореляція»	Перехід до вікна №3

Для швидкого вводу бінарної послідовності можна скористатись кнопкою «M» і записати послідовність в поле, яке з'явиться. Якщо, наприклад, потрібно знайти періодичну АКФ, то в це поле вводиться два-три періоди послідовності. Бінарну послідовність можна також задати вручну за допомогою регуляторів поля "Value", але необхідно встановити амплітуду послідовності не більше 40%. Вибір між частотою, фазою та амплітудою задається відповідними перемикачами "F", "Ф", "A". Для початку генерації послідовності необхідно задати частоту. Її необхідно задати лише для початкового кроку, для наступних вона автоматично стане такою ж. Частоту можна задати з клавіатури в полі біля кнопки "Cycle".

Для запуску циклічної генерації створеної послідовності необхідно натиснути "Cycle", для зупинки – "Stop".

Але при виборі довжини та кількості кроків необхідно пам'ятати, що сумарна кількість заданих та порожніх кроків не повинна перевищувати 1024, а максимальна тривалість послідовності – 30 *мкс*. Останнє можна перевірити у вікні 2, що з'являється при натисканні кнопки "Осцилограф":

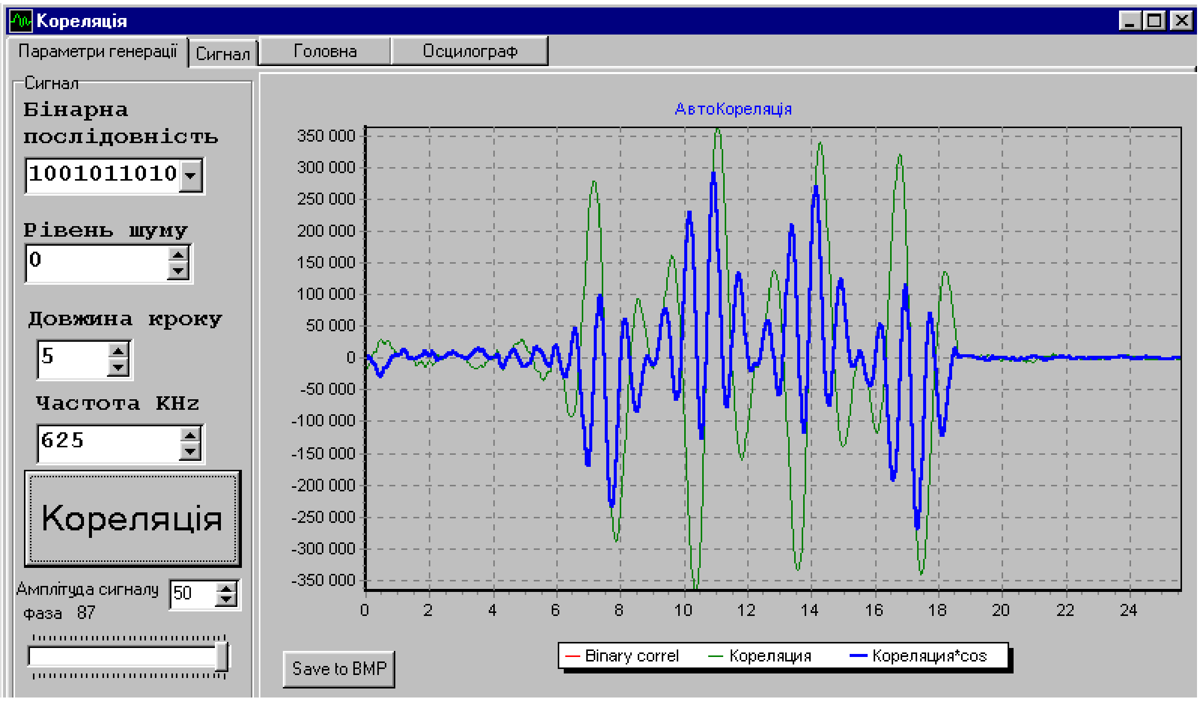
Вікно №2



Графік	Сигнал з АЦП з можливістю масштабування
Кнопка «Capture»	Повторне захоплення сигналу для АЦП
Кнопка «Кореляція»	Перехід до вікна №3
Кнопка «Save as text»	Збереження до текстового файлу
Кнопка «На главную»	Перехід до вікна №1

Натиснувши кнопку "Кореляція" переходимо до аналізу АКФ.

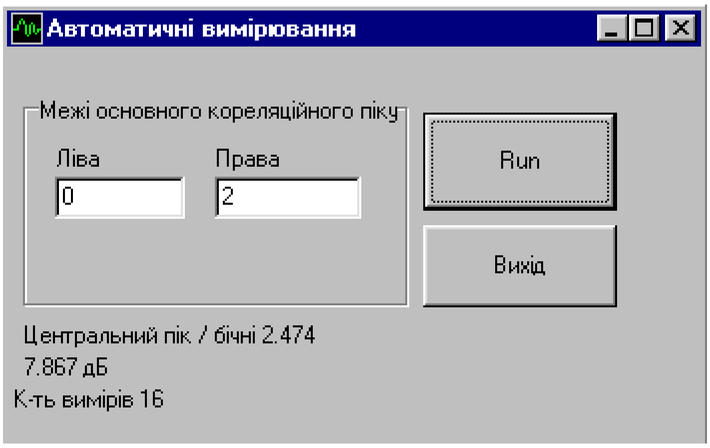
Вікно №3



Поле «Бінарна послідовність»	Задає опорну послідовність (ОП)
Поле «рівень шуму»	Задає відношення сигнал/шум для прийнятого з АЦП сигналу
Поле «Довжина кроку»	Задає довжину кроку ОП
Поле «Частота, KHz»	Задає частоту несучої для фазоманіпульваної (ФМ) ОП
Кнопка «Кореляція»	Розраховує кореляцію прийнятого сигналу з АЦП та ФМ ОП
Вкладка «Сигнал»	Відображає графік з прийнятим сигналом, прийнятим сигналом з шумом та ФМ ОП
Кнопка «Auto»	Перехід до вікна №4
Регулятор «Фаза»	Зміню зсув фаз між прийнятим сигналом та несучою для ОП
Кнопка «Save to bmp»	Зберігає осцилограму в графічний файл
Кнопка «Головна»	Перехід до вікна №1
Кнопка «Осцилограф»	Перехід до вікна №2

Поля "довжина кроку" та "частота" синхронізовані з відповідними полями у вікні №1, але їх можна змінювати. Кнопкою "Auto" можна робити статистику вимірів.

Вікно №4



Поля «Ліва» та «Права» в розділі «Межі основного кореляційного піку»	Визначає межі основного піку кореляційної функції
Кнопка «Run»	Запускає набір статистики з вимірів
Кнопка «Вихід»	Повертає до вікна №3
Напис «Центральний пік / Бічні »	Визначає середнє за час вимірювань відношення максимальної амплітуди в межах основного піку до максимальної амплітуди за його межами, також подається і в децибелах
Напис «К-ть вимірів»	Вказує кількість експериментів вибраних для статистики з різним значенням шуму

Перед запуском вибірки потрібно пересвідчитись, що всі поля у вікні №4 заповнені, а значення правої межі більші за значення лівої.

Завдання

1. Розрахувати вручну АКФ для послідовності Баркера (N на вибір викладача).
2. Дослідити залежність відношення центрального піку АКФ до бокових піків від довжини послідовності для
   a) Послідовностей Баркера (5,7,11,13)
   b) М-послідовностей (5-6 вимірів)
   c) Послідовностей Голда (-//-)
   d) Послідовностей Касамі (-//-)
2. Дослідити крос-кореляційні властивості послідовностей Голда та М-послідовностей. Порівняти їх.
3. Дослідити залежність відношення центрального піку АКФ до бокових піків від співвідношення сигнал-шум на вході для послідовностей Баркера (5,7,11,13) та для послідовності Баркера $N = 7$ з різною довжиною кроку послідовності (4-5 різних значень). Знайти рівень сигнал-шум при якому з'являються помилки (вибірка з 50 значень шуму)
4. Передати за допомогою послідовності Баркера ($N = 5$) сигнал 101 та замалювати графік кореляції.
5. Порівняти лінійну (1 сигнал генеруємо 1 – опорний) та циклічну (2-3 генеруємо 1 – опорний) автокореляцію для М-послідовності та послідовності Баркера (N на вибір викладача).
6. Дослідити АКФ послідовності Уолша (8 порядок матриці Адамара), та циклічну крос кореляцію цих послідовностей.

Контрольні питання

1. Що таке шумоподібні сигнали ?
2. Назвіть основні переваги систем зв’язку з ШПС.
3. В чому суть електромагнітної сумісності?
4. Що таке автокореляційна функція?
5. Яким критеріям повинна відповідати ідеальна АКФ?
6. Що таке білий шум і які його властивості (як випадкового процесу, АКФ)?
7. На що впливає гарна АКФ та ВКФ?
8. Які критерії використовують для псевдовипадкових послідовностей?
9. Чому необхідно збалансовувати псевдовипадкові послідовності?
10. Які основні види псевдовипадкових послідовностей?
11. В чому різниця між псевдовипадковими та ортогональними послідовностями?
12. Назвіть основні апаратні засоби для детектування ШПС.
13. Як впливає ширина спектру завади на прийнятий сигнал?

ДОДАТОК

ЛІСТІНГ ПОЧАТКОВОГО ПРОГРАМНОГО КОДУ ДО ДРУГОЇ ЧАСТИНИ ЛАБОРАТОРНОЇ РОБОТИ №2

```
uses Crt, Communit; // підключення додаткових модулів

// Опис використаних змінних
Var
  buf1,buf2:array[0..7] of byte; // buf1 – массив вихідних
                                 // повідомлень
                                 // buf2 – массив вхідних
                                 // повідомлень
  i,j:integer;
  c:char;
  err:byte;
  s1,s2:string;
  si:array[0..7] of byte;  // Масив бітів одного байту прийнятого
                           // повідомлення

Begin

// Формування початкових вихідних повідомлень (8 байтів)
 buf1[0]:=$FF;
 buf1[1]:=$FF;
 buf1[2]:=$FF;
 buf1[3]:=$FF;
 buf1[4]:=$FF;
 buf1[5]:=$FF;
 buf1[6]:=$FF;
 buf1[7]:=$FF;

// Основний цикл роботи програми, що містить передачу та прийом 8
   байтів інформаційних повідомлень

 Repeat

  ClrScr; // Очищення екрану

 // Виведення на екран 8 байт вихідних повідомлень. Процедура
 // BytetoBits перетворює кожен байт повідомлення у текстовий
 // рядок бітів, який виводиться на екран процедурою WriteLn
  for i:=0 to 7 do
   WriteLn(i,')  ',BytetoBits(buf1[i]));

 // Процедура передачі та прийому 8 байтів інформаційних повідомлень
 // через комунікаційний порт комп'ютера. Отримані повідомлення
 // знаходяться в масиві buf2
  SendAndReceive8(buf1,buf2);
```

```
   WriteLn; // Перехід на новий рядок екрану
// Цикл перебору та обробки кожного байту прийнятих повідомлень
   for i:=0 to 7 do
    begin
      s1:=Bytetobits(buf1[i]);  // Текстовий рядок бітів
                                // переданого повідомлення
      s2:=Bytetobits(buf2[i]);  // Текстовий рядок бітів
                                // прийнятого повідомлення

      err:=0;  // Обнуляємо кількість помилок

      // Перевіряемо кількість помилок в отриманому повідомленні,
      // порівнюючи передане та прийняте повідомлення побітно
      for j:=0 to 7 do
       if s1[j+1]<>s2[j+1] then
        err:=err+1; // При наявності неспівпадіння бітів збільшуємо
                    // кількість помилок на 1

      // Створюємо з текстового рядка бітів s2 масив бітів si у числовому
      // форматі для можливості математичної обробки бітів прийнятого
      // повідомлення
      for j:=0 to 7 do
       if s2[j+1]='1' then si[j]:=1 else si[j]:=0;

//===========================================================
//   Тут можна вставити процедуру обробки даних в масиві si.
//   Після корекції даних в масиві si, необхідно з цих даних створити
//   новий текстовий рядок бітів, що відповідають виправленому
//   повідомленню, який можна буде вивести на екран
//===========================================================

      // Вивод на екран отриманих повідомлень та кількості помилок в
      // них у вигляді таблиці. В цю таблицю можна буде також додати
      // виправлені повідомлення після їх корекції
      WriteLn(i,') ',s2,'    ',err);

    end; // Кінець циклу

   delay(1000); // Затримка роботи програми на 1 секунду

   c:=readkey; // Очікування натискання клавіші на клавіатурі
  until c=#27; // Вихід з програми, якщо натиснута клавіша "ESC"

end.
```

ЗМІСТ

Стор.

Вступ. 3

Лабораторна робота №1. Пряме цифрове перетворення частоти................ 4

Лабораторна робота №2. Ефективне завадостійке кодування інформації в цифрових каналах зв'язку 14

Лабораторна робота №3. Кореляційний прийом цифрових сигналів в каналі з шумом.. 46

Додаток. 74

Навчальне видання

КОЛЄНОВ *Сергій Олександрович*

ЦИФРОВИЙ ЗВ'ЯЗОК

Методичний посібник до лабораторного практикуму для студентів радіофізичного факультету

Підписано до друку __________. Формат $60x80^{16}$.
Гарнітура Times. Папір офсетний. Друк офсетний.
Наклад 30 примірників. Ум. друк. арк. 4.

Видавнича лабораторія радіофізичного факультету
Київського національного університету імені Тараса Шевченка