Тернопільський національний економічний університет

На правах рукопису

УДК 004.934

Шевчук Руслан Петрович

БАГАТОКАНАЛЬНІ КОМП'ЮТЕРНІ ЗАСОБИ ПЕРЕТВОРЕННЯ ФОРМАТІВ ТА КРИПТОГРАФІЧНОГО ЗАХИСТУ СТИСНЕНИХ МОВНИХ СИГНАЛІВ

05.13.05 – комп'ютерні системи та компоненти

Дисертація на здобуття наукового ступеня кандидата технічних наук

Науковий керівник
Мельник Анатолій Олексійович
доктор технічних наук, професор

Тернопіль – 2008

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

CELP - code Excited Linear Predictive Coder (збудження на основі кодової книги)

MOS - mean opinion score (суб’єктивна оцінка якості мовлення)

MPE - multiband Excitation (багатоімпульсне збудження)

RPE - regular-Pulse Excitation (регулярне імпульсне збудження)

АДІКМ – адаптивно-диференціальна імпульсно-кодова модуляція

АКК – адаптивна кодова книга

БКМД - багатоімпульсне квантування з максимальною достовірністю

БМ – блок мікшування

ВихКом – вихідний комутатор

ВхКом – вхідний комутатор даних

ВТ – висота тону

ДІКМ – диференціальна імпульсно-кодова модуляція

ДМ – дельта модуляція

ІКМ – імпульсно-кодова модуляція

КЛП – коефіцієнти лінійного прогнозування (linear prediction coefficients (LPC))

КС – комбінаційна схема

ЛП – лінійне прогнозування

ЛПАКСС – лінійне прогнозування, що генерується алгебраїчним кодом спряженої структури

ЛПГВС – лінійне прогнозування, що генерується векторною сумою

ЛСП – лінійні спектральні пари (linear spectral pairs (LSP))

МС – мовний сигнал

ОП – операційний пристрій

ПрК – пристрій керування

ПрКом – проміжний комутатор

ФКК – фіксована кодова книга

ЦОС – цифрове оброблення сигналів

ВСТУП

Сучасний етап розвитку комп’ютерних систем та їх висока гетерогенність вимагають від мережного обладнання, що працює з різними форматами даних, чіткої взаємодії та можливості в реальному часі гарантувати захищену передачу даних з одного сегменту мережі в інший. Відомо, що сьогодні розроблено велику кількість протоколів, які регламентують передачу інформаційних сигналів різних форматів. Зокрема, тільки форматів стиснення мовних сигналів розроблено декілька десятків, і всі вони структурно та семантично різняться між собою [29, 50, 74, 132]. Тому часто виникають ситуації, при яких приймач повідомлення не в змозі їх відтворити, оскільки не має належних засобів для цього. Часто, під час передачі комп’ютерними мережами втрачається інформаційний зміст даних, оскільки процес передачі слабо захищений від можливих загроз щодо конфіденційності, цілісності та автентичності даних. Для запобігання та уникнення таких ситуацій в комп’ютерні системи вмонтовують засоби перетворення та криптографічного захисту форматів сигналів [40,75,100,107,119,120,121]. Одним з найбільш перспективних на сьогоднішній час напрямів є розробка багатоканальних комп’ютерних засобів перетворення та криптографічного захисту форматів стиснених мовних сигналів, які вмонтовують у комп’ютерні системи IP-телефонії, комп’ютерної телефонії, мультимедіа-конференцій, стільникового зв’язку та спеціалізованих систем зв’язку.

Переважна більшість засобів перетворення форматів стиснених мовних сигналів апаратно реалізовані та працюють згідно з класичним методом - тандем кодера-декодера (тандем) [69,88]. Використання даного методу супроводжується значними часовими затримками, високою складністю оброблення потоків даних в реальному масштабі часу та втратою якості мови [88,107,121]. Це зумовлено структурними особливостями алгоритмів стиснення мовних сигналів та необхідністю виконання процесів декодування і кодування блоків даних із мовними сигналами, що призводить до накопичення помилок квантування.

Передача стиснених мовних сигналів незахищеними комп’ютерними мережами повинна проводитись із врахуванням можливих загроз щодо

конфіденційності, цілісності та автентичності мовної інформації [4,11,21,35,39,55]. У сучасних комп'ютерних мережах широко використовуються протоколи захищеної передачі даних, у яких вирішення завдань конфіденційності, цілісності та автентичності інформації досягається шляхом криптографічного захисту даних [13,19]. Криптографічний захист стиснених мовних сигналів передбачає використання спеціальних засобів, методів та заходів для вирішення наступних завдань [4,39]: запобігання втраті кадрів з інформацією про мовний сигнал під час їх передачі від джерела до приймача, що можуть бути спричинені навмисними або ненавмисними діями чи спотвореннями в каналі зв'язку; запобігання просочуванню мовної інформації за рахунок несанкціонованого прослуховування переговорів в каналах телекомунікацій; запобігання зміні мовного повідомлення за допомогою модифікації сеансу мовних фраз або особистих особливостей учасника сеансу зв'язку. Найчастіше для комплексного вирішення перелічених вище завдань використовують протокол IPSec, який застосовується на транспортному рівні OSI та дає змогу забезпечити цілісність, автентичність та конфіденційності даних, що передаються незахищеними мережами [103]. Однак, невисока продуктивність роботи програмованих процесорів, ресурси яких використовуються для реалізації протоколу IPSec зменшує можливості мультимедійних додатків та комп'ютерних засобів. Крім того, додавання службових полів цим протоколом до результуючого кадру даних значно збільшує його розмір.

Значний вклад у розвиток засобів перетворення форматів стиснених мовних сигналів внесли: Канг Х.Г, Беаджент К, Тадей Х, Кокс Р.В., Шу-Мін Тсай, Яр-Фер Янг, Бесціер Л. Багато наукових робіт присвячено пошуку нових шляхів стиснення мовних сигналів. Зокрема, необхідно відзначити значний внесок відомих вчених: Котельникова В.А, Шенона К, Рафінера Л.Р, Шафера Л.В, Макхоула Дж, Грея А., Семенова В. Ю, Ватоліна Д. та інші. Алгоритми мікшування мовних сигналів досліджувались у роботах: Ренжен П.В, Гарік М, Ременейшн В.С, Хусейн А.В, Раденковіч М, Хедл Р. Спеціальні питання розробки комп'ютерних засобів захисту інформації досліджувались у роботах Мельника А.О., Хорошко В.О., Задіраки В.К., Николайчука Я.М., Карпінського М.П., Дворянкіна С.В., Петракова А.В.

Проте дослідження у цьому напрямі не втрачають своєї актуальності, оскільки залишилось чимало невирішених проблем. Зокрема, відсутня класифікація існуючих алгоритмів стиснення та мікшування мовних сигналів, яка враховувала б особливості їх побудови та можливості використання у процесі перетворення з одного формату в інший. Існуючі багатоканальні комп'ютерні засоби перетворення форматів стиснених мовних сигналів базуються на класичному методі – тандем кодера-декодера, що призводить до значних часових затримок, високої складності оброблення потоків даних в реальному масштабі часу та втрати якості мови. Відомі структури багатоканальних комп'ютерних засобів перетворення форматів стиснених мовних сигналів не враховують особливості роботи алгоритмів стиснення та мікшування. При передачі стиснених мовних сигналів незахищеними комп'ютерними мережами не враховуються загрози щодо конфіденційності, цілісності та автентичності даних.

Таким чином, актуальним є завдання розробки багатоканальних комп'ютерних засобів перетворення та криптографічного захисту форматів стиснених мовних сигналів, що враховують основні закономірності і особливості функціонування комп'ютерних систем реального часу.

Робота присвячена теоретичному аналізу та експериментальному дослідженню багатоканальних комп'ютерних засобів перетворення форматів та криптографічного захисту стиснених мовних сигналів.

Вирішення наведених вище проблем забезпечить підвищення ефективності та захищеності передачі мовних сигналів, а також зменшить результуючі затримки між територіально віддаленими джерелами багатоканальних комп'ютерних систем реального часу.

Зв'язок роботи з науковими програмами, планами, темами. Напрямок виконаних дисертаційних досліджень безпосередньо пов'язаний з науково-дослідним напрямком кафедри "комп'ютерних наук" Тернопільського національного економічного університету протягом 2002 - 2008 років. Дисертаційна робота безпосередньо пов'язана з:

- науково-дослідною роботою МОЕСП-61-02”К” “Розробка теоретичних засад, алгоритмічного та програмного забезпечення моделювання технічних, екологічних та економічних систем на основі аналізу інтервальних даних", (номер державної реєстрації 0102U002565 (2002-2006 рр.);

- науково-дослідною роботою “Співпраця між Україною та Румунією в галузі розподілених систем (CobURDiS)”, (номер державної реєстрації 0106U005307 (2006-2007 рр.).

Мета і завдання дослідження. Метою дисертаційної роботи є розробка методів та багатоканальних комп’ютерних засобів для підвищення продуктивності перетворення форматів і криптографічного захисту стиснених мовних сигналів в реальному масштабі часу.

Для досягнення поставленої мети необхідно розв’язати наступні взаємопов’язані завдання:

- порівняльний аналіз алгоритмів, що використовуються у процесі перетворення форматів стиснених мовних сигналів з метою узагальнення їх структурних особливостей;
- порівняльний аналіз комп’ютерних засобів перетворення форматів стиснених мовних сигналів та визначення областей їх доцільного використання;
- дослідження методів перетворення та мікшування форматів стиснених мовних сигналів;
- формалізація математичних моделей та створення структур багатоканальних комп’ютерних засобів перетворення форматів стиснених мовних сигналів;
- удосконалення структур операційних пристроїв криптографічних модулів апаратно-орієнтованих процесорів підтримки протоколу IPSec, орієнтованих на захист мовних сигналів;
- дослідження багатоканальних комп’ютерних засобів перетворення форматів стиснених мовних сигналів.

Об’єкт дослідження: перетворення форматів стиснених мовних сигналів у багатоканальних комп’ютерних системах реального часу.

Предмет дослідження: методи та комп'ютерні засоби перетворення форматів та криптографічного захисту стиснених мовних сигналів.

Методи дослідження: основні наукові результати і висновки, одержані на основі теорії інформації, теорії цифрових автоматів, теорії кодування, моделюванні алгоритмів і апаратних засобів комп'ютерів та експериментальних досліджень.

Наукова новизна одержаних результатів.

1. Запропоновано метод перетворення форматів стиснених мовних сигналів між G.723.1 та G.729A, який на відміну від відомих дає можливість виконувати пряме перетворення ряду параметрів кадру одного формату у параметри кадру іншого та передбачає виконання чотирьох етапів, зокрема, перетворення лінійних спектральних пар, перетворення висоти тону і пошук у адаптивній та фіксованій кодових книгах. Розроблений метод дозволяє зменшити часову затримку, апаратну складність декодера та покращити якість мовлення.

2. Запропоновано метод перетворення форматів стиснених мовних сигналів між GSM 06.20 та G.729A, який на відміну від відомих враховує структурну подібність модулів короткотермінової фільтрації, довготермінової фільтрації та випадкового збудження алгоритмів лінійного прогнозування, що генерується векторною сумою, та лінійного прогнозування, що генерується алгебраїчним кодом спряженої структури, яка дає можливість провести пряме перетворення параметрів, згенерованих даними модулями. Розроблений метод передбачає виконання трьох етапів, зокрема, перетворення коефіцієнтів лінійного прогнозування, перетворення висоти тону і швидкий пошук у фіксованій кодовій книзі та дозволяє зменшити часову затримку і апаратну складність.

3. Вперше запропоновано метод багатоступінчастого мікшування мовних сигналів на основі пам'яті з довільним доступом, відповідно до якого процес мікшування починається при надходженні хоча б двох блоків даних у цільову комірку пам'яті з довільним доступом, що дозволяє уникати черг у буфері блоку мікшування та зменшити затримки пов'язані з часом очікування блоків даних.

4. Удосконалено структури операційних пристроїв криптографічних модулів апаратно-орієнтованих процесорів підтримки протоколу IPSec для різних сервісів

оброблення даних за різних технологічних характеристик компонентного базису, що забезпечило зменшення затрат обладнання на їх реалізацію.

Практичне значення одержаних результатів

1. Розроблено програмне забезпечення для транскодування стиснених звукових та мовних сигналів, яке дає змогу здійснювати перетворення між такими форматами: WMA Voice Encoder DMO, WM Speech Encoder DMO, WM Audio Encoder DMO, 3ivx D4 Audio Encoder, Indeo Audio Software, Pinnacle AC3 Encoder, Pinnacle AC3 Encoder, Pinnacle MP3 Encoder, Pinnacle MPEG Layer-2 Audio Encoder, Vorbis Encoder, IMC, IAC2, IMA ADPCM, PCM, Ogg Vorbis, Microsoft ADPCM, ACELP.net, DSP Group TrueSpeech, Windows Media Audio, GSM 06.10, G.723.1, CCITT A-Law, CCITT u-Law, AC-3 ACM Codec, MPEG Layer-3.

2. Розроблено програмне забезпечення виконання алгоритмів багатоімпульсного квантування з максимальною достовірністю та лінійного прогнозування, що генерується алгебраїчним кодом спряженої структури для процесорів типу TMS320C6201. Програмні реалізації повністю сумісні з вимогами стандартів ITU-T G.723.1 та ITU-T G.729A (досягнута побітова відповідність тестовим векторам; функціонування в режимі реального часу; сумісність зі всіма цифровими процесорами типу TMS320C6201).

3. На основі використання запропонованого методу транскодування між G.723.1 та G.729A розроблено програмну реалізацію удосконаленого транскодера для процесора типу TMS320C6201. Реалізація забезпечує кращу якість мовлення (0,3%-1,7% - під час перетворення форматів стиснених МС з G.729A до G.723.1 (5,3 Кб/с); 0,3%-6,4% - під час перетворення форматів стиснених МС з G.729A до G.723.1 (6,3 Кб/с)) та меншу апаратну складність (25,9%-51,6%) у порівнянні з класичним методом.

4. На отриманій теоретичній основі та практичних рекомендаціях створено цільове програмне забезпечення для удосконалення структур операційних пристроїв криптографічних модулів апаратно-орієнтованих процесорів підтримки протоколу IPSec, що дало змогу отримати структури з мінімальними апаратними затратами для різних довжин кадрів.

Теоретичні та практичні результати дисертаційної роботи використані та впроваджені при виконанні науково-дослідної роботи МОЕСП-61-02”К” “Розробка теоретичних засад, алгоритмічного та програмного забезпечення моделювання технічних, екологічних та економічних систем на основі аналізу інтервальних даних” (номер державної реєстрації 0102U002565), результати роботи використані в науково-дослідній роботі “Співпраця між Україною та Румунією в галузі розподілених систем (CobURDiS)” (номер державної реєстрації 0106U005307).

Отримані результати дисертаційних досліджень використані у розробках КБ “Стріла” (м. Тернопіль). За результатами проведених досліджень впроваджено: метод перетворення стиснених мовних сигналів між форматами G.723.1 та G.729A для підвищення ефективності використання каналів зв’язку між територіально віддаленими джерелами формування сигналів багатоканальних комп’ютерних систем реального часу; метод багатоступінчастого мікшування на базі пам’яті з довільним доступом для мікшування мовних сигналів по мірі їх поступлення у багатоканальні засоби комп’ютерних систем; удосконалені структури операційних пристроїв криптографічних модулів апаратно-орієнтованих процесорів підтримки протоколу IPSec для забезпечення захисту сигналів, що передаються у системах зв’язку реального часу.

Теоретичні та експериментальні результати досліджень впроваджено у навчальний процес на кафедрі “комп’ютерних наук” Тернопільського національного економічного університету з дисциплін “Програмне забезпечення мультимедіа”, “Методи та засоби захисту програмного забезпечення” і “Методи та засоби вимірювання та цифрової обробки інформації”.

Особистий внесок здобувача. Дисертаційна робота є результатом самостійної роботи автора. У працях опублікованих у співавторстві, здобувачу особисто належать: [20] - виділено базові операції алгоритмів MD5 і SHA-1 та на їх основі побудовано структури операційних пристроїв хешування; виділено ряд граф-алгоритмічних операційних пристроїв та отримано аналітичні вирази, що описують їх часові характеристики; [24,105] - запропоновано математичну модель визначення часових затримок на основних елементах рекурсивної архітектури, що дало змогу

отримати аналітичні залежності часових затримок від кількості паралельних мультимедіа конференцій із змінною кількістю учасників. У роботі [23] запропоновано математичну модель операційного пристрою процесора підтримки протоколу IPSec, на основі якої розроблено програмне забезпечення для удосконалення структур операційних пристроїв криптографічних модулів процесорів підтримки протоколу IPSec. У роботах: [29] - проведено класифікацію алгоритмів стиснення мовних сигналів, окреслено сучасний стан галузі стиснення мовних сигналів та виділено основні напрямки її розвитку; [30,112] - формалізовано математичну модель та створено структуру багатоканального транскодера стиснених мовних сигналів; [31,111] - запропоновано метод багатоступінчастого мікшування та створено структуру блоку мікшування для його реалізації; [60] - створено модель багатоканального транскодера між G.729A і G.723.1 на базі цифрового сигнального процесора типу TMS320C6201; [22] - проведено аналіз архітектур багатоабоненських мультимедіа-конференцій, на підставі якого запропоновано рекурсивну архітектуру; [61] - проведено порівняльний аналіз протоколів організації мультимедіа-конференцій. [113] - досліджено метод багатоступінчастого мікшування; [135] - запропоновано метод перетворення форматів стиснених мовних сигналів між G.729A і GSM 06.20.

Апробація результатів дисертації. Основні положення та результати дисертаційної роботи доповідались та обговорювались на 13-ти міжнародних і національних конференціях: “Проблеми інформатики і моделювання” Харків, 2003; “Сучасні проблеми радіоелектроніки, телекомунікацій та комп’ютерної інженерії” Львів-Славсько, 2004 та 2006; науковій конференції професорсько-викладацького складу, докторантів, аспірантів, здобувачів наукових ступенів “Економічні, правові, інформаційні та гуманітарні проблеми розвитку України в постстабілізаційний період”, Тернопіль 2004-2008; “Наукова конференція Тернопільського технічного університету” Тернопіль, 2004; “Досвід розробки та застосування САПР в мікроелектроніці” Львів-Поляна, 2005 та 2007; “Інтелектуальні засоби збору даних і сучасні обчислювальні системи: розробка і застосування” Софія (Болгарія), 2005 та Дортмунд (Німеччина), 2007.

Публікації. Результати, отримані за час досліджень, опубліковані в шістнадцяти наукових працях, з яких 8 статей у наукових фахових виданнях, одна з яких є одноосібною, 8 тез доповідей в матеріалах конференцій.

Структура та обсяг дисертації. Дисертаційна робота складається із вступу, переліку умовних скорочень, п’яти розділів, висновків, переліку використаних джерел та 8 додатків. Основний зміст викладений на 179 сторінках. Робота містить 45 рисунків, 33 таблиці. Список використаних джерел із 154 найменувань. Додатки на 28 сторінках.

РОЗДІЛ 1
АНАЛІЗ ОСОБЛИВОСТЕЙ ПОБУДОВИ БАГАТОКАНАЛЬНИХ ЗАСОБІВ ПЕРЕТВОРЕННЯ ФОРМАТІВ СТИСНЕНИХ МОВНИХ СИГНАЛІВ

1.1. Перетворення форматів та криптографічний захист стиснених мовних сигналів

1.1.1. Основні визначення та принципи перетворення форматів та криптографічного захисту стиснених мовних сигналів

Нехай S буде визначати множину, що називається мовним сигналом (МС). Множина S складається з цифрових значень визначеної розмірності (відліків) q_x, які визначаються після виконання процесів дискретизації та квантування над аналоговим сигналом, який описує амплітуду хвилі МС [17,139].

Дискретизація базується на теоремі Котельникова, відповідно до якої МС з обмеженим спектром частот представляється множиною відліків з тактовою частотою $f=1/T$, яка принаймні вдвічі перевищує найвищу частоту спектру МС [8,9,22, 23,24,25]. Процес дискретизації виконує дискретизатор, який складається з фільтра нижніх частот та дає змогу усунути спектральні компоненти МС з частотами більшими $f/2$ [12,17]. Якщо частота дискретизації менша подвоєної максимальної частоти спектру вхідного сигналу, то виникає ефект накладання спектрів (аліасинг) [17].

Квантування – округлення значення аналогового сигналу до одного із наперед встановлених рівнів квантування [41,42,130]. Процес квантування виконується квантувальником. Виділяють рівномірне та нерівномірне квантування [12,41,42, 130]. При рівномірному квантуванні діапазон значень МС x_{max}-x_{min} розбивається на $M=2^R$ рівнів, що відрізняються один від одного на крок квантування Δ, так що [12]:

$$x_{max}\text{-}x_{min} < M*\Delta \qquad (1.1)$$

де x_{max} – максимальне значення вхідної величини;

x_{min} – мінімальне значення вхідної величини;

M – кількість рівнів квантування;

Δ – крок квантування.

Значення R визначає розрядність квантувальника. Результатом роботи квантувальника є кодове слово, що отримується після представлення квантованих значень відліків набором двійкових символів [12,42]. У процесі квантування виникають помилки квантування (шум квантування) $e(n)$, які рівні різниці між квантованим та істинним значенням МС. При рівномірному квантуванні $e(n)$ зростає при низьких рівнях МС, що пояснюється зменшенням його потужності. У більшості компонентів комп'ютерних систем виконується нерівномірне квантування, у якому при низьких рівнях МС помилка квантування менша, ніж на високих [12]. При зміні рівня МС відношення "сигнал-шум" зберігають сталим.

Нехай C визначає множину, що називається стисненим МС. Значення c_x ($c_x \in C$) називають відліками стисненого МС.

Значення відліків множин C і S ідентифікуються порядковими номерами x ($x=1,\ldots,k$), де k – кількість відліків.

Нехай F позначає множину, що називається простором форматів стиснених МС. Одним із ідентифікаційних параметрів формату стисненого МС є блок даних $L_{i,j}(c_x)$ (де i – порядковий номер блоку; j – номер джерела від якого одержано i-тий блок; $i=1,\ldots,s$; $j=1,\ldots,N$; s – кількість блоків створених j-им джерелом; N – кількість джерел) [34]. Відповідно до алгоритму стиснення МС a $(a \in F)$, значення q_x перетворюються у значення c_x. Алгоритм стиснення МС визначає послідовність виконання операцій над блоками даних під час стиснення чи декомпресії [74]. Довжина та структура блоків даних $L_{i,j}(c_x)$ залежить від алгоритму стиснення МС, що використовується джерелом j.

K_a позначає функцією стиснення або кодування, D_a - функція декомпресії або декодування.

Процес застосування перетворення K_a до відліків $q_x \in S$ називають стисненням (кодуванням, компресією) МС. Комп'ютерні засоби, що виконують стиснення, називають кодерами або компресорами [74]. Елементи множини C відображають результат стиснення.

Процес застосування перетворення D_a до відліків $c_x \in C$ називають декомпресією (декодуванням) МС. Комп'ютерні засоби, що виконують декомпресію, називаються декомпресорами або декодерами [74]. Елементи множини S будуть відображати результат декомпресії.

Комп'ютерні засоби стиснення та декомпресії називають кодеками алгоритму a [12,74].

Якщо порядкові номери відліків q_x та c_x i-тих блоків даних співпадають, то, у більшості випадків, значення відліку q_x буде схожим до значення відліку c_x.

Процес перетворення блоків даних $L_{i,j}(c_x)$, сформованих згідно функції D_a, в блоки даних $L_{i,j}(c_x)$ сформований згідно функції D_b, коли $a \neq b$ $(a,b \in F)$, називається перетворенням форматів стиснених МС або транскодуванням стиснених МС [40,75,100,107,119-121]. Комп'ютерні засоби, що виконують транскодування стиснених МС, називають транскодерами стиснених МС [75,100,107,121].

Часто у багатоканальних системах реального часу, в процесі транскодування стиснених МС, виконується мікшування [6,36,43,82,127].

При мікшуванні значення відліків q_x з i-тих блоків даних змішуються між собою відповідно до алгоритму u, що є елементом множини U $(u \in U)$ [36,82,127]. Мікшування виконується коли активними є, як мінімум, два джерела формування блоків даних $L_{i,j}(c_x)$ [31]. M_i^u називають функцією мікшування. Результатом мікшування є блоки даних $L_{i,j}(p_x)$, $(p_x \in S)$.

Інформація про технічні характеристики джерел (приймачів) стиснених МС та про пропускну спроможність каналів зв'язку передається у повідомленнях V_j та H_j (де j – ідентифікатор джерела (приймача) стиснених МС). Передача повідомлень V_j та H_j виконується відповідно до протоколів сигналізації [6,43]. Основними функціями протоколів сигналізації є встановлення, адміністрування та завершення сеансу зв'язку між кінцевими точками, що однозначно ідентифікуються заданою схемою адресації [43]. Найбільш популярними протоколами сигналізації є ОКС №7, R2, DSS1, RAS, Q.931, H.245 [9,43].

Коди керування w є елементами множини W та виконують керування транскодуванням стиснених МС.

Структурна схема транскодування стиснених МС складається з набору $\{D_a : a \in F\}$ перетворень декодування, набору $\{M_i^u : u \in U\}$ перетворень мікшування та набору $\{K_b : b \in F\}$ перетворень кодування, а також наборів S, C, W, F, U.

Структура та значення відліків h_x повинні відповідати вимогам встановленим алгоритмом $b \in F$ до значень відліків які стискаються. Створення схеми транскодування стиснених МС потребує вибору простору МС S, простору стиснених МС C, простору форматів стиснених МС F, простору алгоритмів мікшування МС U, простору сигналів керування W, набору перетворень декодування $\{D_a : a \in F\}$, набору перетворень мікшування $\{M_i^u : u \in U\}$ та набору перетворень кодування $\{K_b : b \in F\}$.

Транскодування стиснених МС забезпечує сумісність гетерогенних комп'ютерних систем, що обмінюються кадрами із блоками даних $L_{i,j}(c_x)$ [75,100,107,121].

Часто, під час передачі кадрів із блоками даних $L_{i,j}(c_x)$ комп'ютерними мережами втрачається їх інформаційний зміст, оскільки процес передачі слабо захищений від можливих загроз щодо конфіденційності, цілісності та автентичності даних [4,11,21,26,39,55,58,61]. Для запобігання та уникнення таких ситуацій в обладнання комп'ютерних мереж вмонтовують засоби криптографічного захисту. Одним з найбільш перспективних типів захисту даних у каналах зв'язку є створення віртуальних приватних мереж, де вирішення завдань конфіденційності, цілісності та автентичності інформації досягається шляхом криптографічного захисту даних [13,16,19].

Криптографічний захист кадрів із блоками даних $L_{i,j}(c_x)$ у каналах зв'язку передбачає використання спеціальних засобів, методів та заходів для вирішення наступних завдань [4,39]: запобігання втратам кадрів із блоками даних $L_{i,j}(c_x)$ під час їх передачі, що можуть бути спричинені навмисними або ненавмисними діями і спотвореннями в каналі зв'язку; запобігання витоку мовної інформації за рахунок несанкціонованого прослуховування переговорів в каналах зв'язку; запобігання модифікації мовних фраз або індивідуальних особливостей учасника сеансу зв'язку для підміни мовного повідомлення.

Визначимо умови, при яких доцільно виконувати перетворення форматів та криптографічний захист стиснених МС [30]:

- приймач не має засобів для відтворення блоків даних $L_{i,j}(c_x)$;
- технічні характеристики приймача кадрів із блоками даних $L_{i,j}(c_x)$ не дозволяють в реальному масштабі часу відтворити одержаний МС;
- існують загрози щодо конфіденційності, цілісності та автентичності інформації із блоків даних $L_{i,j}(c_x)$.

1.1.2. Схема транскодування стиснених мовних сигналів

Схема транскодування стиснених МС складається з набору перетворень декодування $\{D_a : a \in F\}$, мікшування $\{M_i^u : u \in U\}$ та кодування $\{K_b : b \in F\}$.

Зв'язок джерел та приймачів блоків даних $L_{i,j}(c_x)$ із використанням схеми транскодування стиснених МС полягає в наступному (рис. 1.1).

Схему транскодування стиснених МС зображену на рис. 1.1 називають традиційною схемою транскодування стиснених МС. Транскодування блоків даних $L_{i,j}(c_x)$ виконується відповідно до класичного методу - тандем кодера-декодера (classical encoder-decoder tandem approach) або просто тандем [76,88].

Перед початком транскодування стиснених МС, джерела та приймачі блоків даних $L_{i,j}(c_x)$ передають транскодеру повідомлення V_j та H_j.

Після встановлення сеансу зв'язку між джерелом та приймачем кадрів із блоками даних $L_{i,j}(c_x)$. Часто, процес передачі кадрів відбувається через незахищені канали зв'язку. Для комплексного захисту кадрів у процесі передачі використовується протокол H.235, що є частиною стандарту H.323 [93]. Однак даний протокол має ряд недоліків: недосконалі механізми захисту передачі кадрів від мобільних терміналів [152], не регламентовані механізми безпечної передачі кадрів згідно з протоколами серій H.26X та T.12X [142]. Тому, для комплексного захисту кадрів із блоками даних $L_{i,j}(c_x)$ запропоновано використовувати протокол IPSec, який працює на транспортному рівні моделі взаємодії відкритих систем та дає змогу зменшити недоліки протоколу H.235 [103].

Рис. 1.1. Схема транскодування стиснених МС

У випадку, коли доцільно виконувати транскодування стиснених МС (див. п.1.1.1), кадри відправляються до транскодерів, вмонтованих в обладнання, що знаходиться на стику конвергентних мереж (зокрема у шлюзи, базові станції, контролери багатоабоненських мультимедіа-конференцій).

Першим кроком при транскодуванні стиснених МС є декомпресія блоків даних $L_{i,j}(c_x)$ в результаті якої визначають значення відліків $q_x \in S$ $(D_a(c_x)=q_x)$. При виконанні транскодування стиснених МС відповідно до класичного методу, значення відліків q_x зберігаються у форматі імпульсно-кодової модуляції (ІКМ) [129] з частотою дискретизації 8 КГц та 16-бітним кодуванням [30,88,112,119]. Далі блоки даних $L_{i,j}(q_x)$ можуть змішуватись згідно з функцією M_i^u або стискатись згідно

з функцією K_b. Відповідно до функції K_b , на виході транскодера формуються блоки даних $L_{i,j}(c_x)$.

Основними проблемами при використанні класичного методу є [88,100,107]:

- значні часові затримки, зумовлені структурними особливостями алгоритмів стиснення, а також необхідність подвійного виконання процесів декомпресії $D_a(c_x)$ та стиснення $K_b(p_x)$;
- висока складність оброблення потоків даних в реальному масштабі часу;
- втрата якості мови, через накопичення помилок квантування під час подвійного виконання процесів декомпресії $D_a(c_x)$ та стиснення $K_b(p_x)$.

Таким чином, актуальним є завдання дослідження методів та комп'ютерних засобів транскодування стиснених МС, що дозволять усунути недоліки класичного методу.

1.2. Порівняльний аналіз алгоритмів, що використовуються при транскодуванні стиснених мовних сигналів

1.2.1. Алгоритми стиснення мовних сигналів

Сьогодні існує декілька десятків алгоритмів стиснення МС, які структурно та семантично відрізняються один від одного [50,74,132]. Деякі з цих алгоритмів (Audiocodes [65], MELP [143], CVSELP [72]) розроблені приватними фірмами під конкретні комп'ютерні засоби, інші (ACELP [95,96] ADPCM [91], LD-CELP [92], IMBE [84] ETSI GSM [114]) створені науково-дослідними установами, які тісно пов'язані з міжнародними організаціями по стандартизації.

Завданням будь-якого алгоритму стиснення МС є отримання цифрової послідовності відліків, яка вимагає мінімальної швидкості передачі і з якої декомпресор зможе відновити МС з мінімальними втратами [49].

Будь-який алгоритм стиснення МС визначається набором характеристик, серед яких можна виділити [29]:

- швидкість – діапазон швидкостей передачі кадрів;
- довжина кадру – міра кількості часу, що визначає розмір кадру;

- затримка – час, необхідний для стиснення МС;
- суб’єктивна оцінка якості мови (MOS (mean opinion score)) – міра якості мовлення, яка отримується шляхом опрацювання оцінок, що даються групами слухачів. Оцінки інтерпретуються наступним чином: 4-5 – висока якість мовлення, 3.5-4 – прийнятна якість мовлення, 3-3.5 – задовільна якість мовлення, 2.5-3 – незадовільна якість мовлення (потребує концентрації уваги для розуміння) [6];
- завадостійкість – здатність алгоритму правильно функціонувати при наявності перешкод;
- коефіцієнт стиснення – залежність обсягу вхідного сигналу до обсягу вихідного сигналу;
- складність виконання алгоритму – кількість операцій необхідних для реалізації алгоритму;
- завантаження пам’яті – обсяг завантаження оперативної та постійної пам’яті.

За принципами роботи алгоритми стиснення МС поділено на три класи (рис. 1.2) [29]:

1. Стиснення форми МС.
2. Стиснення параметрів мовного тракту людини.
3. Гібридне стиснення.

Прикладами алгоритмів, які належать до першого класу, є: PCM [129], DPCM [98], ADPCM [71], DM [74], ATC [154] та інші. До другого класу відносять алгоритми: LPC [3], MBE [141], а також смугові, формантні, фонемні, ортогональні та гомоморфні вокодери. До третього класу відносять алгоритми: RELP [74], MPLPC [138], CELP [133] та інші.

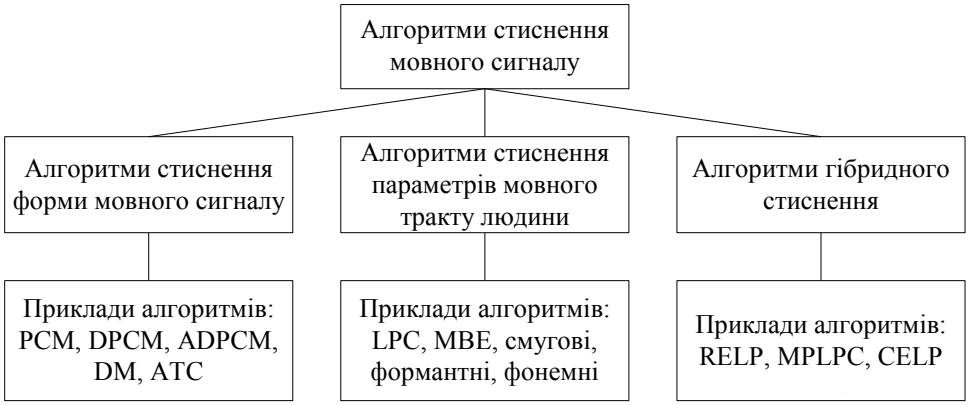


Рис. 1.2. Класифікація алгоритмів стиснення МС

Використання алгоритмів стиснення МС на практиці залежить від особливостей формування та відтворення МС.

Принцип роботи алгоритмів класу стиснення форми МС, полягає у відновлені декомпресором форми МС. Відомо, що сусідні значення відліків слабо відрізняються одне від одного [74], тому з високою точністю можна спрогнозувати значення будь-якого відліку МС на основі значень декількох попередніх відліків [68,74]. При побудові алгоритмів стиснення МС названа закономірність використовується двома способами [74]. По-перше існує можливість зміни параметри квантування в залежності від характеристик МС. У цьому випадку крок квантування вимірюється, що дає змогу кодеру звузити динамічний діапазон МС. Крім того, деякі алгоритми виконують зміну параметрів квантування в рамках мовних складів, а інші змінюють ці параметри на основі аналізу статичних даних про амплітуду сигналу, одержану за відносно короткий інтервал часу.

Найпростішим алгоритмом класу стиснення форми МС є імпульсно-кодова модуляція (ІКМ) [129]. Більш складнішим алгоритмом є диференціальна ІКМ (ДІКМ) [91]. До класифікаційних ознак ДІКМ можна віднести наявність блоку лінійного прогнозування авторегресійних послідовностей (прогнозувальник) та використання багаторівневого (більше двох рівнів) квантувальника. ДІКМ кодує різницю між значенням поточного відліку і оцінкою значень попередніх відліків. Системи з ДІКМ, на відміну від ІКМ, забезпечують на порядок більшу завадостійкість при аналогічній якості МС.

Дельта-модуляція (ДМ) представляє собою частковий випадок ДІКМ з використанням однорозрядного квантувальника [98]. У ДМ приймачу передаються тільки квантовані значення різниці між поточним відліком та його прогнозованим значеням, причому, квантування виконується лише по двох рівнях. Необхідно відмітити, що ДМ забезпечує найкращий коефіцієнт стиснення серед усіх алгоритмів стиснення МС [66]. Кодеки алгоритму ДМ не втрачають працездатності при виникненні одиночних помилок та характеризуються простотою побудови компресора та декомпресора [66,98].

Одним з найпоширеніших алгоритмів класу стиснення форми МС є адаптивно-диференціальна імпульсно-кодова модуляції (АДІКМ) та його різновиди [70,71,91]. Оскільки в аналоговому МС неможливі різкі стрибки інтенсивності, то кодеки АДІКМ стискають не саме значення амплітуди МС, а його зміну в порівнянні з попереднім значенням, що дає змогу зменшити число розрядів для кодування відліку. Для представлення відліку МС в форматі АДІКМ значення зміни рівня сигналу стискається до чотирьохзначного числа, при цьому частота вимірювання амплітуди сигналу зберігається незмінною. Такий підхід забезпечує прийнятну якість мовлення (3,5≤ MOS ≤4) на швидкостях 16-32 Кб/с.

У алгоритмі підсмугового стиснення МС фільтрується на декілька підсмуг. Кожен підсмуговий сигнал адаптивно стискається [76]. Кількість біт для стиснення МС, визначається відповідно до кількості біт квантування призначених критерієм сприйняття. При стиснені кожного підсмугового сигналу шуми квантування обмежуються своєю підсмугою. Найкращі характеристики спостерігаються при збільшенні числа частотних піддіапазонів, а також при динамічній зміні кількості біт на вибірку від одного піддіапазону до іншого [68,74].

В основі принципів побудови алгоритмів класу стиснення параметрів мовного тракту людини закладено класичну модель створення мови, відповідно до якої всі голосові зв’язки являються джерелом збудження, а голосовий тракт – нелінійним фільтром, який формує спектр МС. При цьому параметри джерела та фільтра - нестаціонарні. Кодеки, побудовані згідно з класом стиснення параметрів мовного тракту людини називають вокодерами [28,44].

Робота вокодерів базується на моделі джерела, з якого отримується інформація про параметри МС. Результатом стиснення є коди параметрів джерела МС. Тип вокодера залежить від способів знаходження моделі системи та її параметрів.

При вокодерному стисненні мовний тракт людини представляється нелінійним фільтром із змінними в часі параметрами, що збуджується джерелом білого шуму (при формуванні приголосних звуків) або послідовністю тонових імпульсів (при формуванні голосних звуків) (рис. 1.3).

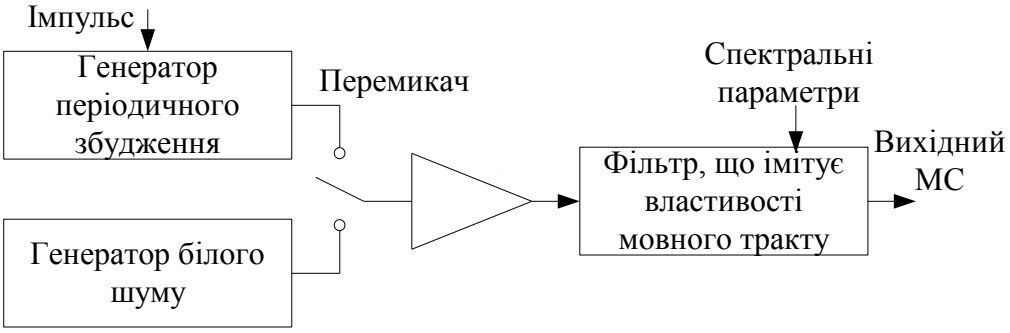


Рис. 1.3. Схема моделі функціонування мовного тракту людини

Компресор вокодерах обробляє наступну інформацію: параметри фільтра, який формує МС; вказівник на голосний/приголосний звук; потужність сигналу збудження; висота тону (ВТ) для голосних звуків [12].

За принципами аналізу та синтезу МС вокодери поділяють на смугові (канальні), формантні, фонемні, ортогональні та гомоморфні.

Одними з перших алгоритмів класу стиснення параметрів мовного тракту людини були канальні вокодери, які використовують слабку чуттєвість слуху людини до незначних фазових зсувів сигналу [141]. Для елементарних відрізків МС за допомогою вузькосмугових фільтрів визначається амплітудний спектр. Сигнали з виходів фільтрів детектуються, пропускаються через фільтри низької частоти, дискретизуються та піддаються двійковому стисненню. Таким чином визначаються: параметри мовного тракту людини, ВТ, збудження МС, вказівник на голосний/приголосний звук. Комп’ютерні засоби реалізації канальних вокодерів

розробляють в цифровій та аналоговій формі. Канальні вокодери забезпечують задовільну якість мовлення на швидкостях порядку 2,4 Кб/с [141].

У формантних вокодерах огинаюча спектру МС визначається комбінацією формант (резонансних частот голосового тракту) [52,141]. Основні параметри формант - центральна частота, амплітуда та ширина.

У ортогональних вокодерах огинаюча спектру розкладається в ряд по вибраній системі ортогональних базисних функцій [76]. Обчислені коефіцієнти цього розкладання передаються декомпресору.

Гомоморфне оброблення сигналів дає змогу представити МС як тимчасову згортку імпульсної перехідної характеристики мовного тракту з сигналом збудження [141]. У частотній області згортка імпульсної перехідної характеристики відповідає добутку частотної характеристики мовного тракту та спектру сигналу збудження.

Вокодери, що базуються на принципах алгоритмів лінійного прогнозування (ЛП) представляють мовний тракт людини лінійним фільтром з безперервно імпульсною перехідною характеристикою, у якому кожне чергове значення відліку МС визначається як лінійна комбінація значень попередніх відліків [141]. У такому вокодері МС ділиться на блоки, для кожного з яких визначаються коефіцієнти фільтру прогнозування. Ці коефіцієнти квантуються та передаються декомпресору. Потім МС пропускається через фільтр, частотна характеристика якого зворотня частотній характеристиці мовного тракту. На виході фільтру отримується помилка прогнозування. У результаті таких перетворень набагато виразніше виявляється довготривала кореляція в сигналі, що забезпечує точніше визначення значення частоти основного тону та дає змогу виділити вказівник на голосний/приголосний звук. Швидкості кодів для вокодерів побудованих на базі алгоритмів ЛП при задовільній якості мовлення, змінюються від 1 Кб/с до 2,4 Кб/с.

Алгоритми гібридного стиснення МС поєднує переваги алгоритмів класу стиснення форми МС та стиснення параметрів мовного тракту людини [28,74,141].

Алгоритми стиснення МС класу гібридного стиснення є найбільш ефективними серед відомих алгоритмів стиснення [12,74]. Вони використовують

відому інформацію про МС для покращення якості мовлення та зменшення швидкості результуючого коду. Зокрема при стисненні МС використовується замаскований слуховий шум, слухова частота розширення, слухова фаза нечутливості, складові зміни енергії, довготермінові звукові властивості тракту, крок квантування [141].

У алгоритмах класу гібридного стиснення, МС дискретизується з метою одержання необхідних параметрів мови. Однак, замість безпосереднього стиснення висоти, модуляції та інших характеристик МС, вони використовуються для синтезу фрагменту МС, з якого вони були отримані. Синтезований фрагмент мови, звичайно, тривалістю від 10 до 30 мс., порівнюється з початковим МС. Якщо вони співпадають з прийнятним допуском, то параметри мови залишаються без змін. Якщо ж реальний та синтезований фрагменти МС відрізняються більш ніж на задану величину, то параметри мови коректуються для досягнення необхідного збігу. Кінцевим етапом процедури зворотного зв'язку є аналіз шляхом синтезу (ABS – analysis by synthesis) - [81]: редагування параметрів мови для забезпечення синтезу форми МС, найбільш наближеної до його початкової форми. Після визначення значень параметрів МС вони співставляються з характеристиками кодової книги. При виявленні збігу, замість значення параметра, використовується його положення в кодовій книзі, що суттєво зменшує обсяг інформації для передачі [54].

За описаними вище принципами працюють алгоритми збудження на основі кодових книг (CELP - Code Excited Linear Predictive) [1,48,54,55]. Кодеки CELP на швидкостях до 4,8 Кб/с забезпечують задовільну якість мовлення. Головним недоліком кодеків типу CELP є висока часова та апаратна складність виконання [12,74].

Для одержання стисненого МС кодеки з регулярним імпульсним збудженням (RPE – Regular Pulse Excited) використовують лінійний прогнозувальник, який покращує показник кореляції між відліками вхідного МС [74]. Якщо прогнозування виконується добре, то на виході прогнозувальника формується практично білий шум з рівномірним спектром. Для декомпресії стисненого МС отриманий білий шум з рівномірним спектром фільтрують фільтром лінійного прогнозування. Проте, через

наявність в МС квазіперіодичних формантних складових, лінійний прогнозувальник не може усунути довготривалої кореляції з ВТ формант, і вони будуть явно відображатись у спектрі помилки прогнозування.

У кодеках з багатоімпульсним збудженням (MPE - Multiband Excitation) в якості сигналу збудження використовується послідовність з чотирьох - шести коротких імпульсів [12,83]. Тимчасове положення кожного імпульсу та їх амплітуди визначаються в процесі виконання процедури аналізу шляхом синтезу, яка виконується до досягнення максимальної кореляції між початковим та синтезованим МС. Параметри імпульсів збудження, що мінімізують помилку прогнозування, підбирають послідовно. Для забезпечення прийнятної якості мови при швидкості коду близько 10 Кб/с достатньо задати положення імпульсів з кроком близько 1 мс., і точністю амплітуд до 5 %. [83].

У таблиці 1.1 наведено результати порівняльного аналізу алгоритмів стиснення МС.

Таблиця 1.1

Порівняльний аналіз алгоритмів стиснення МС

Стандарт / Формат	Алгоритм стиснення МС	Швид-кість, Кб/с	Довжина кадру, мс	Затримка, мс	MOS	Завадо-стійкість	Передача одночасто-тного сигналу	Складність алгоритму, млн.. оп./с	Клас
1	2	3	4	5	6	7	8	9	10
ITU G.711	PCM A-law / PCM u-law	64	0,125	0,125/0,75/5	4,15	10	так	0	1
ITU G.722	SB-ADPCM	64	40	5	4,1	9	так	1	1
		56	35			9			
		48	30			8			
ITU G.721	ADPCM	32		5	4,1	9	так	1	1
ITU G.726	ADPCM	40	25	5	3,91	9	так	1	1
		32	0,125/20	1/5		9			
		24	15	5		8			
		16	10	5		8			
ITU G.728	LD-CELP	16	0,625/10	2,5/3 … 5	3,69	4	так	30	3
ITU G.729	CS-ACELP	8	10	10	3,96		так	20	3
ITU G.729a	CS-ACELP	8	10	10	3,71		так	11	3
ITU G.723.1	MP-MLQ	6,3	30/24	30/37,5	3,93		так	16	3
ITU G.723	ACELP	5,3	30/20	30/37,5	3,66				3
INMAR-SAT-M	IMBE	6,4		80	3,1				3
	IMBE	3,6							
ETSI GSM	RPE-LTP	13	20		3,3			10	2
eXpressDSP	MMBE	2,4	30	45	3,5				3
eXpressDSP	ICELP	4,8	30	60	3,7				3
AudioCodes	NetCoder	6,4	20		3,85				
AudioCodes	NetCoder	7,2	20		3,91				
AudioCodes	NetCoder	8	20		4,1				
DSPSE	Voice-Wave	4,8-12,8					так-ні		
USFS 1016	CELP	4,8					ні	30	3
USFS 1015	LPC10e	2,4					ні	15	2
ETSI TETRA	ACELP	4,8			3,4				3
MELP	MELP	2,4		45	3,5				3

продовження таблиці 1.1.

1	2	3	4	5	6	7	8	9	10
TIA IS-54	VSELP	5,6	20				ні		3
D-AMPS	VSELP	7,95	20		3,3				3
TETRA	ACELP	4,57	30						3
eXpressDSP	RCELP	3,6		30	3,5				3
3GPP AMR-WB	ACELP	6, 6 – 23. 85	20					14	3
AMR	ACELP	4, 75– 12, 2	20					14	3

У першій колонці таблиці 1.1 наведено список найпоширеніших форматів стиснених МС розроблених міжнародними організаціями по стандартизації та приватними фірмами. У другій колонці - список алгоритмів стиснення МС. У наступних колонках наведено характеристики, за якими виконувалось порівняння алгоритмів стиснення МС:

- швидкість – діапазон швидкостей передачі кадрів;
- довжина кадру – міра кількості часу, що визначає розмір кадру;
- затримка – час необхідний для стиснення МС;
- MOS - міра якості мовлення;
- завадостійкість - здатність алгоритму правильно функціонувати при наявності завад;
- передача одночастотного сигналу – здатність кодека передавати одночастотні сигнали;
- складність виконання алгоритму – кількість операцій необхідних для виконання алгоритму;
- клас стиснення МС: 1- стиснення форми МС; 2 – стиснення параметрів мовного тракту людини; 3 – гібридне стиснення.

До характеристик алгоритмів стиснення МС, які не включені в табл.1.1, відносять: завантаження пам’яті, споживання енергії, коефіцієнт стиснення, частота квантування та інші. Однак більшість перелічених характеристик залежать від комп’ютерних засобів реалізації кодеків [29].

Проведений аналіз алгоритмів стиснення МС показує, що вибір алгоритму є достатньо складним завданням, вирішення якого залежить від характеристик алгоритмів стиснення МС та вимог системи, в якій алгоритм буде функціонувати. Однією з найважливіших характеристик алгоритмів стиснення МС є якість мови [74]. До важливих характеристик алгоритмів стиснення віднесемо швидкість передачі, затримку та складність виконання алгоритму [12,74]. Для кодеків з низькими бітовими швидкостями необхідна менша смуга частот, тому вони забезпечуються більш високу ефективність використання спектру та потужності МС.

Аналіз алгоритмів класу стиснення форми МС показує, що вони ефективно стискати будь-які сигнали, а їх функціонування не залежать від типу сигналів. Їх перевагами є: стійкість до широкого діапазону характеристик джерела сигналу, простота реалізації, невисокі швидкості передачі. Недоліком алгоритмів цього класу є низька ефективність при роботі з сигналами, в яких спостерігаються різкі стрибки амплітуди.

Оскільки, принципи побудови та функціонування вокодерів враховують особливості створення та сприйняття мови, то їх рекомендується використовувати для стиснення МС. Недоліком алгоритмів класу стиснення параметрів мовного тракту людини є синтетична якість мови. При цьому навіть суттєве збільшення швидкості передачі практично не покращує якості мови. Вокодери вмонтовують в обладнання спеціалізованих систем зв’язку, де головне не якість мови, а високий коефіцієнт стиснення.

Найбільш перспективними є алгоритми класу гібридного стиснення МС, оскільки вони забезпечують найкращий коефіцієнт стиснення та прийнятну якість мови при низьких швидкостях передачі. Крім того, вони характеризуються високою обчислювальною стійкістю та складністю виконання. Алгоритми гібридного класу стиснених МС рекомендується використовують в спеціалізованих системах зв’язку та в системах зв’язку загального призначення.

1.2.2. Алгоритми мікшування мовних сигналів

Сеанс зв'язку між двома учасниками, що взаємодіють між собою в реальному масштабі часу, починається з одержання учасниками комплексного мовного потоку, який є сумою лінійних комбінацій впливових звукових хвиль активних учасників [31]. Для створення такого потоку використовують алгоритми мікшування МС. Процес мікшування виконується блоком мікшування (БМ, мікшер) та виконує змішування вхідних потоків, що поступають від джерел генерування МС до одного вихідного потоку [31,112,125,124].

Сьогодні існує ряд алгоритмів мікшування МС, які відрізняються архітектурою БМ та особливостями мікшування вхідних потоків [36,82,125,127]. До особливостей мікшування віднесено формат МС та спосіб виконання мікшування МС [31]. Спільним для всіх алгоритмів мікшування МС є те, що вони працюють з декомпресованими МС, та на виході генерують один вихідний потік.

Відповідно до особливостей роботи алгоритмів мікшування МС їх поділено два класи [126]:

1) Одноступінчасте мікшування (monostage mixing). До цього класу відносять алгоритми, які перед початком виконання процесу мікшування очікують всі *i*-ті блоки даних $L_{i,j}(q_x)$.

2) Багатоступінчасте мікшування (multistage mixing). До цього класу відносять алгоритми, які виконують процес мікшування по мірі надходження блоків даних $L_{i,j}(q_x)$.

Нехай у кожен канал g ($g=1,...B,$ де B – кількість каналів блоку мікшування) БМ надходять блоки даних $L_{i,j}(q_x)$ від декількох активних джерел N. Тоді для одержання змікшованого мовного потоку обчислюється вираз [31,82]:

$$M_i^u(q_x)=\sum_{j=1}^{N} L_{i,j}(q_x), \qquad (1.1)$$

де $M_i^u(q_x)$ – функція мікшування.

Крім того, у БМ для кожного активного джерела N з потоку даних $M_i^u(q_x)$ виключаються його власні блоки даних [31,82]:

$$P_j(q_x)= M_i^u(q_x) - L_{i,j}(q_x), \quad (1.2)$$

де P_j – функція виключення власного блоку даних j-го джерела сеансу зв'язку.

Вирази (1.1) та (1.2) є базою для побудови будь-якого алгоритму мікшування МС [18].

Припустимо, що всі активні джерела N генерують i-ті кадри із сталим періодом per ($per=const$). Найменший інтервал, протягом якого два незалежні джерела гарантовано згенерують свої i-ті кадри, рівний $per/2$. Для N активних джерел таким інтервалом буде $per=per-(per/2^{(N-1)})$. Мікшування відліків q_x з двох незалежних блоків даних $L_{1,1}(q_x)$ та $L_{1,2}(q_x)$, що поступають у різні канали БМ, регламентується правилом [127]: значення відліків q_x з блоків даних $L_{1,1}(q_x)$ та $L_{1,2}(q_x)$ можуть мішкуватись, якщо $|l(L_{1,1}(q_x))-l(L_{1,2}(q_x))|<per$, де $l(L_{1,1}(q_x))$ і $l(L_{1,1}(q_x))$ - часи створення блоків даних $L_{1,1}(q_x)$ та $L_{1,2}(q_x)$ джерелами зв'язку.

Алгоритм бінарного мікшування МС. Нехай блоки даних $L_{1,1}(q_x)$ та $L_{1,2}(q_x)$ передаються в БМ з часом τ_1 та τ_2, відповідно. Приймемо, що Δmax і Δmin максимальна і мінімальна затримки передачі блоків даних $L_{i,j}(q_x)$ від джерела до БМ.

Алгоритм бінарного мікшування складається з наступних етапів [127]:

1. Якщо $\Delta max-\Delta min>per$ то БМ не в змозі визначити набір блоків даних для мікшування.
2. Якщо $\Delta max-\Delta min\le per$ то час генерування кадрів із блоками даних $L_{1,1}(q_x)$ і $L_{1,2}(q_x)$ буде визначатись інтервалами $[g^e{}_1(L_{1,1}(q_x)), g^I{}_1(L_{1,1}(q_x))]$ і $[g^e{}_2(L_{1,2}(q_x)), g^I{}_2(L_{1,2}(q_x))]$, відповідно, де $g^e{}_s(L_{i,j}(q_x))=\tau_s-\Delta max$, $g^l{}_s(L_{i,j}(q_x))=\tau_s-\Delta min$, і $s\in[1,2]$. Без загальних втрат одержимо $g^e{}_{min}=min(g^e{}_1(L_{1,1}(q_x)), g^e{}_2(L_{1,2}(q_x)))=g^e{}_2(L_{1,2}(q_x))$ і $g^l{}_{max}=max(g^l{}_1(L_{1,1}(q_x)), g^l{}_2(L_{1,2}(q_x))= g^l{}_1(L_{1,1}(q_x))$.

   а) Якщо $g^l{}_1(L_{1,1}(q_x))- g^e{}_2(L_{1,2}(q_x))\le p$ то відліки із блоків даних $L_{1,1}(q_x)$ і $L_{1,2}(q_x)$ можна мішкувати.

б) Якщо інтервали генерування кадрів із блоками даних $L_{1,1}(q_x)$ і $L_{1,2}(q_x)$ перекриваються і $g^l_1(L_{1,1}(q_x))- g^e_2(L_{1,2}(q_x))>per$, то БМ не в змозі визначити набір відліків із блоків даних для мікшування.

в) Якщо інтервали генерації блоків даних $L_{1,1}(q_x)$ і $L_{1,2}(q_x)$ не перекриваються і існує ціле $k \geq 1$, таке що $(k-1)*p \leq g^l_1(L_{1,1}(q_x))-g^e_2(L_{1,2}(q_x))<k*p$, то:

1) якщо $(k-1)*p \leq g^l_1(L_{1,1}(q_x))-g^e_2(L_{1,2}(q_x))<k*p$ то БМ може бути визначено два набори блоків даних для мікшування $\{L_{1,1}(q_x), L_{1,2}(q_x)+k-1\}$ і $\{L_{1,1}(q_x), L_{1,2}(q_x)+k\}$;
2) якщо $k*p \leq g^l_1(L_{1,1}(q_x))-g^e_2(L_{1,2}(q_x))<(k+1)*p$, то для мікшування формується набір блоків даних $\{L_{1,1}(q_x), L_{1,2}(q_x)+k\}$;
3) якщо $(k+1)*p \leq g^l_1(L_{1,1}(q_x))-g^e_2(L_{1,2}(q_x))<(k+2)*p$, то БМ не в змозі визначити набір блоків даних для мікшування.

Ефективність вище розглянутого алгоритму залежить від величини джитера ($\Delta max-\Delta min$), і періоду per.

Алгоритм множинного мікшування МС. Нехай $[g^e_1,g^e_1]$, $[g^e_2,g^e_2]$,… $[g^e_N,g^e_N]$ інтервали створення пакетів $L_{1,1}(q_x)$, $L_{1,2}(q_x)$,..., $L_{s,N}(q_x)$ джерелами N, $g^e_{\min} = \min\limits_{s\in[1,N]} g^e_s$ і $g^l_{\max} = \max\limits_{s\in[1,N]} g^l_s$. Алгоритм множинного мікшування складається з наступних етапів [14]:

1. БМ отримує блоки даних $\{L_{1,1}(q_x), L_{1,2}(q_x),..., L_{s,N}(q_x)\}$ від N джерел і $g^{\sim l}_{\max} = \max\limits_{s\in[2..N]} g^l_s(L_{s,N}(q_x))$. Припускають, що k_1 найменше ціле при якому $g^e_1(L_{1,1}(q_x))+ k_1*per> g^{\sim l}_{max}$. Виконується заміна блоку даних $L_{1,1}(q_x)$ на $L'_{1,1}(q_x)= L_{1,1}(q_x)+ k_1$. Блок даних $L'_{1,1}(q_x)$ називають фіксованим.
2. $\forall s \in [2..N]$ виконуються наступні дії:

а) відповідно до алгоритму бінарного мікшування обчислюються набори блоків даних для мікшування;

б) нехай набір блоків даних для мікшування буде $\{L'_{1,1}(q_x), L'_{s,N}(q_x)\}$. Блок даних $L'_{s,N}(q_x)$ позначається як вибраний, якщо існує вибір між набором блоків даних для мікшування, в іншому випадку він позначається як фіксований.

3. а) g^l_{max} і g^l_{min} обчислюються наступним чином: $g^l_{\max} = \max\limits_{s\in[1..N]} g^l_s(L'_{s,N}(q_x))$,

$g^e_{\min} = \min\limits_{s\in[1..N]} g^e_s(L'_{s,N}(q_x))$.

б) Якщо g^l_{max}- $g^e_{min} \le per$, то виконання алгоритму закінчується, при цьому формується набір блоків даних $\{L'_{1,1}(q_x), L'_{1,2}(q_x),..., L'_{s,N}(q_x)\}$, які можна мікшувати.

в) Якщо g^l_{max}- $g^e_{min} > per$, то виконуються наступні дії:

1) Нехай $g^e_{min} = g^e_s(L'_{s,N}(q_x))$. Якщо блок даних $L'_{s,N}(q_x)$ помічений як фіксований, то БМ не в змозі визначити набір даних для мікшування.
2) Якщо блок даних $L'_{s,N}(q_x)$ помічений для вибору, то він змінюється блоком даних $(L'_{s,N}(q_x)+1)$ і помічається як фіксований.
3) Якщо g^l_{max}- $g^e_{min} \le per$ то виконання алгоритму закінчується, при цьому формується набір блоків даних $\{L'_{1,1}(q_x), L'_{1,2}(q_x),..., L'_{s,N}(q_x)\}$ для мікшування.
4) В іншому випадку відбувається повернення до кроку 3.в.1.

Алгоритм частково розподіленого мікшування. Алгоритм частково розподіленого мікшування (distributed partial mixing) змішує значень блоків даних кадрів, що передаються згідно з протоколом TCP [125]. Даний алгоритм успішно працює на перевантажених комп'ютерних мережах, у яких спостерігається високий процент втрати кадрів [124,125]. Особливостями алгоритму є його орієнтація на ієрархічну архітектуру багатоабонентських мультимедіа-конференцій [22,126-128] та використання часових інтервалів при визначенні часу мікшування. На рис. 1.4 наведена структура БМ алгоритму частково розподіленого мікшування [125].

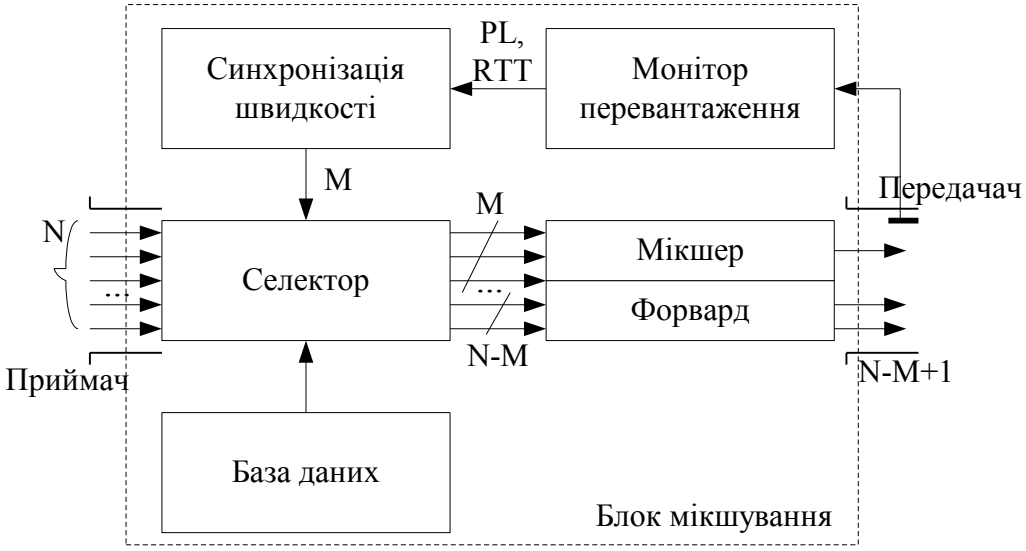


Рис. 1.4. Структура БМ алгоритму частково розподіленого мікшування

Елементами структури БМ є:

- приймач отримує кадри з блоками даних $L_{i,j}(q_x)$;
- селектор визначає кадри, які потрібно мікшувати, або передати без мікшування (форвард);
- мікшер виконує змішування блоків даних $L_{i,j}(q_x)$;
- передавач відправляє кадри функціональним елементам мережі зв'язку, а також отримує повідомлення про завантаження мережі;
- монітор перевантаження оцінює час передачі кадрів (Round Trip Time, RTT) та процент втрати кадрів (Packet Loss, PL);
- синхронізатор швидкості оцінює мережні характеристики та аналізує процес виконання алгоритму частково розподіленого мікшування з метою визначення смуги пропускання, адаптивної до перевантаження та втрат кадрів;
- база даних забезпечує БМ інформацією необхідною для визначення пріоритетів оброблення вхідних кадрів.

Агоритм Гонсалеза-Вахаба. Алгоритм мікшування МС, запропонований Гонсалезом та Вахабом, базується на організації логічних черг в розподіленому буфері БМ [82]. Цей алгоритм виконується на рівні додатків функціональної моделі OSI та відноситься до класу одноступінчатого мікшування.

Алгоритм Гонсалеза-Вахаба залежить від частоти дискретизації відліків q_x інкапсульованих у блоки даних $L_{i,j}(q_x)$. Мікшування відповідно до алгоритму Гонсалеза-Вахаба виконується коли частота дискретизації відліків q_x i-тих кадрів стала [82].

Блоки даних $L_{i,j}(q_x)$ зберігаються у розподіленому буфері мікшування організованого у вигляді FIFO черг. Кількість активних учасників N сеансу зв'язку, визначає кількість логічних черг у розподіленому буфері БМ. Передача блоків даних $L_{i,j}(q_x)$ з розподіленого буфера БМ у мікшер відбувається коли одержані i-ті блоки даних активних учасників сеансу зв'язку. Мікшування виконується БМ відповідно до виразів 1.1 та 1.2.

1.3. Порівняльний аналіз відомих комп'ютерних засобів транскодування стиснених мовних сигналів

Широкому впровадженню транскодерів стиснених МС сприяє розвиток IP-телефонії, комп'ютерної телефонії, мультимедіа конференцій, стільникового зв'язку, дистанційного навчання та інших напрямів галузі зв'язку.

У більшості випадків транскодування стиснених МС виконується у реальному масштабі часу, тобто в темпі поступлення кадрів із блоками даних $L_{i,j}(c_x)$. При цьому частота надходження кадрів та видачі результату залежить від особливостей сеансу зв'язку та характеристик мереж зв'язку. Тому, комп'ютерні засоби транскодування стиснених МС повинні забезпечувати безперервне приймання кадрів та здійснювати оброблення в темпі їх надходження.

Часто транскодери стиснених МС вмонтовують в обладнання, що знаходиться на стику сегментів мережі або на стику конвергентних мереж, зокрема у шлюзи, пристрої управління багатоточковими конференціями, базові станції та інше.

Перелічене обладнання не завжди дозволяє забезпечити конфіденційність, цілісність та автентичність даних, тому широко використовуються протоколи захищеної передачі даних [13,19].

Програмна реалізація транскодерів використовується лише у випадках, коли показник реального часу не є важливим.

1.3.1. Програмні засоби реалізації транскодування стиснених мовних сигналів

Переважна більшість комп'ютерних програмних засобів транскодування стиснених МС реалізовано на об'єктно-орієнтованих мовах програмування та поставляються кінцевому користувачу, як окремі комерційні програмні продукти.

Принцип роботи програмних засобів можна представити наступною таблицею (таблиця 1.2).

Таблиця 1.2

Принцип роботи програмних засобів реалізації транскодування стиснених мовних сигналів

№	Назва операції	Примітка	Виконавець
1	Додавання файлів із МС	Кожна програма має визначений набір кодеків, що обмежує вибір форматів файлів	Користувач
2	Вибір вихідного формату МС	Кожна програма має визначений набір кодеків для транскодування, , що обмежує вибір форматів	Користувач
3.	Встановлення параметрів формату файлу для транскодування	Залежить від особливостей роботи кодека, який використовуються під час транскодування стиснених МС	Користувач
4	Транскодування файлів	Перетворення форматів стиснених мовних сигналів	Програмна реалізація

Як видно з таблиці 1.3 програмні засоби транскодування стиснених МС оперують файлами із відліками МС, що мають визначену структуру. У програмному забезпеченні, транскодування стиснених МС виконуються з допомогою функцій відображення (mapper functions) бібліотеки Audio Compression Manager (ACM) [45].

У бібліотеці ACM описано функції [45]:

- запису та відтворення звуків з використанням стандартної бібліотеки оброблення звуку (mmsystem);
- роботи із системним мікшером звуку;
- роботи із менеджером стиснення аудіо (Audio Compression Manager, ACM).

У ACM існує два види оброблення звукових та мовних сигналів [45]:

- перетворення формату - зміна способу представлення даних, перетворення їх з одного формату в інший без зміни загальних властивостей самого сигналу;
- фільтрування - оброблення звуку в потоці (посилення/послаблення, зміна амплітудно-частотних характеристик, накладення звукових ефектів та інше) без зміни формату.

одулі драйверів ACM, що реалізовують функції перетворення форматів, називають перетворювачами формату (format convertors). Модулі, що реалізовують функції фільтрування, називаються фільтрами (filters). Один і той же драйвер, що підключається, може суміщати в собі різні функції, а також містити декілька перетворювачів або фільтрів.

Спільним для всіх комерційних програмних засобів транскодування стиснених МС є:

- необхідність втручання користувача у процес перетворення стиснених МС;
- робота на універсальних процесорах.

У таблиці 1.3 наведено результати порівняльного аналізу характеристик обладнання для виконання процесу транскодування стиснених МС.

Час транскодування у комп’ютерних програмних засобах залежить від розміру файлу з МС, вхідного і вихідного форматів та характеристик обладнання для виконання процесу транскодування стиснених МС.

Комп’ютерні програмні засоби використовуються лише у системах, де показник реального часу не є важливим.

Таблиця 1.3

Порівняльний аналіз комп'ютерних програмних засобів транскодування

Назва продукту	Виробник	Операційна система	Ціна, $	Частота дискретизації, КГц	Кількість каналів	Швидкість, Кб/с	Підтримка форматів (A->B)*
All To MP3 Converter 1.6	Litex Media, Inc	Windows All	19.95	32 000, 44100, 48000	1-4	32-320	WMA, MP3, OGG, APE, FLAC, WAV -> MP3
Switch Sound Format Converter 1.02	NCH Swift Sound Software	Windows All, MAC OS, Linux	-	6000 - 196000	1,2	8-320	WAV, MP3, OGG, WMA, AIF, AIFF, AU, RA, RAM, RM, RMJ, REC, FLAC, GSM, AAC, M4A, VOX, RAW, DCT -> WAV, MP3, AIFF, AU, GSM, VOX, FLAC, RAW, OGG, AAC, M4A
Xilisoft Audio Converter 2.1	Xilisoft Inc	Windows All	23	8000-48000	1-4	8-320	MP3, WAV, WMA, AAC, FALC, OGG, APE, MP4, M4A, MP2, VQF
AudioConvert 3.1	R. M. de Bour Software	Windows All	49,95	8000- 48000	1-4	32-320	WAV, MP3, MP2, MPEG, M4A, MP4, AAC, G7231, AC3, VOX, AMR, AIF, AIFF, AU -> WAV, AC3, MP2, MP3, MP4, OGG, G721, G723, G726, G729, MPC, VOX, WMA, AMR, RAW, AIFC
Advanced WMA Workshop 2.1	LitexMedia, Inc.	Windows All	24,95	8000- 48000	1-4	0-320	WMA9, MP3, OGG, WAV -> WMA9, MP3, OGG, WAV
MP3 RM Converter	Audio Tools Factory	Windows All	19,95	8000-48000	1-4	8-320	MP3, RM, WMA, OGG, WAV -> MP3, RM, WMA, OGG, WAV
MP3 Encoder	LineSoft	Windows All	-	8000-48000	1-4	8-320	WAV, PCM, ADPCM, GSM 6.10, G721, G723, DPCM, WMA, AIFF, AIFC, AIF, SND, AU, RAW, PAF, VOC, W64, MAT4, MAT5, PVF, HTK ->MP3
Recorder v. 5.1	Microsoft	Windows All	-	8000-48000	1-6	8-320	AC3-ACM, ACELP, G.711, G.723.1, GSM 6.10, ADPCM, IMC, PCM, WMA, OGG, MP3 -> AC3-ACM, ACELP, G.711, G.723.1, GSM 6.10, ADPCM, IMC, PCM, WMA, OGG, MP3

Примітка: * (A->B) – транскодування з A до B.

1.3.2. Апаратні засоби транскодування стиснених мовних сигналів

У сучасних мультисервісних мережах транскодери стиснених МС вмонтовують у шлюзи, контролери багатоточкових конференцій, базові станції та інше комунікаційне обладнання.

Шлюз забезпечує взаємодію між VoIP-мережами та мережами інших типів, зокрема, загальною телефонною мережею, а також забезпечують фізичне підключення для локальних аналогових телефонів, факсів і приватних телефонних станцій. Шлюзи різних виробників відрізняються способом підключення до телефонної мережі, апаратною платформою, реалізованими кодеками, інтерфейсом та іншими характеристикам [7]. У таблиці 1.4 наведено результати аналізу характеристик шлюзів IP-телефонії корпоративного класу.

Контролер багатоточкових конференцій забезпечує передачу даних між територіально розподіленими учасниками мультимедійного сеансу в реальному масштабі часу. Цей пристрій складається з багатоточкового контролера і, можливо, багатоточкового процесора. Контролер призначений для узгодження параметрів опрацювання аудіо та відео потоків даних між терміналами. Процесор виконує комутацію, мікшування та оброблення потоків даних. У таблиці 1.5 наведено результати аналізу характеристик сучасних контролерів багатоточкових конференцій.

Необхідно відмітити, що проаналізовані комп’ютерні засоби (табл. 1.4, табл. 1.5) працюють згідно з класичним методом транскодування стиснених МС. У переважної більшості проаналізованих комп’ютерних засобів відсутні механізми забезпечення конфіденційності, цілісності та автентичності мовної інформації.

Аналіз таблиць показує, що багатоканальні комп’ютерні засоби із вмонтованими транскодерами стиснених МС мають від 4 до 480 VoIP-каналів, яких при реалізації багатоточкових сеансів зв’язку не завжди достатньо. В той же час сучасні тенденції розвитку систем зв’язку вимагають від мережного обладнання постійного збільшення кількості каналів та частоти надходження даних, а також забезпечення захисту даних під час їх передачі.

Таблиця 1.4

Порівняльний аналіз характеристик шлюзів IP-телефонії корпоративного класу

Виробник	AddPac		AudioCodes		BOSCom		Cisco		MultiTech		Patton		Quintum		Samsung	
Модель	AP1100B	AP2620	MP-108 FXS	Mediant 2000	BV1008	BV2001	Cisco 827-4V	Cisco AS5350	MVP810	MVP60	SN4528	SN2300	AXG 800	Tenor DX4120	SMG-400	SMG-3200
Кількість VoIP- каналів	8	до 60	8	30-480	8	30	4	30-240	8	60	8	4-60	8	120	4	30
Кількість аналогових портів	8FXS	4-8	8 FXS	-	8 FXS		4 FXS	-	8	-	8 FXS	16	8 FXS	-	4 FXS	16
Кількість цифрових потоків E 1	-	1-2	-	1-16	-	1	-	1-8	-	2		3	-	4	-	
E&M	-	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ISDN PRI	-	+	-	-	-	+	-	+	-	-	-	+	-	+	-	+
ISDN BRI	-	-	-	-	-	-	-	-	-	-	-	+	-	-	-	-
Версія H .323	v.3	v.3	v.4	v.4	v.4	v.4	v.4	v.4	v.4	v.4	v.3	v.3	v.3	v.3	v.3	v.3
SIP	+	+	+	+	+	+	+	+	+	+	+	+	+	+	-	-
MGCP	+	+	+	+	-	-	+	+	-	-	+	+	-	-	-	-
Кодеки (G.711/G.723/ G.726/G.729/G.729a/b)		+/+/+/+/+	+/+/+/+/+	+/+/+/+/+	+/+/+/-/+	+/+/+/-/+		+/+/+/+/+	+/+/+/-/+		+/+/+/-/+		+/+/+/-/+	+/+/+/-/+	+/+/-/+/+	+/+/-/+/+
T.38 FAX	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
Fast Start/Slow Start	+/+	+/+	+/+	+/+	+/+	+/+	+/+	+/+	+/+	+/+	+/+	+/+	+/+	+/+	+/+	+/+
Придушення exo	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
Генерація комфортного шуму	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
Буфер джитера (адаптивний / динамічний)	+/+	+/+	+/+	+/+	+/+	+/+	+/+	+/+	+/+	+/+	+/-	+/-	+/+	+/+	+/-	*/-
Вмонтований gatekeeper	+	+	-	-	-	-	-	+	-	-	-	-	-	+	+	+
H.225	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
RAS	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
H.245	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
Функції IVR	-	+	+	+	+	+	+	+	-	-	+	+	+	+		
Інтерфейс WAN (Ethernet/V.35)	+/-	+/+	+/-	+/-	+/-	+/-	+/-	+/+	+/-	+/-	+/-	+/+	+/-	+/-	+/-	+/-
Маршрутизатор	+	+	-	-	-	-	+	+	-	-	+	+	-	-	-	-
Міжмережевий екран (NAT/PAT)	+/+	+/+	-/-	-/-	+/+	+/+	+/+	+/+	+/+	+/+	+/+	+/+	-/-	-/-	-/-	-/-
Якість обслуговування (IP ToS / Diff Serv)	+/+	+/+	+/+	+/+	+/+	+/+	+/+	+/+	+/+	+/+	+/+	+/+	+/+	+/+	+/+	+/+
Інтерфейс керування (Web / telnet / SNMP)	+/+/+	+/+/+	+/-/+	+/-/+	+/-/+	+/-/+	+/+/+	+/+/+	+/+/+	+/+/+	+/+/+	+/+/+	+/+/+	+/+/+	+/+/+	+/+/+
Статистика викликів (CDR/Radius)		+/+	+/+	+/+	+/+	+1+	+/+	+/+		+/-	+/-	+/-	+/+	+/+	+/-	+/-
Підтримка транскодування	-		+	+	+	+	-	+	+	+	+	+	+	+	+	+

Таблиця 1.5

Порівняльний аналіз характеристик сучасних контролерів багатоточкових конференцій

Виробник	Codian	Emblaze	Polycom	Radvision	Tandberg	Cisco	Accord	Aethra
Модель	MCU 4200	VCON VCB5	MGC-50	SCOPIA-100-12/24	MPS-200	IP/VC 3510	MGC-100	MCU-24
Архітектура	фіксована	фіксована	модульна	фіксована	модульна	фіксована	модульна	модульна
Кількість VoIP- каналів	40	48	128	72	64	15	84	45
Підтримка мереж (IP/ISDN/T1)	+/-/-	+/-/-	+/+/+	+/+/-	+/+/+	+/-/-	+/+/+	+/+/-
Версія Н .323	v.4	v.3	v.3	v.3	v.2	v.3	v.3	v.4
SIP	+	+	-	+	-	-	-	+
MGCP	-	-	-	-	-	-	-	-
Кодеки (G.711/G.722/G.723/G.723.1/G.726/G.729/G.729a/b)	+/+/-/-/-/+/-	-/+/-/+/-/-/-	+/+/+/+/-/-/-	+/+/+/+/+/+/+	+/+/-/-/-/-/-	+/-/-/-/-/-/-	+/+/+/-/-/-/-	+/+/+/+/+/+/-
Інші кодеки	G.728	G.728, AAC	Siren 7, Siren 14	-	G.728, MPEG4	-	G.728	G.728
Відеокодеки (H.261/H.263/H.263+/H.263++/H.264)	+/+/+/+/+	+/+/+/+/+	+/+/-/-/+	+/+/-/-/+	+/-/+/-/+	+/+/-/-/-	+/+/-/-/-	+/+/-/+/+
Мережеві протоколи (HTML/HTTP/FTP/STMP/RTP/RTCP)	-/+/+/-/+/+	+/+/+/+/+/+			-/+/+/+/-/-			
Алгоритми захисту (AES/DES/3DES)	+/-/-	+/+/+	+/-/-	+/+/-	+/+/-	-/-/-	-/-/-	+/+/-
Вмонтований gatekeeper	-	+	-	+	-	+	-	+
Якість обслуговування (IP ToS / Diff Serv)	-/-	-/+	+/+	+/+	+/+	+/+	-/-	-/-
Інтерфейс керування (Web / telnet / SNMP)	+/-/+	+/-/-	+/+/-	+/-/+	+/+/+	+/-/-	+/+/+	+/-/+
Підтримка транскодування	+	+	+	+	+	-	+	+

Часто існуючі комп'ютерні засоби транскодування стиснених МС не можуть задовольнити постійно зростаючі вимоги до швидкості оброблення даних, зокрема, в частині багатоканального оброблення, - тому виникає завдання розробки нових методів та комп'ютерних засобів транскодування, здатних підвищити продуктивність роботи транскодерів.

1.4. Постановка завдання дослідження

Порівняльний аналіз алгоритмів, що використовуються при транскодуванні стиснених МС, проведений в параграфі 1.2 показав, що сьогодні розроблено велику кількість алгоритмів стиснення МС, що формують десятки форматів стиснених МС. У зв'язку з цим, та, враховуючи стрімке збільшення користувачів і поступову конвергенцію мереж зв'язку, постає завдання узгодження форматів МС, що передаються між територіально віддаленими гетерогенними системами.

Найефективнішим засобом для вирішення цього завдання є використання транскодерів, що перетворюють стиснені МС з одного формату в інший [75,100,107,119-121]. Проведений у параграфі 1.3 порівняльний аналіз комп'ютерних засобів транскодування стиснених МС показав, що сьогодні розроблено достатньо багато програмних та апаратних засобів для вирішення цього завдання. Програмні засоби транскодування стиснених МС, на відміну від апаратних, використовуються лише у комп'ютерних системах, де показник реального часу не важливий. Більшість проаналізованих комп'ютерних засобів транскодування стиснених МС працюють відповідно до класичного методу, який характеризується високими часовими затримками, високою складністю оброблення потоків даних та втратою якості мови [4,40,101].

Відмітимо, що місце розміщення транскодерів залежить від архітектури, топології, пропускної спроможності та способу адресації у

мережах зв’язку [22]. Однак, у більшості випадків транскодери вмонтовують в обладнання, що знаходиться на стику сегментів мереж.

Аналіз літератури [2,30,88,86,112,119] показав, що у теорії та практиці перетворення форматів та криптографічного захисту стиснених МС залишилось чимало не вирішених проблем. Зокрема, відсутня класифікація існуючих алгоритмів стиснення та мікшування мовних сигналів, яка враховувала б особливості їх побудови та можливості використання у процесі перетворення з одного формату в інший. Існуючі багатоканальні комп’ютерні засоби перетворення форматів стиснених мовних сигналів базуються на класичному методі – тандем кодера-декодера, що призводить до значних часових затримок, високої складності оброблення потоків даних в реальному масштабі часу та втрати якості мови. Відомі структури багатоканальних комп’ютерних засобів перетворення форматів стиснених мовних сигналів не враховують особливості роботи алгоритмів стиснення та мікшування. При передачі стиснених мовних сигналів незахищеними комп’ютерними мережами не враховуються загрози щодо конфіденційності, цілісності та автентичності даних.

Таким чином, актуальним є наукове завдання розробки багатоканальних комп’ютерних засобів перетворення та криптографічного захисту форматів стиснених мовних сигналів, що враховують основні закономірності і особливості функціонування комп’ютерних систем реального часу.

Вирішення цього завдання дозволить підвищити ефективність та захищеність передачі МС каналами зв’язку, а також зменшити результуючі затримки між територіально віддаленими джерелами багатоканальних комп’ютерних систем реального часу.

ВИСНОВКИ

1. Проведено аналіз особливостей побудови багатоканальних засобів перетворення форматів стиснених МС. Висвітлено принципи функціонування комп’ютерних засобів стиснення МС у незахищених багатоканальних системах реального часу. Показано переваги та недоліки класичного методу транскодування стиснених МС та визначено умови при яких доцільно виконувати транскодування та криптографічний захист стиснених МС.
2. Проведено порівняльний аналіз алгоритмів стиснення та мікшування МС, який дозволив їх класифікувати, виділити переваги і недоліки. Обґрунтовано, що найперспективнішими є алгоритми класу гібридного стиснення МС та багатоступінчастого множинного мікшування МС.
3. Проведено порівняльний аналіз комп’ютерних засобів транскодування стиснених МС. Доведено, що для комп’ютерних систем реального часу найбільш перспективними є апаратно реалізовані транскодери стиснених МС.
4. Обґрунтовано актуальність завдання дослідження багатоканальних комп’ютерних засобів перетворення та криптографічного захисту форматів стиснених МС. Показано, що розв’язання цього завдання дозволить підвищити ефективність та захищеність передачі МС каналами зв’язку, зменшити результуючі затримки між територіально віддаленими джерелами та підвищити ефективність функціонування комп’ютерних систем зв’язку.

РОЗДІЛ 2
МЕТОДИ ПЕРЕТВОРЕННЯ МОВНИХ СИГНАЛІВ

2.1. Формування мовних сигналів відповідно до моделі збудження на основі кодових книг

У першому розділі доведено, що найбільш перспективними алгоритмами стиснення МС є алгоритми класу гібридного стиснення, більшість з яких для формування МС використовує модель збудження на основі кодових книг (CELP) [92,95,96,114].

Модель CELP представляє відліки q_x лінійної авторегресійною моделлю [74,77]:

$$q_x = \sum_{i=1}^{s} a_i q_{x-i} + e_x \ , \tag{2.1}$$

де a_i - вагові коефіцієнти, які називають коефіцієнтами лінійного прогнозування (КЛП), e_x – білий шум.

Для кожного кадру із блоком даних $L_{i,j}(q_x)$, КЛП a_i підбираються таким чином, щоб згенерований спектр МС найбільше наближався до спектру вхідного МС. Обчислення КЛП виконується відповідно до алгоритму Левінсона-Дарбіна [28,48,74].

Кодер CELP працює наступним чином [74]:

1. Вхідний МС сегментується на кадри, які діляться на підкадри.
2. Для кожного кадру відповідно до процедури короткотермінового аналізу з використанням ЛП обчислюються КЛП. Аналогічна процедура виконується для кожного підкадру, однак замість короткотермінового аналізу виконується процедура довготермінового аналізу. Це дає змогу визначити коефіцієнти перцептуально вагового фільтру, коефіцієнти формантно-синтезуючого фільтру та параметри довготермінового лінійного прогнозування.

3. Для кожного підкадру визначаються вектори збудження, довжина яких рівна довжині підкадру. Процедура пошуку векторів збудження починається з генерування множини послідовностей фільтрованих векторів збудження з відповідними коефіцієнтами підсилення. Для кожної послідовності обчислюється середньоквадратична помилка. Далі вибирається кодовий вектор і коефіцієнт підсилення, які забезпечують найменшу середньоквадратичну помилку.
4. Стиснення значень індексу збудження з кодової книги коефіцієнтів підсилення, параметрів довготермінового прогнозування і КЛП. Передача стиснених параметрів у модуль формування кадрів.

На рис. 2.1 наведено узагальнену структурну схему кодера CELP [74,77].

Рис. 2.1. Узагальнена структурна схема кодера CELP

На рис. 2.2 наведено узагальнену структурну схему декодера CELP [74,77].

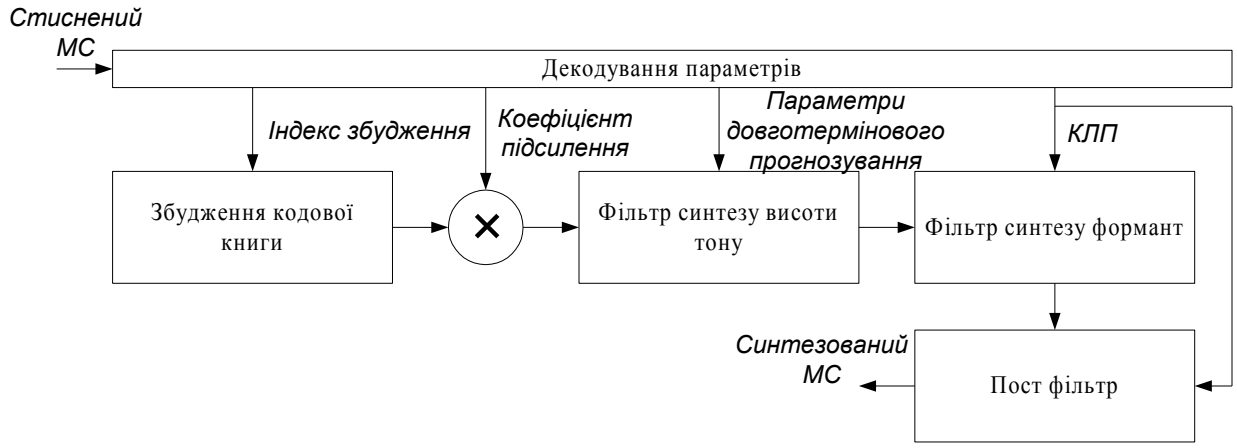


Рис.2.2. Узагальнена структурна схема декодера CELP

Декодер наведений на рис. 2.2, працює наступним чином [74,77]:

1. Декодуються параметри кожного підкадру кадру стиснених МС: індекс збудження, коефіцієнт підсилення, параметри довготермінового прогнозування і КЛП.
2. На основі індексу збудження визначаються вектори збудження. За допомогою фільтру синтезу висоти тону (ВТ) та фільтру синтезу формант синтезується МС.
3. Пост фільтр покращує якість синтезованого МС.

Кодеки CELP забезпечують швидкості коду від 4,8 Кбіт/с до 16 Кбіт/с. При цьому кодеки, що працюють на швидкостях 16 Кбіт/с забезпечують якість мови аналогічну ІКМ.

2.2. Методи транскодування стиснених мовних сигналів

У першому розділі показано, що виконання процесу транскодування стиснених МС пов’язано з високими часовими затримками, високою складністю оброблення потоків даних в реальному масштабі часу та втратою якості мови. Таким чином, актуальним завданням є розробка методів

перетворення форматів стиснених МС, які дозволяють покращити показники класичного методу.

У даному розділі запропоновано два методи транскодування стиснених МС. Вибір кодеків для побудови методів зумовлений їх широкою розповсюдженістю у системах зв'язку реального часу [83].

Запропоновані методи [59,136,137] базуються на перетворенні параметрів кадрів стиснених МС.

У таблиці 2.1 наведено характеристики кадрів алгоритмів класу гібридного стиснення МС, які використані у запропонованих методах транскодування стиснених МС [79,95,96,109].

Таблиця 2.1

Характеристики кадрів алгоритмів класу гібридного стиснення МС

Назва алгоритму	Назва формату	Довжина кадру, мс	Кількість відліків в кадрі	Швидкість, Кб/с	Кількість підкадрів	Час аналізу КЛП, мс.
MP-MLQ	G.723.1	30	240	16	4	7,5
CS-ACELP	G.729A	10	80	16	2	5
VSELP	GSM 06.20	20	160	13	4	

У таблиці 2.2 наведено структуру розміщення бітів у кадрі формату G.729A [96].

Таблиця 2.2

Розміщення бітів у кадрах формату G.729A

Параметр	Кодове слово	Підкадр 1	Підкадр 2	Сума у кадрі
1	2	3	4	5
Лінійні спектральні пари	*L*0, *L*1, *L*2, *L*3	-	-	18
Затримка адаптивної кодової книги	*P*1, *P*2	8	5	13
Парність тонової затримки	*P*0	1	-	1
Індекс фіксованої кодової книги	*C*1, *C*2	13	13	26
Знак фіксованої кодової книги	*S*1, *S*2	4	4	8
Коефіцієнти підсилення кодової книги (крок 1)	*GA*1, *GA*2	3	3	6

продовження табл.. 2.2

1	*2*	3	4	5
Коефіцієнти підсилення кодової книги (крок 2)	*GB*1, *GB*2	4	4	8
Всього				80

У таблицях 2.3 та 2.4 наведено структуру розміщення бітів у кадрі формату G.723.1 зі швидкістю 6.3 Кб/с та 5.3 Кб/с, відповідно [95].

Таблиця 2.3

Розміщення бітів у кадрах формату G.723.1 зі швидкістю 6.3 Кб/с

Параметр	Підкадр 0	Підкадр 1	Підкадр 2	Підкадр 3	Сума у кадрі, біт
Індекси КЛП					24
Затримки адаптивної кодової книги	7	2	7	2	18
Всі комбіновані коефіцієнти підсилення	12	12	12	12	48
Позиції імпульсів	20	18	20	18	73
Знаки імпульсів	6	5	6	5	22
Індекс сітки	1	1	1	1	4
Всього					189

Таблиця 2.4

Розміщення бітів у кадрах формату G.723.1 зі швидкістю 5.3 Кб/с

Параметр	Підкадр 0	Підкадр 1	Підкадр 2	Підкадр 3	Сума у кадрі, біт
Індекси КЛП					24
Затримки адаптивної кодової книги	7	2	7	2	18
Всі комбіновані коефіцієнти підсилення	12	12	12	12	48
Позиції імпульсів	12	12	12	12	48
Знаки імпульсів	4	4	4	4	16
Індекс сітки	1	1	1	1	4
Всього					158

У таблиці 2.5 наведено структуру розміщення бітів у кадрах формату GSM 06.20 [79,109].

Таблиця 2.5

Розміщення бітів у кадрі формату GSM 06.20

Параметр	Кодове слово	Сума у кадрі, біт
Енергія кадра	*R0*	5
Коефіцієнт відбиття	*LPC1-LPC10*	38
Затримка висоти тону	*LAG1-LAG4*	28
Індекси кодової книги	*C*ODE1_1-CODE1_4, CODE2_1-CODE2_4	56
Індекси коефіцієнтів підсилення	*GSP0_1-GSP04*	32
Всього		159

2.2.1. Транскодування між G.723.1 та G.729A

Запропонований метод транскодування між G.723.1 та G.729A базується на перетворенні ряду параметрів кадрів між форматами.

Розроблений метод транскодування між G.723.1 та G.729A виконує перетворення кадрів з формату G.723.1 до G.729A і навпаки та складається з чотирьох етапів: перетворення ЛСП, перетворення ВТ, пошук в АКК, пошук в ФКК.

Подібність структур модулів збудження алгоритмів багатоімпульсного квантування з максимальною достовірністю (БКМД) та лінійного прогнозування, що генерується алгебраїчним кодом спряженої структури (ЛПАКСС), дають змогу виконувати пряме перетворення значення затримки ВТ і індексу ФКК з одного формату в інший [45].

Відомо, що вектор ЛСП формату G.723.1 обчислюється з КЛП [3,140]. Процес виконання першого етапу транскодування з G.723.1 до G.729A здійснює перетворення вектору ЛСП формату G.723.1 у вектор ЛСП формату G.729A. Оскільки, довжина кадру формату G.723.1 втричі більша від довжини кадру формату G.729A, то в процесі перетворення вектору ЛСП використано метод лінійної інтерполяції (рисунок 2.3) [64].

Після виконання цього процесу для кожного підкадру формату G.729A будується перцептуально ваговий синтезуючий фільтр, а вектори ЛПС квантуються та перетворюються до КЛП.

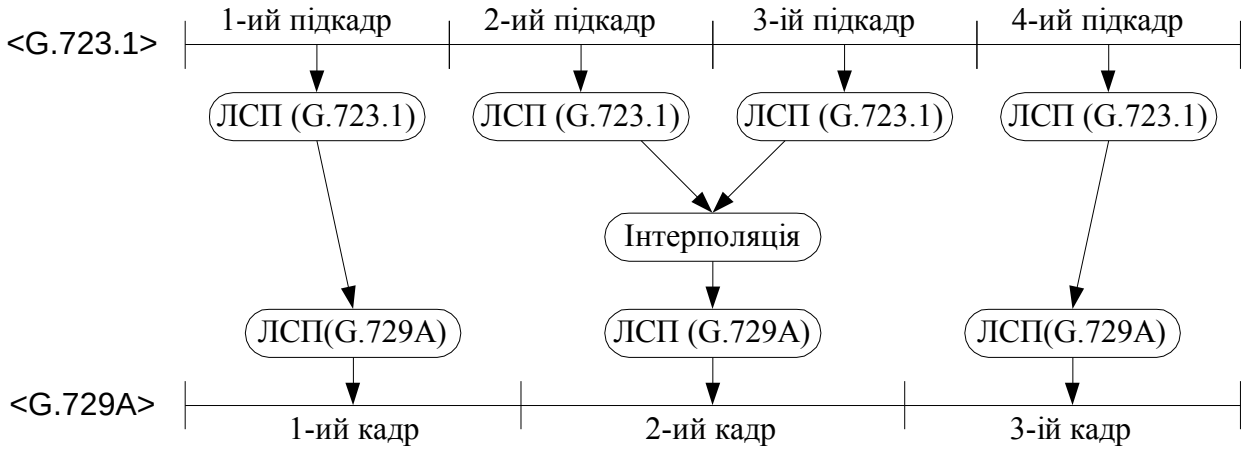


Рис. 2.3. Структурна схема перетворення векторів ЛСП

На рисунку 2.3 показано, як вектори, що містяться у другому та третьому підкадрах кадру формату G.723.1 перетворюються у ЛСП кадрів формату G.729A.

Для визначення ВТ у циклі без зворотного зв'язку формату G.729A використано метод згладжування, у якому враховується подібність та неперервність параметрів ВТ. Для цього значення ВТ у циклі із зворотним зв'язком формату G.723.1 порівнюється із значенням ВТ другого підкадру попереднього кадру формату G.729A. Якщо відстань між двома значеннями ВТ менша 10 відліків, то вони неперервні, і значення ВТ одного формату можна представити значенням ВТ іншого формату. У іншому випадку, якщо різниця між значеннями ВТ форматів G.723.1 та G.729A є більшою за 10 відліків, їх локальні затримки максимізуються, а пошук продовжується в діапазоні ±3 відліків навколо значення затримки ВТ у циклі з зворотним зв'язком для двох форматів згідно з виразом:

$$\begin{cases} R(k_1) = \sum_{x=0}^{79} sw(c_x) - sw(c_x - k_1),\ p_1 - 3 \le k_1 \le p_1 + 3 \\ R(k_2) = \sum_{x=0}^{239} sw(c_x) - sw(c_x - k_2),\ p_2 - 3 \le k_2 \le p_2 + 3 \end{cases}, \qquad (2.2)$$

де $R(k_1)$, $R(k_2)$ – максимізовані локальні затримки алгоритмів ЛПАКСС та БКМД; $sw(c_x)$ – зважений МС, визначений згідно з стандартами G.729A та G.723.1;

k_1,k_2 – шукані значення ВТ у циклі без зворотного зв'язку алгоритмів ЛПАКСС та БКМД;

p_1,p_2 – часова затримка ВТ у циклі із зворотним зв'язком алгоритму ЛПАКСС та БКМД.

Після визначення локальних затримок для форматів G.723.1 та G.729A, значення $R(k_1)$ та $R(k_2)$ нормалізуються через енергію локальних максимальних затримок $R`(t_1)$ та $R`(t_2)$:

$$\begin{cases} R`(t_1) = \dfrac{R(t_1)}{\sqrt{\sum_{x=0}^{79} sw^2(c_x - t_1)}} \\ R`(t_2) = \dfrac{R(t_2)}{\sqrt{\sum_{x=0}^{239} sw^2(c_x - t_2)}} \end{cases}, \tag{2.3}$$

де t_1, t_2 – час локальних затримок алгоритмів ЛПАКСС та БКМД.

Якщо в процесі порівняння локальний максимум алгоритму ЛПАКС більший, ніж 3/4 часу БКМД, то затримка ВТ у циклі без зворотного зв'язку алгоритму ЛПАКСС буде рівна локальній максимальній затримці алгоритму БКМД. В іншому випадку, значення затримки ВТ T_{op} визначається відповідно до виразу:

$$T_{opt} = \begin{cases} t_1, \text{ якщо } R'(t_2) \le 0{,}75 \cdot R(t_1) \\ t_2, \text{ якщо } R'(t_2) > 0{,}75 \cdot R(t_1) \end{cases}, \tag{2.4}$$

На рис.2.4 проілюстровано процес визначення ВТ з використанням методу лінійного згладжування для формату G.729A.

Третій та четвертий етапи запропонованого методу – пошук в АКК та ФКК. Процедури пошуку повністю ідентичні аналогічним процедурам алгоритму ЛПАКСС. При цьому шуканими параметрами АКК є затримка ВТ - Z_{pot} та коефіцієнт підсилення ВТ - G_{pot}, а шуканими параметрами ФКК є індекс кодової книги - I_{cb} та коефіцієнт підсилення кодової книги - G_{cb}.

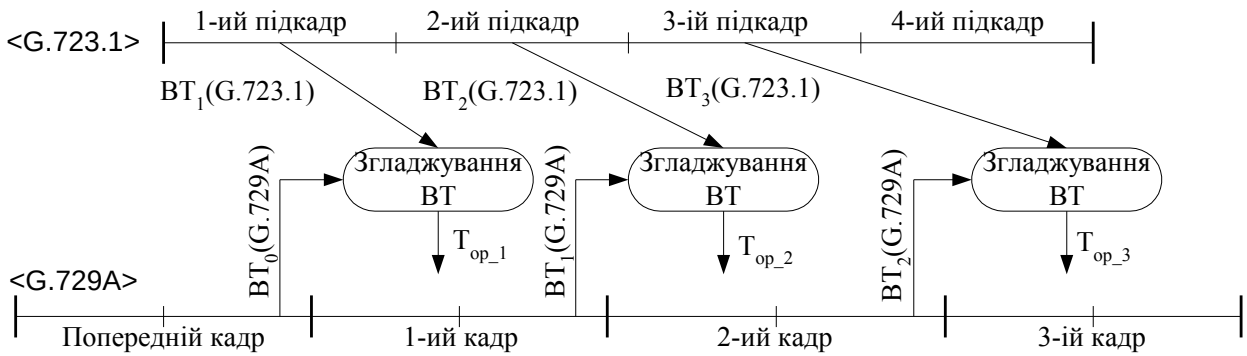


Рис. 2.4. Схема визначення ВТ з використанням функції лінійного згладжування для формату G.729A

При транскодуванні з G.723.1 до G.729A виконуються однотипні етапи. Для перетворення вектора ЛСП формату G.729A до вектора ЛСП формату G.723.1 використано вектори ЛСП з трьох кадрів формату G.729A. Метод лінійної інтерполяції використовується тільки для знаходження вектора ЛСП четвертого підкадру G.723.1 (рис. 2.5).

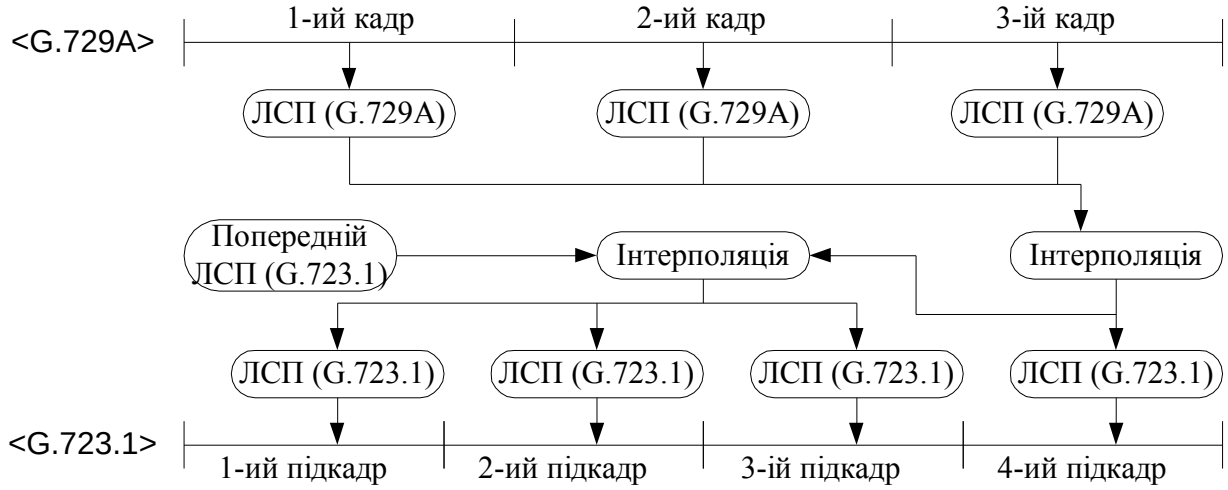


Рис. 2.5. Структурна схема перетворення векторів ЛСП з формату G.729A до G.723.1

Для визначення ВТ у циклі без зворотного зв’язку для кадру формату G.723.1 з цільового МС обчислюється перцептуально зважений МС. Також використовується функція зладжування ВТ, а також обчислюється нормалізована крос-кореляція ВТ. Схема визначення ВТ для формату G.723.1 наведена на рис. 2.6.

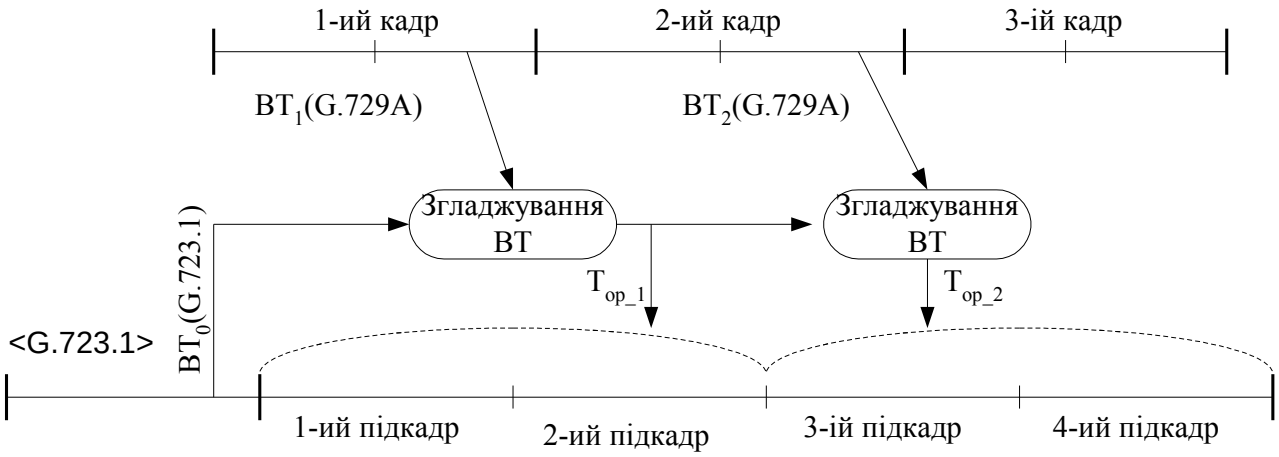


Рис. 2.6. Схема визначення BT з використанням методу лінійного згладжування

Процедури пошуку параметрів в АКК та ФКК ідентичні аналогічним процедурам алгоритму БКМД [95].

Використання запропонованого методу дає можливість виконувати пряме перетворення параметрів кадрів одного формату у параметри кадру іншого, що дозволяє:

- зменшити часову затримку;
- зменшити апаратну складність декодера;
- покращити якість мовлення.

2.2.2. Транскодування між GSM 06.20 та G.729A

Найпоширенішим в галузі Інтернет телекомунікацій, до якої відносять IP-телефонію, комп'ютерну телефонію, мультимедіа-конференції є формат G.729A, який працює згідно з алгоритмом ЛПАКСС [96]. У галузі бездротових комунікацій, до якої відносять стільникову телефонію та супутниковий зв'язок часто використовується формат GSM 06.20, який працює згідно з алгоритмом лінійного прогнозування що генерується векторною сумою (ЛППГВС) [79].

Кодеки ЛПАКСС та ЛППГВС для моделювання лінійного тракту людини використовують модель ЛП у якій мовний тракт людини представляється довготерміновим прогнозувальником, що визначає періодичність МС [78,149]. Модель ЛП алгоритму ЛППГВС застосовується у процесі стиснення коефіцієнтів відбиття. У алгоритмі ЛППГВС модель ЛП застосовується у процесі стиснення параметрів ЛСП. Наведені алгоритми відрізняються оптимізаційними процедурами визначення параметрів МС. У таблиці 2.6 наведено результати порівняльного аналізу обчислювальної складності алгоритмів ЛПАКСС та ЛППГВС.

Таблиця 2.6

Порівняльний аналіз обчислювальної складності алгоритмів ЛПАКСС і ЛППГВС

Функціональний опис	ЛПАКСС	ЛППГВС
Аналіз КЛП, квантування, інтерполяція	21,3 %	20,1 %
Довготерміновий аналіз у відкритому та закритому циклах	23,2 %	24,1 %
Визначення індексу фіксованої кодової книги	19,4 %	15,8 %
Інше	36,1 %	40 %

Аналіз табл. 2.6 показав, що більше 60 процентів обчислювальної складності затрачається для визначення параметрів МС у процесі короткотермінової та довготермінової фільтрації та у процесі обчислення випадкового збудження [135].

Порівняння параметрів алгоритмів ЛПАКСС та ЛППГВС показує, що вони різні, а структури та принцип роботи короткотермінового, довготермінового синтезуючих фільтрів та модулів визначення випадкового збудження двох алгоритмів – однакові.

За результатами досліджень запропоновано ефективний метод транскодування стиснених МС між GSM 06.20 та G.729A, який дає

можливість виконувати пряме перетворення параметрів з одного формату в інший.

Модулі алгоритмів ЛПАКСС та ЛППГВС генерують КЛП, ВТ і параметри збудження.

Для здійснення прямого перетворення параметрів збудження, КЛП та ВТ враховано структуру та довжину кадрів форматів G.729A та GSM 06.20. Запропонований метод реалізується етапами: перетворення КЛП, перетворення ВТ та швидкий пошук в ФКК.

На першому етапі короткотерміновий синтезуючий фільтр $A_i(z)$ моделі спектрального перетворення МС має вигляд:

$$A_i(z) = \frac{1}{1 - \sum_{j=1}^{P} a_{ij} z^{-j}}, \quad 0 \le i \le 3 \tag{2.4}$$

де a_{ij} – КЛП;

P – порядок лінійного прогнозувальника (для алгоритмів ЛПАКСС та ЛППГВС $P=10$).

Інформація про КЛП кадру формату G.729A закодована у векторах ЛСП. Для обчислення значень ЛСП виконуються процедури декомпресії та інтерполяції КЛП кадрів формату G.729A. ЛСП перетворюються до КЛП a_{ij} через обчислення 11-ти відліків імпульсного відгуку аналізуючих фільтрів відповідно до математичної моделі:

$$A(z) = \frac{Q(z) + P(z)}{2} \tag{2.5}$$

де $A(z)$ – значення аналізуючого фільтру;

$Q(z)$, $P(z)$ – функції визначення коефіцієнтів ЛСП відповідно згідно з стандартом GSM 06.20.

Вираз (2.5) вимагає значних обчислень для отримання оптимальних рішень, тому пропонується використовувати метод обчислення енергії цільового сигналу для кожного тракту (рис. 2.8).

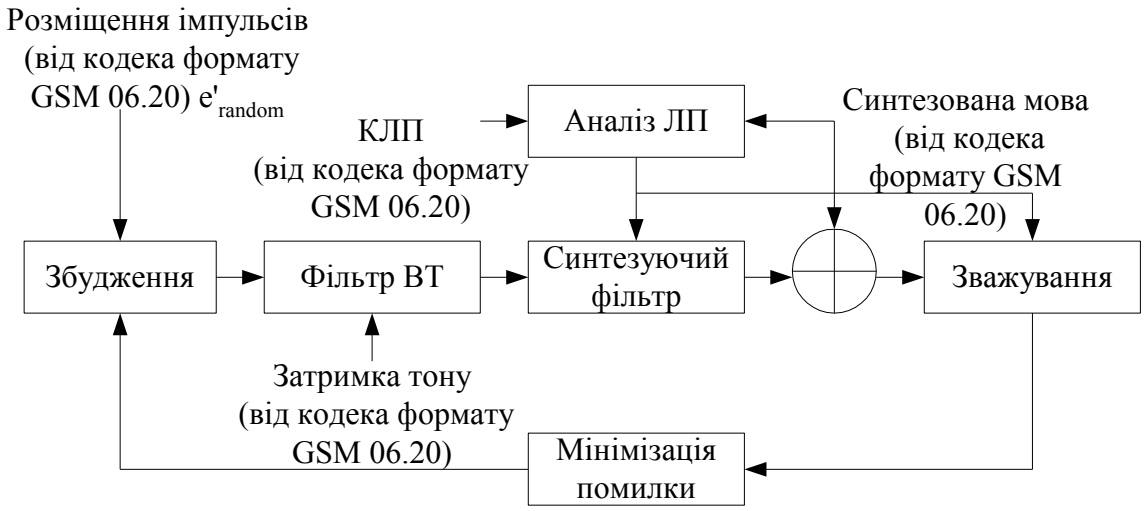


Рис.2.8. Визначення енергії цільового сигналу для кодека формату G.729A

Для визначення діапазону пошуку у ФКК використовується сигнал збудження з кадру формату GSM 06.20.

Далі виконується пошук найкращих кодових слів коефіцієнтів відбиття відповідно до математичних моделей, визначених в стандарті GSM 06.20.

На другому етапі обчислюється оптимальне значення ВТ. Обидва кодеки використовують процедуру підвищення частоти дискретизації та дробові значення ВТ: 0,33 для G.729A; 0,33, 0,166 і 0,5 для GSM 06.20. ВТ першого підкадру стискається незалежно від ВТ інших підкадрів. Стиснення значень ВТ наступних підкадрів виконується з врахуванням різниці поточної та попередньої затримки. Визначення ВТ починається з пошуку значення в розімкнутому циклі та продовжується у закритому циклі.

ВТ у форматі G.729A значно більша, ніж у форматі GSM 06.20, особливо на діапазонах зміни фонем, що пояснюється різним розміром кадрів та діапазонами зміни ВТ обидвох кодеків. ВТ підкадру, одержаного з підкадру формату G.729A, визначається як фіксована ВТ, передана підкадру формату GSM 06.20. На основі використання процесу фіксації ВТ центрального підкадру виконується прямий та зворотний пошук значення ВТ з метою ідентифікації параметрів *LAG1-LAG4* та формування траєкторії ВТ в межах кадру. Для виконання алгоритму прямого пошуку діапазон значень ВТ наступного підкадру обмежується відрізком $[-2^{M-1}+C;\ 2^{M-1}-1-C]$ тонових

рівнів (де *M*- визначає кількість біт для кодування значення ВТ, *C* – кількість тонових рівнів), що відповідають значенням ВТ поточного підкадру. Для реалізації алгоритму зворотнього пошуку діапазон значень ВТ обмежується відрізком $[-2^{M-1}+1+C;\ 2^{M-1}-C]$ тонових рівнів. Для обох алгоритмів пошуку, вибирається значення ВТ відповідно до якого значення нормалізованої кореляції в межах пошукового ряду є максимальним. Для кадру формату GSM 06.20 доцільно вибрати траєкторію ВТ з найменшою енергією помилки довготермінового прогнозувальника.

На третьому етапі запропонованого методу у модулях алгоритмів ЛПАКСС і ЛППГВС обчислюється випадкове збудження із залишку МС після оцінки ВТ. Значення випадкового збудження алгоритму ЛПАКСС визначається наступним чином:

$$e'_{random}[c_x] = e_{pitch}[c_x] + e_{random}[c_x] - e'_{pitch}[c_x] \qquad (2.6)$$

де $e_{pitch}[c_x]$ – функція випадкового збудження;

$e_{random}[c_x]$ – функція адаптивного збудження;

$e'_{pitch}[c_x]$ – функція збудження ВТ, що отримується після перетворення параметрів ВТ.

Транскодування з GSM 06.20 до формату G.729A доцільно реалізовувати за три етапи: перетворення КЛП, перетворення ВТ та швидкий пошук у ФКК.

Перетворення ЛСП залежить від довжини кадрів та виконується відповідно до методу лінійної інтерполяції. Для формування математичної моделі перетворення ЛСП присвоїмо номери підкадрам: *g1, g2, g3, g4* – кадру формату GSM 06.20; *b1, b2* – кадру формату G.729A; *c1, c2* – наступного кадру формату G.729A. Оскільки довжина одного кадру формату GSM 06.20 вдвічі більша за довжину кадру формату G.729A, то перетворення ЛСП відображає така математична модель:

$$LSP_{b2}^{G.729A}(i) = t \cdot LSP_{g1}^{GSM\ 06.20}(i) + y \cdot LSP_{g2}^{GSM\ 06.20}(i), \qquad (2.7)$$

$$LSP_{c2}^{G.729A}(i) = t \cdot LSP_{g3}^{GSM\ 06.20}(i) + y \cdot LSP_{g4}^{GSM\ 06.20}(i), \quad (2.8)$$

де $1 \le i \le 10$;

t, y – вагові коефіцієнти;

$LSP_{b2}^{G.729A}(i)$, $LSP_{c2}^{G.729A}(i)$ – ЛСП підкадрів формату G.729A;

$LSP_{g1}^{GSM\ 06.20}(i)$ $LSP_{g2}^{GSM\ 06.20}(i)$ $LSP_{g3}^{GSM\ 06.20}(i)$ $LSP_{g4}^{GSM\ 06.20}(i)$ – ЛСП підкадрів формату GSM 06.20.

Для перетворення ЛСП підкадру *b1* використано метод лінійної інтерполяції:

$$LSP_{b1}^{G.729A}(i) = 0{,}5 \cdot LSP_{b2}^{G.729A}(i) + 0{,}5 \cdot LSP_{c2}^{G.729A}(i), \quad (2.9)$$

де $LSP_{b1}^{G.729A}(i)$ - ЛСП підкадру *b1* формату G.729A.

Оскільки у процесі транскодування не виконуються процедури обчислення автокореляційних коефіцієнтів, рекурсії Дарбіна при визначення КЛП та перетворення КЛП в ЛСП.

Запропоновані математичні моделі (2.7)-(2.9) дозволяють скоротити часову затримку, яка виникає при аналізі КЛП, та зменшити обчислювальну складність у порівнянні з класичним методом,

Оскільки, стиснений МС формату GSM 06.20 описується в сегменті голосних та приголосних звуків, то визначення ВТ для підкадру формату G.729A відбувається окремо для кожного сегменту звуків. У випадку, коли МС належить сегменту голосних звуків, то ВТ у розімкнутому циклі алгоритму ЛПАКСС є цілочисельним значенням тонової затримки *T* одержаним з кадру формату GSM 06.20. Далі виконується процедура пошуку ВТ в замкнутому циклі для значень: *T, T+1/3, T+2/3* і *T+1*. У випадку, коли МС належить сегменту приголосних звуків, тонову затримку визначити неможливо. Тому для оцінки тонової затримки використовується значення ВТ, отримане з попереднього підкадру формату G.729A.

Результатом виконання процедури пошуку в ФКК алгоритму ЛПГВС є значення функції вартості описаної в стандарті GSM 06.20, яка досягає максимуму.

2.2.3. Аналіз часових характеристик запропонованих методів транскодування стиснених мовних сигналів

Для аналізу часових характеристик запропонованого методу транскодування між G.723.1 та G.729A приймемо, що: p^E_A, p^E_B – час, необхідний для виконання операції стиснення згідно з алгоритмами БКМД та ЛПАКСС, відповідно; p^D_A, p^D_B – час, необхідний для виконання операції стиснення згідно з алгоритмами БКМД та ЛПАКСС, відповідно; p^{TR}_{AB}, p^{TR}_{BA} – час, необхідний для транскодування з G.723.1 до G.729A та навпаки.

У таблицях 2.7 та 2.8 наведено результати порівняльного аналізу часових характеристик методів транскодування з G.723.1 в G.729A та навпаки.

Таблиця 2.7

Порівняльний аналіз часових характеристик методів транскодування з G.723.1 в G.729A

Примітка	Назва операції	Затримка, мс	
		Тандем	Запропонований метод
G.723.1	Буферизація	37,5	37,5
	Стиснення	p^E_A	p^E_A
Проміжне опрацювання	Декомпресія	p^D_A	p^{TR}_{AB}
	Стиснення	$3* p^E_A$	-
	Затримка	5	0
G.729A	Декомпресія	$3*p^D_A$	$3*p^D_A$
Загальна затримка		$42,5+ p^E_A + p^D_A+3*(p^E_A + p^D_A)$	$37,5+ p^E_A+p^{TR}_{AB}+3*p^D_A$

Таблиця 2.8

Порівняльний аналіз часових характеристик методів транскодування з G.729A в G.723.1

Примітка	Назва операції	Затримка, мс	
		Тандем	Запропонований метод
1	2	3	4
G.723.1	Буферизація	35	35
	Стиснення	$3*p^E_B$	$3*p^E_B$

Проміжне опрацювання	Декомпресія	$3*p^{D}_{B}$	p^{TR}_{BA}
	Стиснення	p^{E}_{A}	-
1	2	3	4
	Затримка	5	0
G.729A	Декомпресія	p^{D}_{A}	p^{D}_{A}
Загальна затримка		$40+ 3*(p^{E}_{B} + p^{D}_{B}) p^{E}_{A} + p^{D}_{A}$	$35+ p^{E}_{A}+p^{TR}_{BA}+3*p^{E}_{B}$

Проведений аналіз часових характеристик показав, що загальна затримка, як мінімум, на 5 мс. менша, ніж у класичному методі, що пояснюється відсутністю у запропонованому методі процесу аналізу КЛП.

2.3. Метод багатоступінчастого мікшування мовних сигналів

Завдання побудови нових методів мікшування МС зумовлене сьогоднішніми умовами розвитку комп'ютерних систем та компонент. У протоколах підтримки сеансів зв'язку спостерігається тенденція до використання декількох алгоритмів стиснення МС [8,80,89]. Зокрема, у протоколі H.323 регламентовано використання п'яти алгоритмів стиснення [89], у протоколі SIP – чотирьох [8], у протоколі MGSP - 3 [80]. Проведений в першому розділі аналіз алгоритмів мікшування МС показав, що відомі алгоритми володіють рядом недоліків, основним з яких є змішування МС одного формату. Дана особливість алгоритмів мікшування МС обмежує використання алгоритмів стиснення МС регламентованих протоколами підтримки сеансів зв'язку.

У процесі мікшування блоків даних $L_{i,j}(q_x)$ часто виникає проблема неузгодженості значень частот квантування та бітрейду відліків блоків даних. Крім того, блоки даних $L_{i,j}(q_x)$ *j*-тих джерел можуть відрізнятись розміром.

На відміну від одноступінчастого мікшування, методи багатоступінчастого мікшування характеризуються кращою швидкодією, яка отримується за рахунок відсутності затримок пов'язаних з часом очікування *i*-тих блоків

даних [32]. Однак алгоритми виконання цих методів слабо дослідженні та мало описані в літературі.

З врахуванням наведених особливостей та, дотримуючись представлених у першому розділі принципів мікшування, запропоновано новий метод багатоступінчастого мікшування на базі пам’яті з довільним доступом (адресної пам’яті) [31,111].

Структурна схема блоку мікшування МС (рис. 2.9) ілюструє реалізацію запропонованого методу.

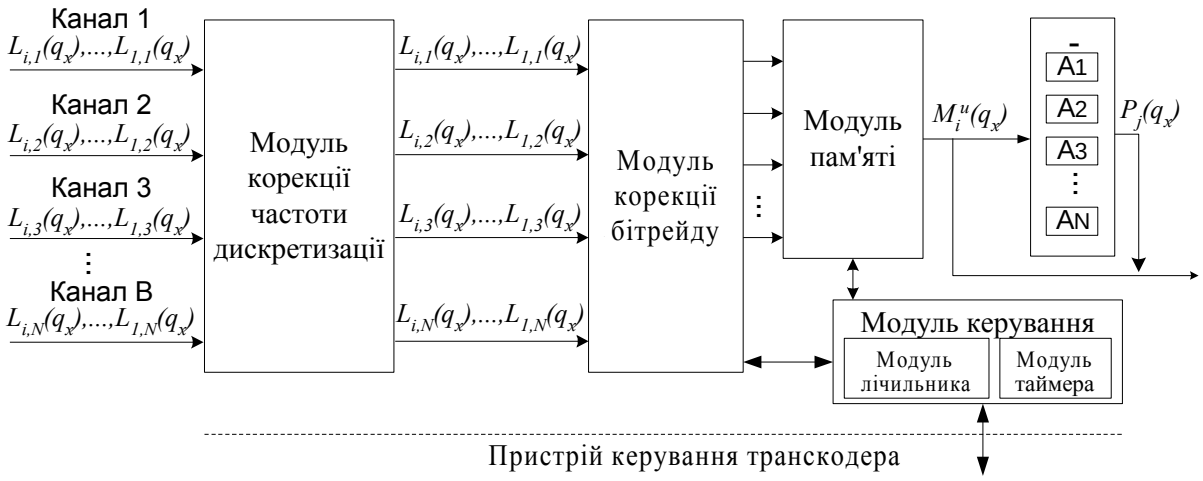


Рис. 2.9. Структурна схема блоку багатоступінчастого мікшування

Як видно з рис. 2.9, значення відліків МС q_x через B вхідних каналів блоку мікшування (БМ) поступають у модуль корекції частоти дискретизації. Якщо частота дискретизації відліків q_x не рівна 8 КГц, то виконується алгоритм передискретизації. В іншому випадку дані передаються у модуль корекції бітрейду, де значення відліків q_x приводяться до 16-ти бітового формату та синхронізується розмір блоку даних. З модуля корекції бітрейду блоки даних $L_{i,j}(q_x)$ передаються у модулі пам’яті з довільним доступом, де для кожного i-го блоку виділяється комірка пам’яті обсягом $Lb(L_{i,j}(q_x))$ байт, якій присвоюється відповідна адреса. Процес мікшування виконується у комірках модуля пам’яті відповідно до принципу роботи методів

багатоступінчастого мікшування. Тому значення відліків q_x з i-го блоку даних сумуються у комірках пам'яті по мірі їх надходження згідно виразу (1.1).

Усі комірки модуля пам'яті адресуються відповідно до відмітки RTP-часу. Пристрій керування виконує процес заповнення комірок модуля пам'яті блоками даних $L_{i,j}(q_x)$. Модулі лічильника та таймера визначають момент часу в який необхідно передати значення $M_i^u(q_x)$ з комірки модуля пам'яті у модуль виключення власного блоку даних. Після отриманням даних $M_i^u(q_x)$ виконується процедура виключення власного блоку даних для кожного активного джерела згідно з виразом (1.2).

Розглянемо детальніше функції кожного модуля БМ:

- модуль корекції частоти дискретизації призначений для узгодження частоти дискретизації відліків q_x. Якщо частота дискретизації відліків не рівна 8 КГц, то виконується алгоритм передискретизації. Частота 8 КГц вибрана базуючись на проведеному порівняльному аналізі алгоритмів стиснення МС, більшість з який забезпечують саме таку частоту дискретизації. У іншому випадку, блоки даних передаються у модуль корекції бітрейду;
- модуль корекції бітрейду дає можливість привести значення відліків q_x до одного бітрейду (16 біт) та забезпечити їх передачу у комірки модуля пам'яті. У цьому модулі синхронізуються розмір блоків даних $Lb(L_{i,j}(q_x))$;
- модуль пам'яті складається з комірок пам'яті обсягом $Lb(L_{i,j}(q_x))$ байт, що помічаються відповідно до відмітки RTP-часу. Кожній комірці присвоюється відповідна адреса. Ще однією функцією модуля пам'яті є передача блоків даних $L_{i,j}(q_x)$ у модуль виключення власного блоку даних учасника сеансу зв'язку;
- модуль виключення власного блоку даних учасника виконує виключення власного блоку даних для кожного активного учасника сеансу зв'язку згідно виразу (1.2) та відправляє блоки даних до модулів компресії;

- модуль керування на основі аналізу інформації про характеристики *i*-тих кадрів та інформації з модулів корекції бітрейду та керування сигналізує про можливість передачі блоків даних у комірки модуля пам'яті. Модуль керування складається з лічильника і таймера. Якщо кількість *i*-тих блоків даних рівна кількості активних учасників *N*, то модуль керування сигналізує про готовність передачі блоку даних $M_i^u(q_x)$ з комірки модуля пам'яті у наступний модуль. Для зменшення затримок пов'язаних з часом очікування втрачених у процесі передачі кадрів вмонтовано таймер, який визначає інтервали пакетування по аналогії з [125].

Для досліджння запропонованого методу розроблено алгоритм мікшування wav-файлів [53], що описується наступними кроками (рис. 2.11):

1. Одержання інформації про кількість wav-файлів (*N*) та їх wave-форму. Wave-форма категорії PCM має наступний вигляд:

$$wave\_form = wave + ftm\text{-}чанк + data\text{-}чанк, \quad (2.9)$$

де, *wave* - сигнатура wave форми,

ftm-чанк – чанк з інформацією про МС,

data-чанк – чанк з МС.

Для забезпечення ефективної роботи алгоритму використовується значення наступних полів wave-форми: частота дискретизації (freq), кількість біт, якими представляється відлік сигналу (bits), число каналів (chanell), розмір data-чанку (ckSize), послідовність байт, яка описує сигнал (signal).

2. Підготовка data-чанку wav-файлу до мікшування. На цьому етапі проводяться наступні дії:
   а) Перевіряється поле freq, що міститься в структурі WaveFormat ftm-чанку. Якщо значення цього поля не рівне 8 КГц, то до вмісту поля signal data-чанку застосовується алгоритм передискретизації, з метою встановлення частоти дискретизації 8 КГц.

b) Перевіряється поле bits, що міститься в структурі WaveFormat ftm-чанку. Якщо значення цього поля не рівне 16 біт, то відліки сигналу, які містяться у полі signal data-чанку, приводиться до 16-ти бітного формату.

c) Перевіряється поле chanell, що міститься в структурі WaveFormat. Якщо його значення рівне 1, то вміст поля signal data-чанку дублюється, створюючи на базі поля signal двохмірний масив.

d) Визначається максимальна довжина поля signal data-чанку wav-файлів, інформація про які була одержана на попередньому кроці.

e) Значення всіх полів signal, data-чанку wav-файлів, окрім значення поля signal wav-файлу з максимальною довжиною, приводяться до максимальної довжини поля signal, що була одержана в результаті виконання етапу 2d. Для виконання цього етапу необхідно знайти різницю між максимальною довжиною data-чанку wav-файлу та довжиною поточного data-чанку wav-файлу. Далі формується масив, що складається з нулів та має розмір одержаної різниці. Значення одержаного масиву додається до значень поля signal data-чанку поточного wav-файлу.

3. Змішування значень полів signal data-чанку wav-файлів, одержаних на першому кроці.

З метою дослідити залежність швидкості виконання реалізованого методу від кількості wav-файлів та їх характеристик, отримано експериментальні дані над випадковим чином вибраними файлами (табл. 2.9). Кількість wav-файлів для експериментів визначає кількість каналів g, по яких блоки даних $L_{i,j}(q_x)$ поступають в БМ (рис. 2.10).

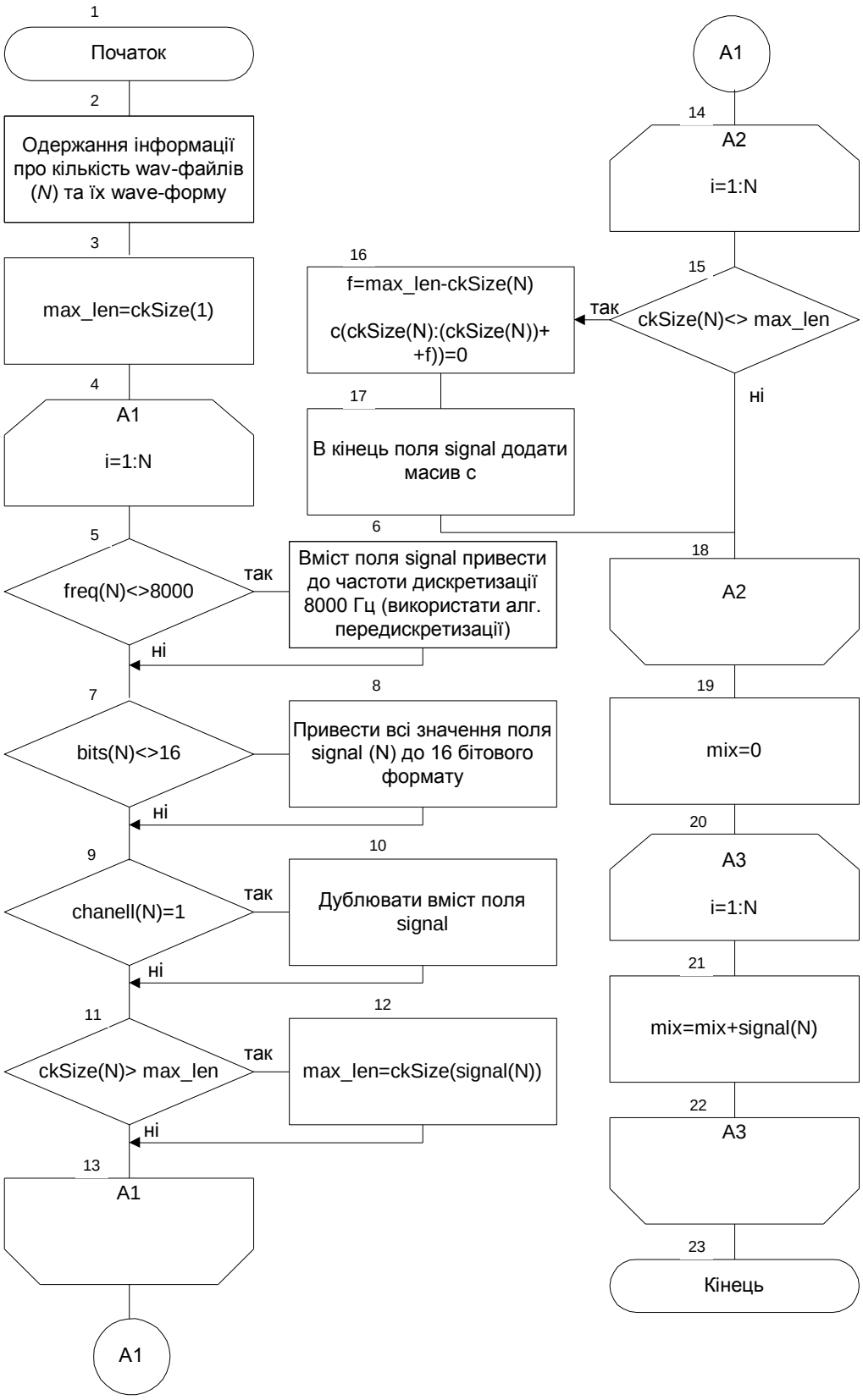


Рис. 2.10. Блок-схема алгоритму запропонованого методу багатоступінчастого мікшування

Таблиця 2.9

Характеристики wav-файлів

№ п\п	Кількість відліків	Бітрейд	Частота дискретизації, КГц	Довжина блоку даних, біт	Довжина блоку даних, сек.	Кількість каналів	Опис
1	14592	8	8	240016	30	1	фрагмент мови, жіночий голос
2	240588	16	8	481176	30	1	жіночий спів
3	1323008	16	44,1	2646016	30	1	звук віаланчелі
4	1323008	16	44,1	5292032	30	2	музичний фрагмент
5	661520	16	22,05	2646080	30	2	стандартний звук Windows (Chimes)
6	330768	16	11,025	661536	30	1	звук, аплодисменти
7	240016	8	20,1	603250	30	1	фрагмент мови, чоловічий голос

Розмір data-чанку всіх wav-файлів (табл. 2.9) значно більший від довжин кадрів, у яких передається блоки даних $L_{i,j}(q_x)$ від модулів декомпресії до БМ. Довжина кадрів із блоками даних змінюється від 0.125 до 40 мс (в залежності від алгоритму стиснення МС).

Оскільки, така мала довжина data-чанку не дасть об’єктивних результатів, то всі обчислення в експериментах проводились над data-чанками wav-файлів з довжиною блоків даних 30 секунд.

На рисунку 2.11 показано зміну часу мікшування двох wav-файлів.

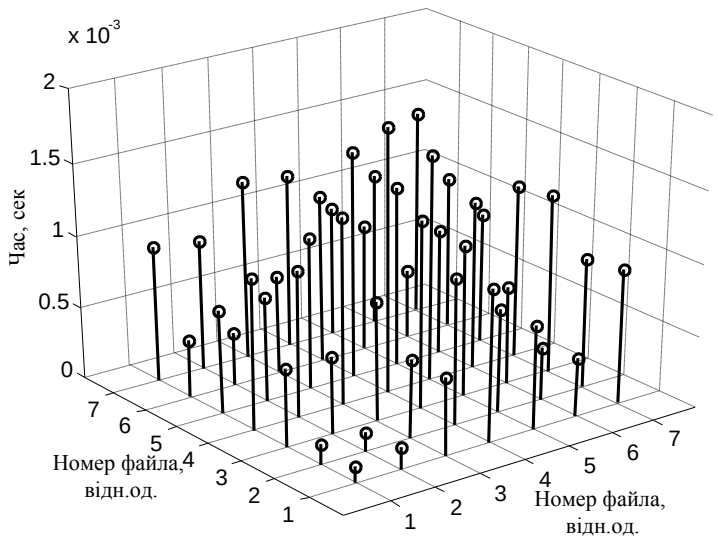


Рис. 2.12. Залежність часу мікшування двох wav-файлів від їх порядкових номерів

На рис. 2.12 показано результати мікшування трьох і більше wav-файлів, вибраних випадковим чином. Для спрощення ідентифікації wav-файлів, на графіку введено умовне позначення, що відображає номери wav-файлів (див. табл. 2.1) які мішкувались. Для прикладу, запис 1246 означає, що мішкувались відліки МС, які містяться у першому, другому, четвертому та шостому wav-файлах.

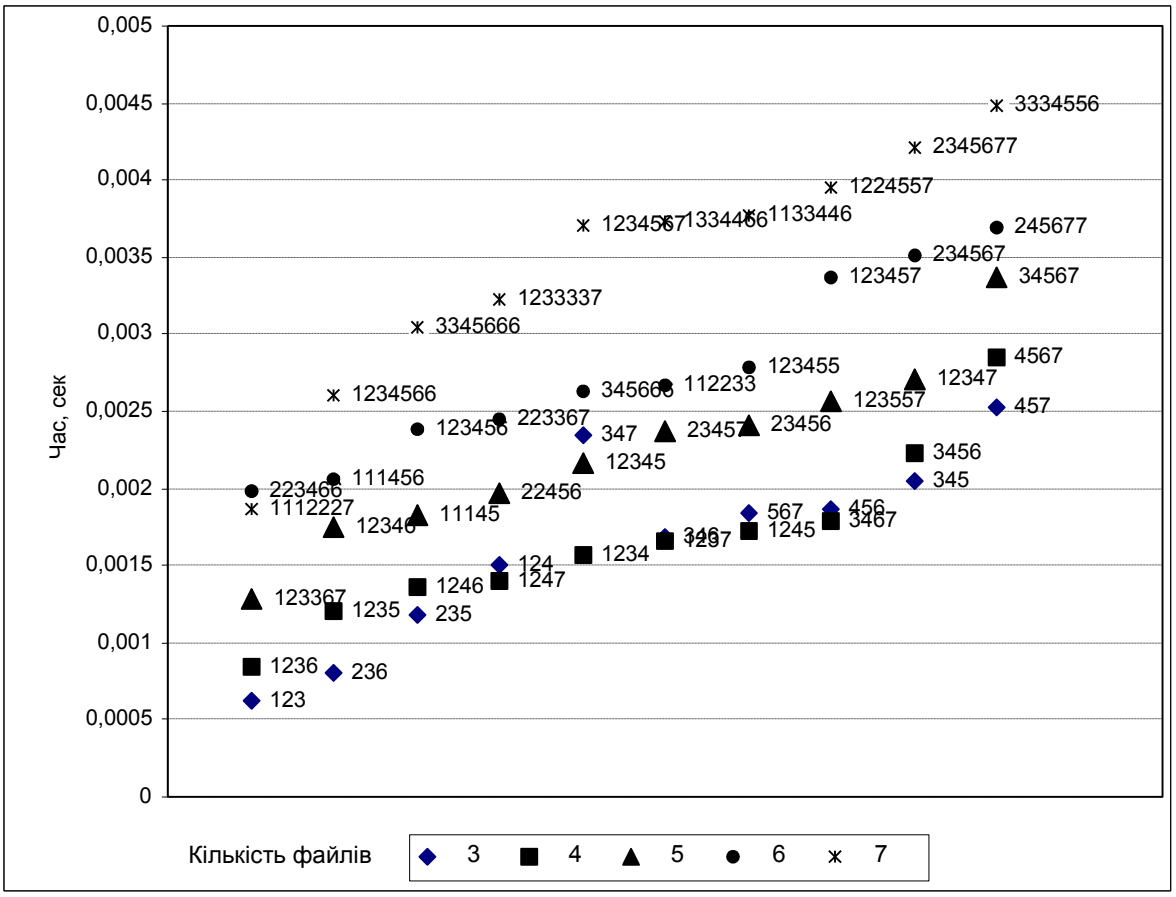


Рис. 2.12. Час мікшування трьох і більше wav-файлів

Результати досліджень показують, що на час мікшування значно впливають: частота дискретизації; кількість біт, якими кодується відлік; кількість каналів; розмір data-чанку.

Із побудованих графіків слудує, що при частоті дискретизації 8 КГц, 16-ти бітному кодуванню відліку та стерео звучанню запропонований метод є найефективніший.

Кількісні показники, представлені на рис. 2.11 та рис. 2.12, отримані засобами профілювання розробленого алгоритму в середовищі Matlab 6.0 без використання підпрограм виведення графічної інформації на процесорі Celeron 666 МГц.

Запропонований метод дає можливість опрацьовувати значення відліків з блоків даних $L_{i,j}(q_x)$, що були одержані шляхом декомпресії стиснених МС різних форматів. Процес мікшування починається при надходжені хоча б двох блоків даних у цільову комірку пам’яті з довільним доступом, що дозволяє уникати черг у буфері БМ та зменшити затримки пов’язані з часом очікування блоків даних.

ВИСНОВКИ

1. Запропоновано метод перетворення форматів стиснених МС між GSM 06.20 та G.729A, що враховує структурну схожість модулів короткотермінової фільтрації, довготермінової фільтрації та випадкового збудження алгоритмів ЛППГВС та ЛПАКСС, яка дає можливість провести пряме перетворення параметрів, згенерованих даними модулями. Розроблений метод дозволяє зменшити часову затримку, як мінімум, на 5 мс. і апаратну складність у порівнянні з класичним методом.
2. Запропоновано метод перетворення форматів стиснених МС між G.723.1 та G.729A, який дає можливість виконувати пряме перетворення ряду параметрів кадру одного формату у параметри кадру іншого та передбачає виконання чотирьох етапів: перетворення ЛСП, перетворення ВТ і пошук у АКК та ФКК. Розроблений метод дозволяє зменшити часову затримку, апаратну складність декодера та покращити якість мовлення.
3. Запропоновано метод багатоступінчастого мікшування МС на основі пам’яті з довільним доступом, який дає можливість опрацьовувати значення відліків з блоків даних, що були одержані шляхом декомпресії стиснених МС різних форматів. Процес мікшування починається при надходженні хоча б двох блоків даних у цільову комірку пам’яті з довільним доступом, що дозволяє уникати черг у буфері БМ та зменшити затримки пов’язані з часом очікування блоків даних.

РОЗДІЛ 3
СТРУКТУРИ БАГАТОКАНАЛЬНИХ ТРАНСКОДЕРІВ СТИСНЕНИХ МОВНИХ СИГНАЛІВ

3.1. Принципи оброблення кадрів із стисненими мовними сигналами

У цьому розділі сформовано принципи оброблення кадрів із блоками даних $L_{i,j}(c_x)$ багатоканальним транскодером, що працює відповідно до класичного та запропонованих у другому розділі методів.

3.1.1. Оброблення кадрів відповідно до класичного методу транскодування стиснених мовних сигналів

Кадри із стисненими МС характеризуються множиною векторів параметрів $Z=\{Zj \mid Zj= [A_{дж}, A_{пр}, a, L_{i,j}(c_x)]$, де $A_{дж}$ – адреса джерела повідомлення, $A_{пр}$ – адреса приймача повідомлення, a – алгоритм стиснення МС $(a \in F)$, $L_{i,j}(c_x)$ – блок даних із стисненим МС. На вхід багатоканального транскодера з кожного каналу g $(g=1,...B)$ надходять блоки даних $L_{i,j}(c_x)$, які зберігаються у вхідному комутаторі даних (ВхКом). Вхідний комутатор, отримавши інформацію про початковий a $(a \in F)$ та кінцевий b $(b \in F)$ формат даних, від пристрою керування (ПрК) відправляє блоки даних у модулі транскодера для подальшого оброблення. Отже, робота алгоритму транскодування стиснених МС пов’язана з інформацією про вхідний та вихідний формат стиснених МС, а також кількість активних учасників сеансу зв’язку [30].

Згідно з стандартом H.323v2 на вхід багатоканального транскодера, можуть поступати наступні формати стиснених МС: G.111, G.722, G.723.1, G.728 та G.729A [89]. Отже, у вихідний комутатор (ВихКом), функцією якого є зберігання та відправлення даних у вихідні канали g' $(g'=1,...B')$, також може зберігати будь-який із цих форматів. Тому, в загальному випадку

максимальна кількість комбінацій транскодування, що може виконуватись багатоканальним транскодером, рівна $m*(m-1)$ [30].

Для визначення всіх можливих варіантів оброблення блоків даних $L_{i,j}(c_x)$ приймемо, що [30]:

- a_i={G.111, G.722, G.723.1, G.728 , G.729A};
- b_i={G.111, G.722, G.723.1, G.728 , G.729A};
- N – кількість джерел, що формують блоки даних $L_{i,j}(c_x)$ (якщо $N>2$, то сеанс зв'язку називається багато абонентським) [22, 24].

Визначимо основні блоки багатоканального транскодера та побудуємо алгоритм його роботи. Крім пристрою керування, вхідного, вихідного та проміжного комутатора у структуру багатоканального транскодера введено наступні модулі:

- $D_1(c_x)$ виконує декомпресію блоків даних $L_{i,j}(c_x)$ відповідно до алгоритму підсмугової адаптивно-диференціальної імпульсно-кодової модуляції, формат G.722;
- $D_2(c_x)$ виконує декомпресію блоків даних $L_{i,j}(c_x)$ відповідно до алгоритму БКМД, формат G.723.1;
- $D_3(c_x)$ виконує декомпресію блоків даних $L_{i,j}(c_x)$ відповідно до алгоритму лінійного прогнозування, що генерується кодом з низькою затримкою (LD-CELP), формат G.728;
- $D_4(c_x)$ виконує декомпресію блоків даних $L_{i,j}(c_x)$ відповідно до алгоритму ЛПАКСС, формат G.729A;
- $M_i^u(q_x)$ виконує мікшування відліків q_x;
- $K_1(p_x)$ виконує стиснення блоків даних $L_{i,j}(p_x)$ відповідно до алгоритму підсмугової адаптивно диференціальної імпульсно-кодової модуляції, формат G.722;
- $K_2(p_x)$ виконує стиснення блоків даних $L_{i,j}(p_x)$ відповідно до алгоритму БКМД, формат G.723.1;

- $K_3(p_x)$ виконує стиснення блоків даних $L_{i,j}(p_x)$ відповідно до алгоритму лінійного прогнозування, що генерується кодом з низькою затримкою, формат G.728;
- $K_4(p_x)$ виконує стиснення блоків даних $L_{i,j}(p_x)$ відповідно до алгоритму ЛПАКСС, формат G.729A.

Алгоритм оброблення блоків даних $L_{i,j}(c_x)$ реалізується наступними етапами [30]:

1) Якщо $a_i = b_i$ і $N \leq 2$, то блоки даних $L_{i,j}(c_x)$ відправляються у ВихКом.
2) Якщо $a_i = b_i$ і $N>2$, то виконується декомпресія $L_{i,j}(c_x)$ (блок декомпресії $(D_a(c_x)=q_x)$), виконується мікшування $(N-1)$ i-тих блоків даних $(M_i^u(q_x)=p_x)$. Сформовані блоки даних $L_{i,j}(p_x)$ передаються у проміжний комутатор (ПрКом), а з нього у відповідні модулі стиснення. Після стиснення $(K_b(p_x)=c_x)$, блоки даних $L_{i,j}(c_x)$ відправляються у вихідний комутатор.
3) Якщо $a_i \neq b_i$ і $N \leq 2$, то необхідно виконується декомпресія блоків даних $L_{i,j}(c_x)$ (модуль декомпресії $(D_a(c_x)=q_x)$) та передача декомпресованих даних у проміжний комутатор. З проміжного комутатора блоки даних $L_{i,j}(q_x)$ відправляються у відповідні модулі стиснення. Після стиснення $(K_b(p_x)=p_x)$ блоки даних $L_{i,j}(p_x)$ відправляються у вихідний комутатор.
4) Якщо $a_i \neq b_i$ і $N>2$, то виконується декомпресія $L_{i,j}(c_x)$ (модуль декомпресії $(D_a(c_x)=q_x)$), виконується мікшування $(N-1)$ i-тих блоків даних $(M_i^u(q_x)=p_x)$. Сформовані блоки даних $L_{i,j}(p_x)$ передаються у проміжний комутатор (ПрКом), а з нього у відповідні модулі стиснення. Після стиснення $(K_b(p_x)=c_x)$, блоки даних $L_{i,j}(c_x)$ відправляються у вихідний комутатор.

У таблиці 3.1 в табличній формі наведено алгоритм оброблення блоків даних $L_{i,j}(c_x)$ багатоканальним транскодером відповідно до класичного методу транскодування стиснених МС.

Таблиця 3.1

Оброблення блоків даних $L_{i,j}(c_x)$ багатоканальним транскодером

№ п/п	Правила перевірки кадрів	Алгоритм роботи транскодера стиснених МС
1.	$a_i = b_i$ і $N \le 2$	ВхКом→ВихКом
2.	$a_i = b_i$ і $N > 2$ або $a_i \neq b_i$ і $N > 2$	ВхКом→$D_a(c_x)$→$M_i^u(q_x)$→ПрКом→$K_b(p_x)$→ ВихКом
3.	$a_i \neq b_i$ і $N \le 2$	ВхКом→ $D_a(c_x)$ →ПрКом→ $K_b(p_x)$→ВихКом

3.1.2. Оброблення кадрів відповідно до запропонованих методів транскодування стиснених мовних сигналів

Запропоновані у другому розділі методи транскодування стиснених МС працюють із блоками даних $L_{i,j}(c_x)$, що містять інформацією про параметри МС. Зокрема, алгоритм ЛПАКСС використовує наступні параметри: ЛСП, затримка АКК, парність тонової затримки, індекс ФКК, знак ФКК, коефіцієнти підсилення кодової книги. Дані параметри використовуються для синтезу МС згенерованих учасниками сеансу зв'язку. Процес оброблення блоків даних $L_{i,j}(c_x)$ багатоканальним транскодером відповідно до запропонованих методів, буде напряму залежати від вхідного *a* та вихідного *b* формату МС, а також кількості активних учасників сеансу зв'язку (N).

На практиці часто зустрічаються ситуації, коли на вхід багатоканального транскодера поступають кадри, формати яких не регламентовані стандартом H.323v2, тому у структуру багатоканального транскодера введено модулі декомпресії $(D_1(c_x))$ та стиснення $(K_1(p_x))$, що змінюють свою структуру відповідно до алгоритму z $(z \in F)$.

Враховуючи особливості побудови та функціонування запропонованих методів транскодування у структуру багатоканального транскодера введено наступні модулі:

- $D_1(c_x)$ виконує декомпресію блоків даних $L_{i,j}(c_x)$ відповідно до алгоритму z $(z \in F)$;

- $D_2(c_x)$ виконує декомпресію блоків даних $L_{i,j}(c_x)$ відповідно до алгоритму БКМД, формат G.723.1;
- $D_3(c_x)$ виконує декомпресію блоків даних $L_{i,j}(c_x)$ відповідно до алгоритму ЛППГВС, формат GSM 06.20;
- $D_4(c_x)$ виконує декомпресію блоків даних $L_{i,j}(c_x)$ відповідно до алгоритму ЛПАКСС, формат G.729A;
- $M_i^u(q_x)$ виконує мікшування відліків q_x;
- $K_1(p_x)$ виконує стиснення блоків даних $L_{i,j}(p_x)$ відповідно до алгоритму z ($z \in F$);
- $K_2(p_x)$ виконує стиснення блоків даних $L_{i,j}(p_x)$ відповідно до алгоритму БКМД, формат G.723.1;
- $K_3(p_x)$ виконує стиснення блоків даних $L_{i,j}(p_x)$ відповідно до алгоритму ЛППГВС, формат GSM 06.20;
- $K_4(p_x)$ виконує стиснення блоків даних $L_{i,j}(p_x)$ відповідно до алгоритму ЛПАКСС, формат G.729A.

Алгоритм оброблення блоків даних $L_{i,j}(c_x)$ відповідно до запропонованих методів реалізується наступними етапами:

1) Якщо $a_i = b_i$ і $N \leq 2$, то блоки даних $L_{i,j}(c_x)$ відправляються у вихідний комутатор;
2) Якщо $a_i = b_i$ і $N>2$, то виконується декомпресія $L_{i,j}(c_x)$ (блок декомпресії $(D_a(c_x)=q_x)$), виконується мікшування $(N-1)$ i-тих блоків даних $(M_i^u(q_x)=p_x)$. Сформовані блоки даних $L_{i,j}(p_x)$ передаються у проміжний комутатор (ПрКом), а з нього у відповідні модулі стиснення. Після стиснення $(K_b(p_x)=c_x)$, блоки даних $L_{i,j}(c_x)$ відправляються у вихідний комутатор.
3) Якщо $a_i \neq b_i$ і $N>2$:
   3.1) якщо a_i=G.729A, b_i=G.723.1, то виконується транскодування з G.729A до G.723.1, відповідно до запропонованого у п.2.2.1

методу. У процесі транскодування частково використовуються процедури регламентовані у блоках $D_4(c_x)$ і $K_2(p_x)$ (див. рис. 2.7).

3.2) якщо a_i=*G.729A,* b_i=*GSM 06.20*, то виконується транскодування з G.729A до GSM 06.20, відповідно до запропонованого у п.2.2.2 методу. У процесі транскодування частково процедури регламентовані у модулях $D_4(c_x)$ і $K_3(p_x)$ (див. рис. 2.8).

3.3) якщо a_i=*G.723.1,* b_i=*G.729A*, то виконується транскодування з G.723.1 до G.729A відповідно до запропонованого у п.2.2.1 методу. У процесі транскодування частково використовуються процедури регламентовані у модулях $D_2(c_x)$ і $K_4(p_x)$ (див. рис. 2.4).

3.4) якщо a_i=*GSM 06.20,* b_i=*G.729A*, то виконується транскодування з GSM 06.20 до G.729A, відповідно до запропонованого у п.2.2.2 методу. У процесі транскодування частково використовуються процедури регламентовані у модулях $D_3(c_x)$ і $K_4(p_x)$ (див. рис. 2.4).

3.5) інакше, виконується декомпресія (модуль декомпресії $(D_a(c_x)=q_x)$ та передача блоків даних $L_{i,j}(c_x)$ у проміжний комутатор, з якого блоки даних відправляються у відповідні модулі стиснення. Після стиснення $(K_b(p_x)=c_x)$ блоки даних $L_{i,j}(c_x)$ відправляються у вихідний комутатор.

4) Якщо $a_i \neq b_i$ і $N>2$, то виконується декомпресія $L_{i,j}(c_x)$ (модуль декомпресії $(D_a(c_x)=q_x)$), виконується мікшування $(N-1)$ i-тих блоків даних $(M_i^u(q_x)=p_x)$. Сформовані блоки даних $L_{i,j}(p_x)$ передаються у проміжний комутатор (ПрКом), а з нього у відповідні модулі стиснення. Після стиснення $(K_b(p_x)=c_x)$, блоки даних $L_{i,j}(c_x)$ відправляються у вихідний комутатор.

3.2. Створення структур багатоканальних транскодерів стиснених мовних сигналів

Структурна організація багатоканального транскодера стиснених МС представляється у виді трьох складових: процедури декомпресії, мікшування та компресії. У даному підрозділі наведено результати щодо створення структур багатоканальних транскодерів стиснених мовних сигналів. Запропоновані структури орієнтовані на використання в обладнанні багатоканальних конвергентних мультисервісних мереж зв'язку.

3.2.1. Структура багатоканального транскодера, що працює відповідно до класичного методу транскодування стиснених мовних сигналів

Для розробки структури багатоканального транскодера використаємо інформацією, одержану вище, що дозволило побудувати таблицю, яка визначає зв'язки між модулями багатоканального транскодера стиснених МС, що працює відповідно до класичного методу транскодування стиснених МС (таблиця 3.2).

Таблиця 3.2

Зв'язки між блоками багатоканального транскодера

	ВхКом	$D_a(c_x)$	$M_i^u(q_x)$	ПрКом	$K_b(p_x)$	ВихКом
ВхКом		→	→	→		→
$D_a(c_x)$			→	→		
$M_i^u(q_x)$				→		→
ПрКом					→	
$K_b(p_x)$						→
ВихКом						
ПрК	→,←		→,←	→,←		→,←

Стрілки наведені в таблиці показують зв'язки мід модулями транскодера.

На основі таблиці 3.2 побудовано структуру багатоканального транскодера стиснених МС (рис. 3.1).

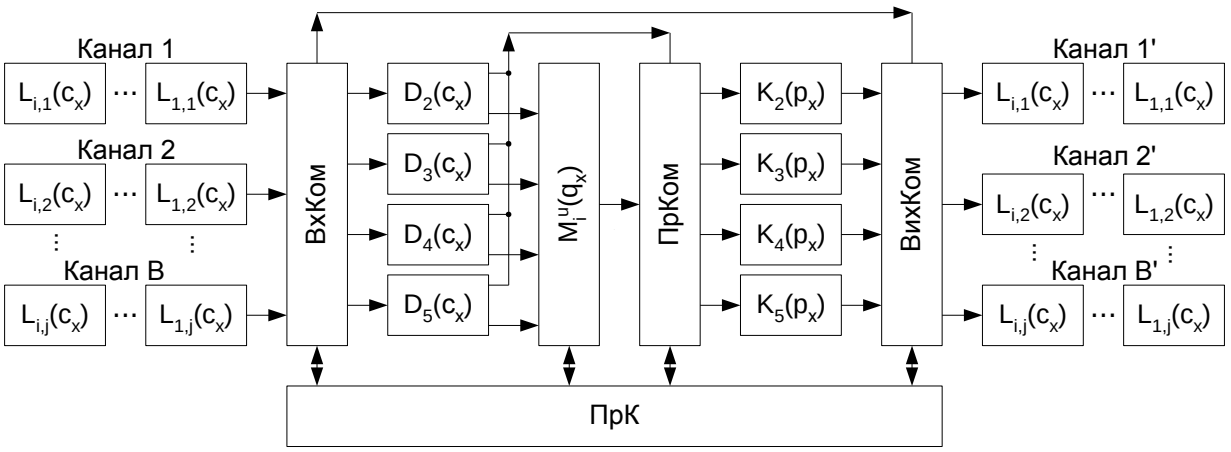


Рис. 3.1. Структура багатоканального транскодера, що працює відповідно до тандему

З рис.3.1 слідує, що багатоканальний транскодер є лінійною конвеєрною схемою із послідовно з'єднаних вхідного комутатора (ВхКом), декомпресорів $(D_1(c_x)\text{-}D_4(c_x))$, БМ $(M_i^u(q_x))$, проміжного комутатора (ПрКом), модулів стиснення $(K_1(p_x)\text{-}K_4(p_x))$ та вихідного комутатора (ВихКом). Розглянемо детальніше процес проходження блоків даних у багатоканальному транскодері. Чергові блоки даних $L_{i,j}(c_x)$ з активних каналів передачі подаються у ВхКом, який паралельно отримує інформацію з ПрКом про початковий a $(a \in F)$ та кінцевий формат даних b $(b \in F)$, а також кількість учасників сеансу зв'язку N. Далі, відповідно до запропонованого алгоритму, виконується опрацювання вхідних блоків даних $L_{i,j}(c_x)$, які з вхідного комутатора відправляються у модулі транскодера для подальшого перетворення. Функціями проміжного комутатора є зберігання та відправлення блоків даних $L_{i,j}(p_x)$ у відповідні модулі стиснення. Функціями вихідного комутатора є зберігання та видача у канали зв'язку опрацьованих даних. Керування процесом транскодування виконується пристроєм керування, який відправляє у модулі транскодера інформацію про алгоритми

стиснення та мікшування МС, а також інформацію про стан опрацювання блоків даних $L_{i,j}(c_x)$.

На рис. 3.2 наведено деталізовану структуру модуля $D_1(c_x)$.

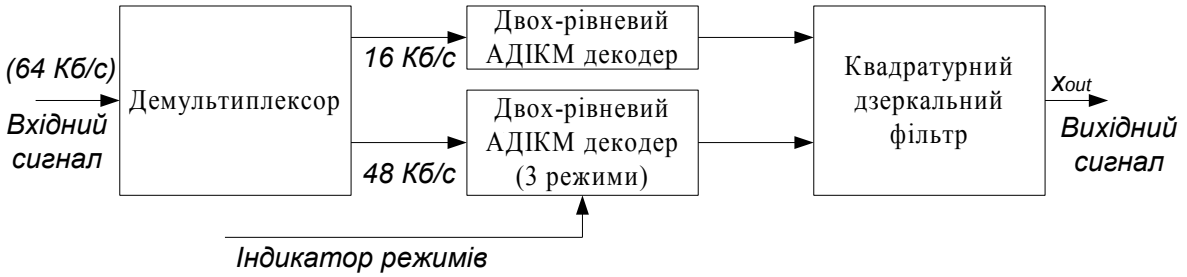


Рис. 3.2. Структурна схема модуля $D_1(c_x)$

Опис принципів декодування кадру формату G.722 наведено у роботі [94].

На рис. 3.3 наведена деталізована структура модуля $D_2(c_x)$.

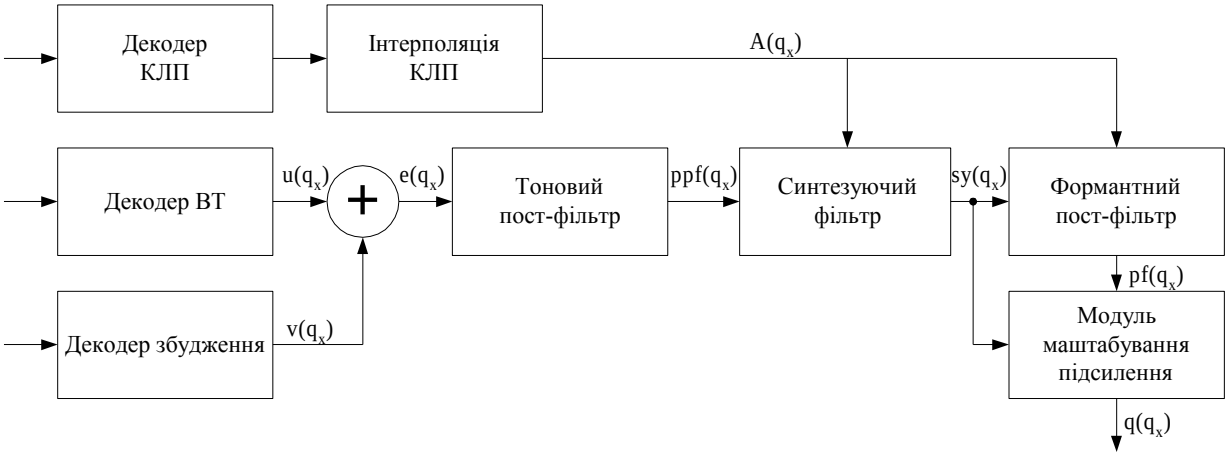


Рис. 3.3. Структурна схема модуля $D_2(c_x)$

Опис принципів декодування кадру формату G.723.1 наведено у роботі [71].

На рис. 3.4 наведено деталізовану структуру модуля $D_3(c_x)$.

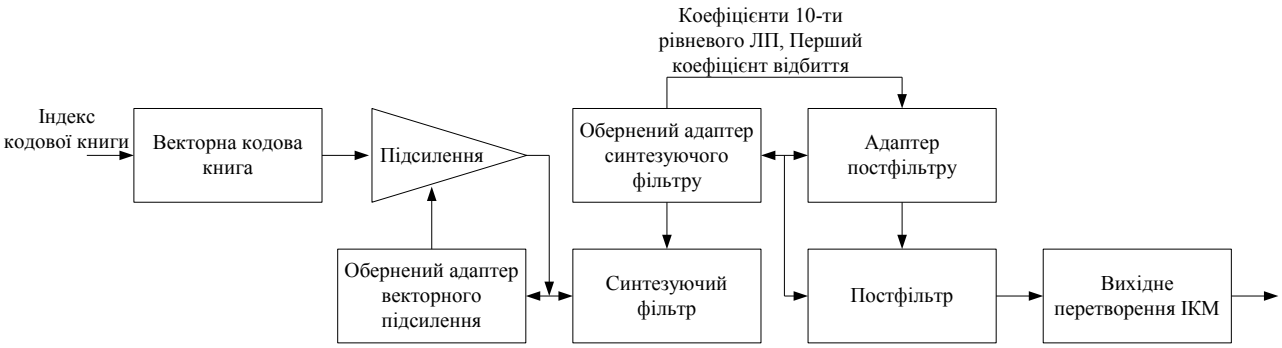


Рис. 3.4. Структурна схема модуля $D_3(c_x)$

Опис принципів декодування кадру формату G.728 наведено у роботі [92].

На рис. 3.5 наведено деталізовану структуру модуля $D_4(c_x)$.

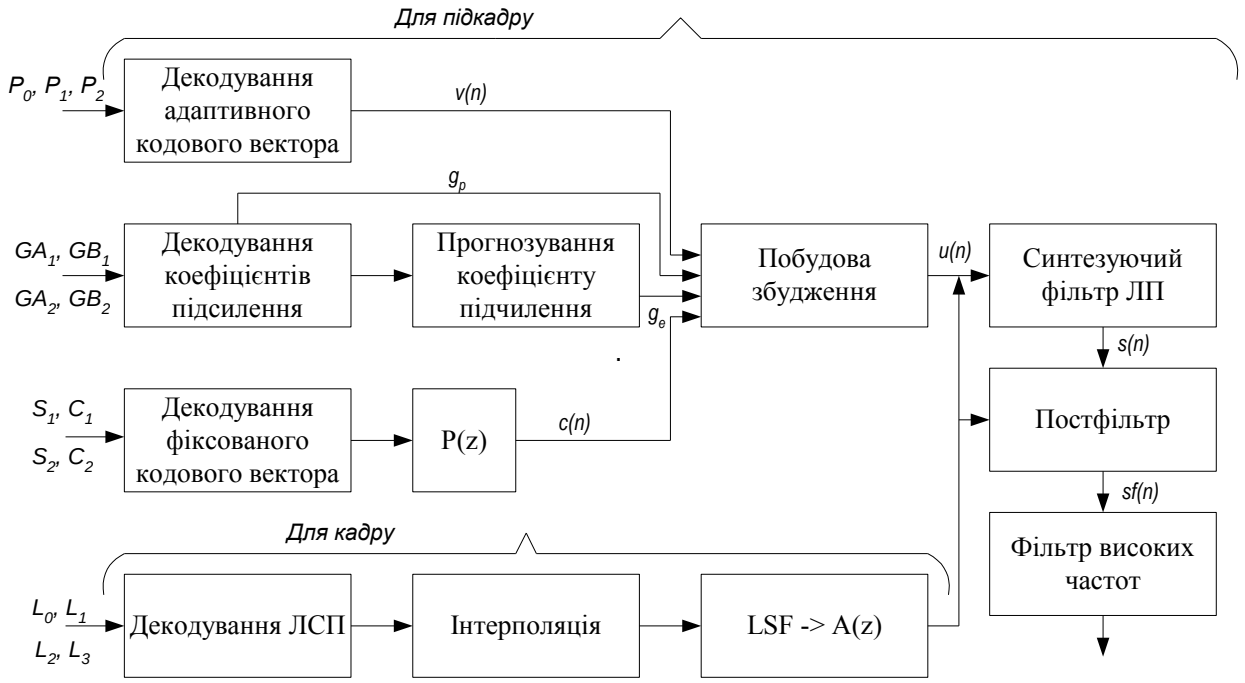


Рис. 3.5. Структурна схема модуля $D_4(c_x)$

Опис принципів декодування кадру формату G.729A наведено у роботах [96, 131].

На рис. 3.6 наведено деталізовану структуру блоку $K_1(p_x)$.

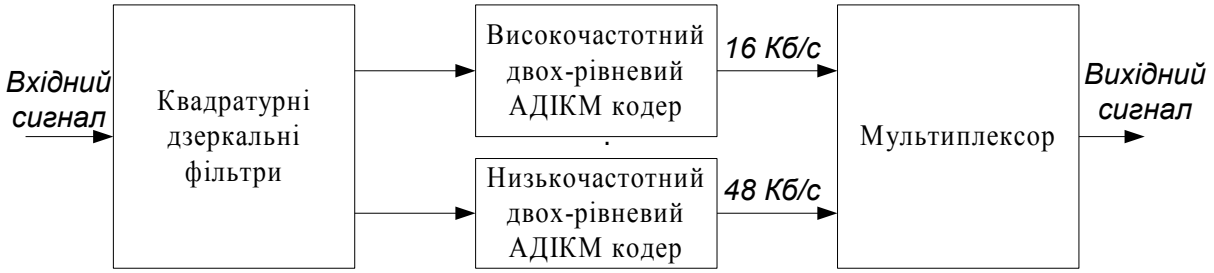


Рис. 3.6. Структурна схема модуля $K_1(p_x)$

Опис принципів стиснення кадру формату G.722 наведено у роботі [94].

На рис. 3.7 наведено деталізовану структуру модуля $K_2(p_x)$.

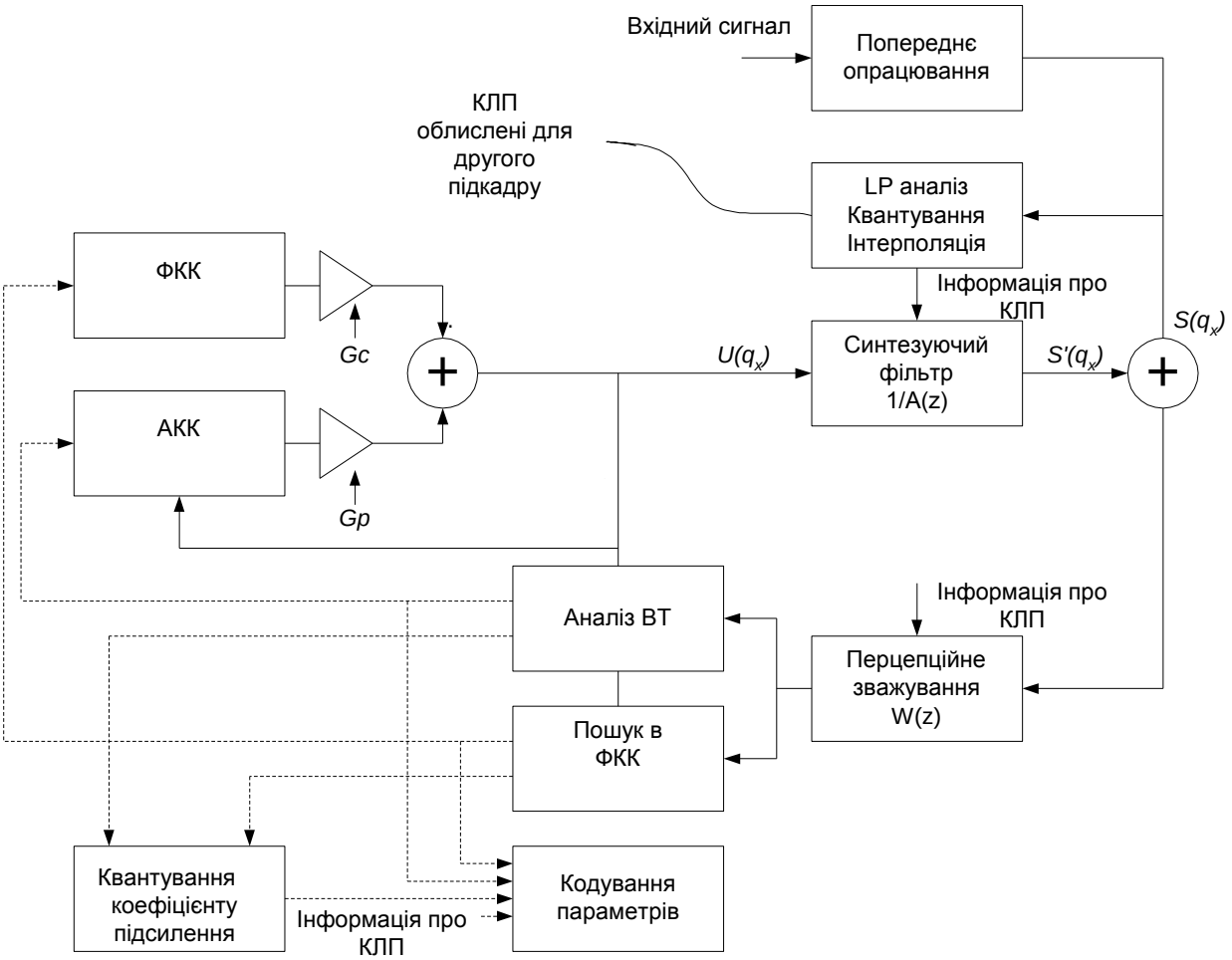


Рис. 3.7. Структурна схема модуля $K_2(p_x)$

На рис. 3.8 наведено деталізовану структуру модуля $K_3(p_x)$.

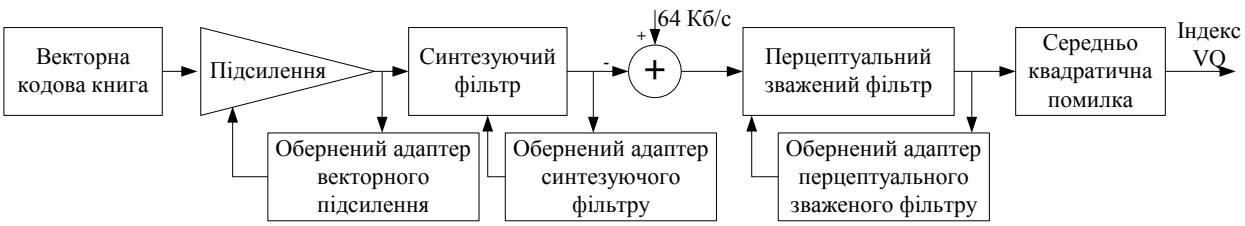


Рис.3.8. Структурна схема модуля $K_3(p_x)$

Опис принципів стиснення кадру формату G.728 наведено у роботі [92].

На рис. 3.9 наведено деталізовану структуру модуля $K_4(p_x)$.

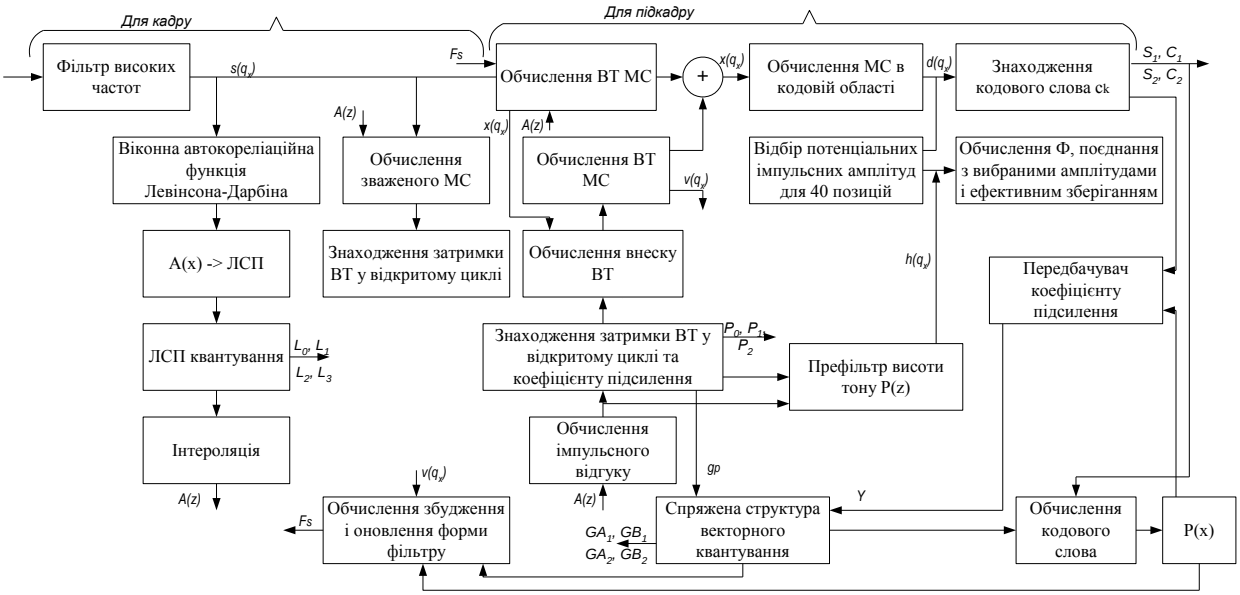


Рис.3.9. Структурна схема модуля $K_4(p_x)$

Опис принципів стиснення кадру формату G.729A наведено у роботах [96, 131].

Структура модуля $M_i^u(q_x)$ наведена на рис. 2.11, а в пункті 2.3 та роботах [31, 111] описано принцип роботи методу багатоступінчастого мікшування на базі пам’яті з довільним доступом.

Удосконалені структурні схеми дають змогу ефективно виконувати транскодування стиснених МС, відповідно до класичного методу, між

наступними форматами: G.711, G.722, G.723.1, G.728, G.729A. Перелічені формати регламентовані стандартом H.323v2.

3.2.2. Структури багатоканальних транскодерів, що працюють відповідно до запропонованих методів транскодування стиснених мовних сигналів

У другому розділі запропоновано два методи транскодування стиснених МС, що реалізують найбільш популярні алгоритми стиснення МС гібридного классу: ЛПГВС, БКМД та ЛПАКСС. Спрощена структурна схема для запропонованих методів транскодування буде аналогічна структурі багатоканального транскодера, що працює відповідно до класичного методу транскодування стиснених МС (рис. 3.1). Однак функції модулів та алгоритм роботи запропонованих методів суттєво відрізняються.

Деталізуємо структури основних блоків багатоканального транскодера, що працює відповідно до запропонованих методів транскодування стиснених МС. На базі запропонованого у пункті 3.1.2 алгоритму оброблення блоків даних $L_{i,j}(c_x)$ багатоканальним транскодером, побудовано структури багатоканального транскодера стиснених МС для кожної пари форматів:

- з G.723.1 до G.729A;
- з G.729A до G.723.1;
- з G.729A до GSM 06.20;
- з GSM 06.20 до G.729A.

На рис. 3.10 проілюстровано процес транскодування транскодування з G.723.1 до G.729A відповідно до запропонованого методу.

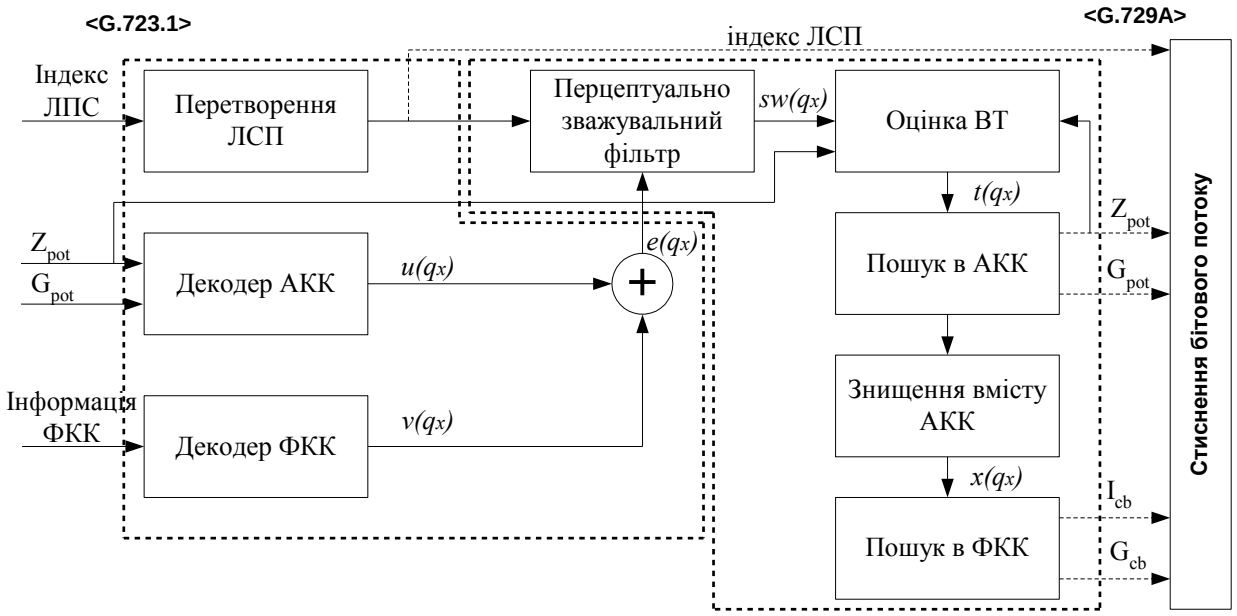


Рис. 3.10. Структурна схема транскодування з G.723.1 до G.729A

Необхідно відмітити, що ліва половина рис. 2.4 містить модулі декодера алгоритму БКМД, а права – модулі кодера алгоритму ЛПАКСС.

На рис. 3.11 наведено структурну схему транскодування з G.729A до G.723.1.

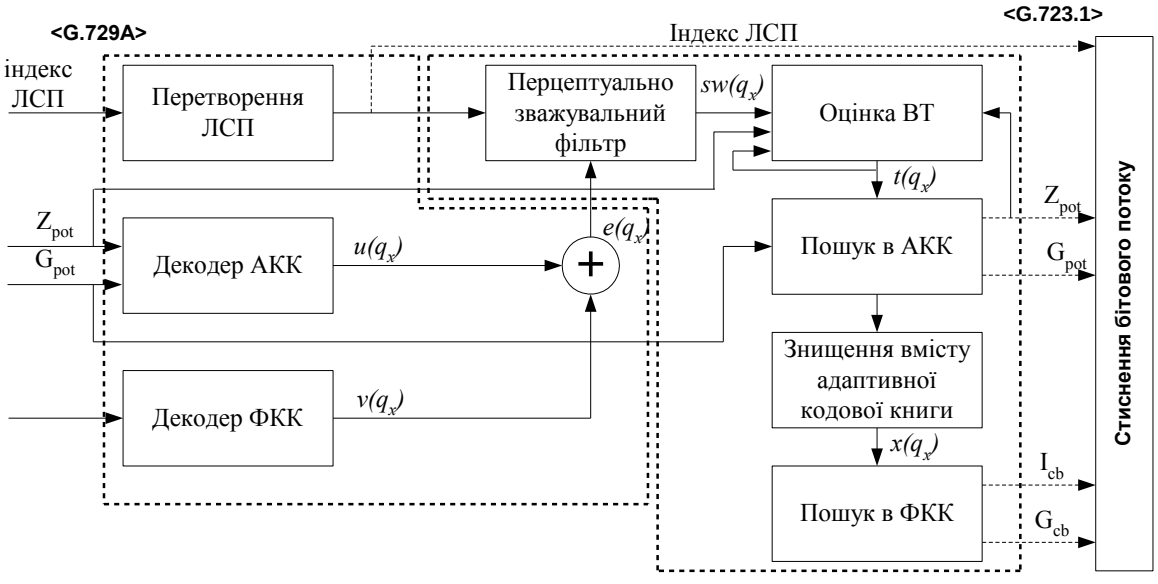


Рис.3.11. Структурна схема транскодування з G.729A до G.723.1

Ліва половина рис. 3.11 містить модулі декодера алгоритму ЛПАКСС, а права – модулі кодера алгоритму БКМД.

На рис. 3.12 наведено структурну схему транскодування з G.729A до GSM 06.20.

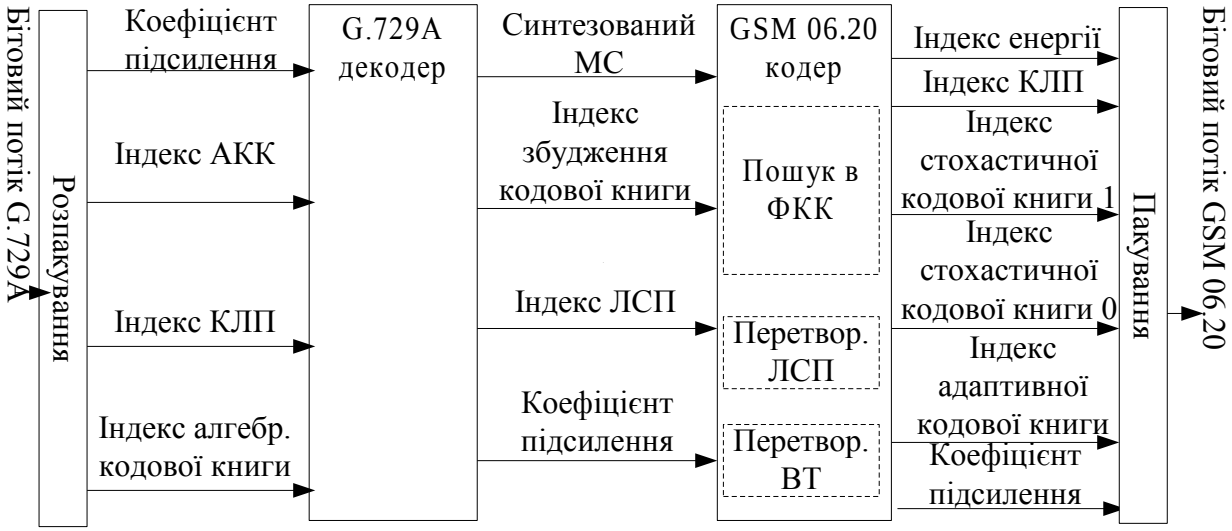


Рис. 3.12. Структурна схема транскодування з G.729A до GSM 06.20

На рис. 3.13 наведено структурну схему транскодування з GSM 06.20 до G.729A.

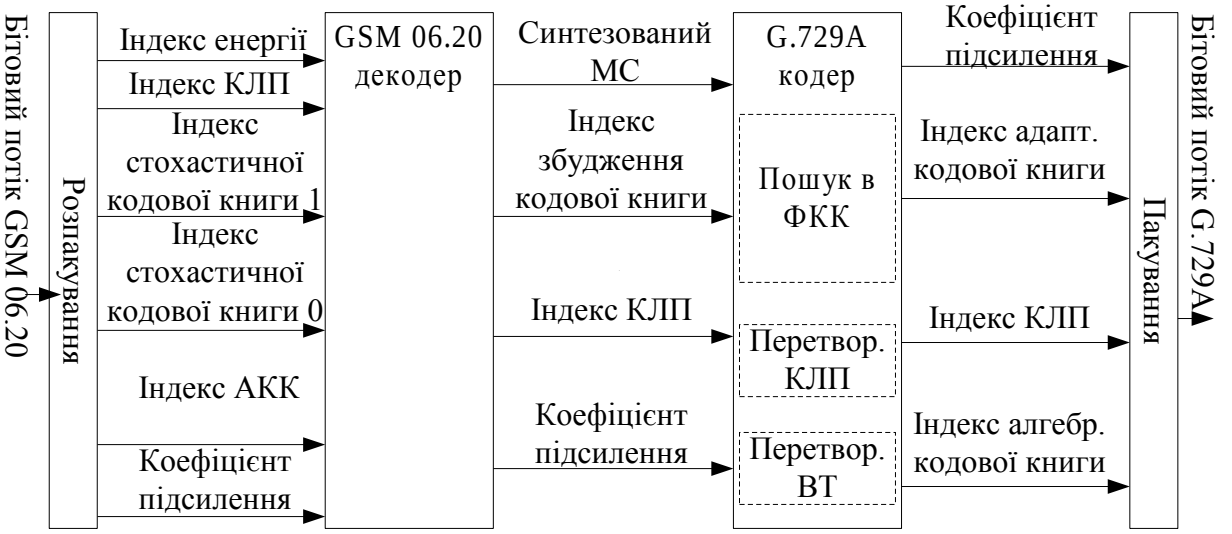


Рис. 3.13. Структурна схема транскодування з GSM 06.20 до G.729A

Удосконалені структури орієнтовані на використання в мережному обладнанні багатоабоненських конвергентних мереж та дають можливість обробляти формати стиснених МС регламентовані стандартом H.323v2. Використання удосконалених структур транскодерів дозволить підвищити продуктивність оброблення блоків даних $L_{i,j}(c_x)$.

ВИСНОВКИ

1. Сформовано принципи та запропоновано алгоритми оброблення кадрів із блоками даних $L_{i,j}(c_x)$. Запропоновані алгоритми орієнтовані на використання в комп'ютерних засобах транскодування стиснених МС, що вмонтовуються в мережне обладнання багатоабонентських конвергентних мультисервісних мереж.

2. Удосконалено структуру багатоканального транскодера стиснених МС, що працює відповідно до тандему, яка дає можливість опрацьовувати формати стиснених МС регламентовані стандартом H.323v2.

3. Удосконалено структури багатоканальних транскодерів стиснених МС, які працюють відповідно до запропонованих у другому розділі методів транскодування стиснених МС та дають змогу виконувати перетворення форматів між G.729A та G.723.1, а також між G.729A та GSM 06.20.

РОЗДІЛ 4
ПРОЕКТУВАННЯ КОМП'ЮТЕРНИХ ЗАСОБІВ КРИПТОГРАФІЧНОГО ЗАХИСТУ СТИСНЕНИХ МОВНИХ СИГНАЛІВ

4.1. Задача ефективного проектування комп'ютерних засобів криптографічного захисту стиснених мовних сигналів

Передача даних незахищеними комп'ютерними мережами, відповідно до протоколів H.323, SIP та MGSP, повинна проводитись із врахуванням можливих загроз щодо конфіденційності, цілісності та автентичності інформації [4,11,21,26,39,55,58,61].

Задача ефективного проектування криптозахищених пристроїв перетворення форматів стиснених МС займає одне з чільних місць у вирішенні загальної проблеми захисту [4,11,39].

У сучасних телекомунікаційних мережах широко використовуються протоколи захищеної передачі даних, де вирішення завдань конфіденційності, цілісності та автентичності інформації досягається шляхом криптографічного захисту даних [13,16,19]. Криптографічний захист стиснених МС передбачає використання спеціальних засобів, методів та заходів для вирішення наступних завдань [4,39]: запобігання втраті кадрів із інформацією про МС під час їх передачі від джерела до приймача, що може бути спричинена навмисними або ненавмисними діями чи спотвореннями в каналі зв'язку; запобігання просочуванню мовної інформації за рахунок несанкціонованого прослуховування переговорів в каналах телекомунікацій та запобігання зміні мовного повідомлення за допомогою модифікації сеансу мовних фраз або індивідуальних особливостей учасника сеансу зв'язку. Такого роду завдання входять до складу проблеми безпеки мовного зв'язку, яка останнім часом є особливо актуальною [4,11,39,55,16].

Часто для комплексного вирішення перелічених вище завдань використовують протокол H.235, що є частиною стандарту H.323 [93]. Однак

даний протокол має ряд недоліків. Зокрема, недосконалі механізми захисту передачі кадрів із блоками даних $L_{i,j}(c_x)$ від мобільних терміналів [152], не регламентовані механізми безпечної передачі кадрів з інформацією про МС згідно з протоколами серій H.26X та T.12X [142]. Тому, для комплексного захисту кадрів із блоками даних $L_{i,j}(c_x)$ використовують протокол IPSec, який застосовується на транспортному рівні моделі взаємодії відкритих систем та дає змогу зменшити недоліки H.235 [103].

Протокол IPSec використовується для забезпечення цілісності, автентичності та конфіденційності даних, що передаються незахищеними комп'ютерними мережами [103]. Головною перевагою IPSec, яка зумовила його широке використання, є можливість шифрування і/або автентифікування всієї інформації, що передається на транспортному рівні моделі OSI. Однак, невисока продуктивність роботи програмованих процесорів, ресурси яких використовуються для реалізації протоколу зменшують продуктивність мультимедійних додатків та комп'ютерних засобів. Крім того, додавання додаткових службових полів цим протоколом до результуючого кадру значно збільшує його розмір. Тому актуальною є задача ефективного проектування спеціалізованих процесорів підтримки протоколу IPSec оптимізованих для опрацювання кадрів з блоками даних $L_{i,j}(c_x)$.

Проведений аналіз протоколу показав, що компонентом, який найбільше впливає на час оброблення кадрів та його розмір, є операційний пристрій (ОП) спеціалізованого процесора IPSec, який реалізує криптографічні модулі [58]. В основу побудови ОП закладено принцип апаратного відображення структури виконуваного алгоритму на ОП тракту оброблення даних [32,33]. Структури ОП для виконання алгоритмів симетричного блокового шифрування досліджено у [20], а структури алгоритмів хешування у [26]. Тому в даному розділі досліджуються структури комплексних ОП процесорів підтримки протоколу IPSec, у яких

мінімізовано часові характеристики оброблення кадрів із блоками даних $L_{i,j}(c_x)$.

4.2. Базові структури та часові характеристики операційного пристрою підтримки протоколу IPSec

Сервіси протоколу IPSec дозволяють вибрати базові криптографічні модулів захисту для протоколів та встановити значення криптографічних ключів, що використовуються для встановлення сеансу зв'язку. Стандартом IPSec регламентовано два протоколи: автентифікування – AH (Authentication Header) [102] та комбінований протокол автентифікування/шифрування – ESP (Encapsulating Security Payload) [101], що в комплексі забезпечують такі сервіси:

- керування доступом;
- цілісність без встановлення з'єднання;
- автентифікування джерела даних;
- захист від відтворення;
- конфіденційність і частковий захист від аналізу потоку даних.

Кожен з цих сервісів вносить додаткову затримку при опрацюванні кадру із інформацією про МС, оскільки додаються відповідні поля до результуючого кадру.

Розглянемо структуру ОП виконання базових криптографічних модулів відповідно до протоколу IPSec в режимі передавання (рис. 4.1) і приймання даних (рис. 4.2).

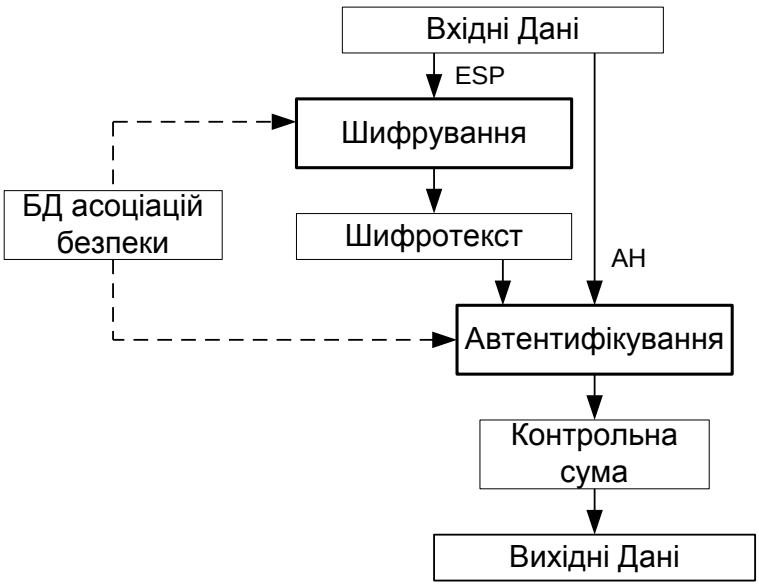


Рис. 4.1. Структура ОП виконання базових криптографічних операцій відповідно до протоколу IPSec в режимі передавання даних

Вхідними даними ОП є кадри з інформацією про МС. Залежно від методу передавання даних (приймання/ передавання/суміщення) протоколу IPSec (AH/ ESP/AH+ESP) та режиму передачі даних на виході отримуються захищені кадри різного розміру. Часова затримка вихідних кадрів буде визначається вихідними характеристиками ОП.

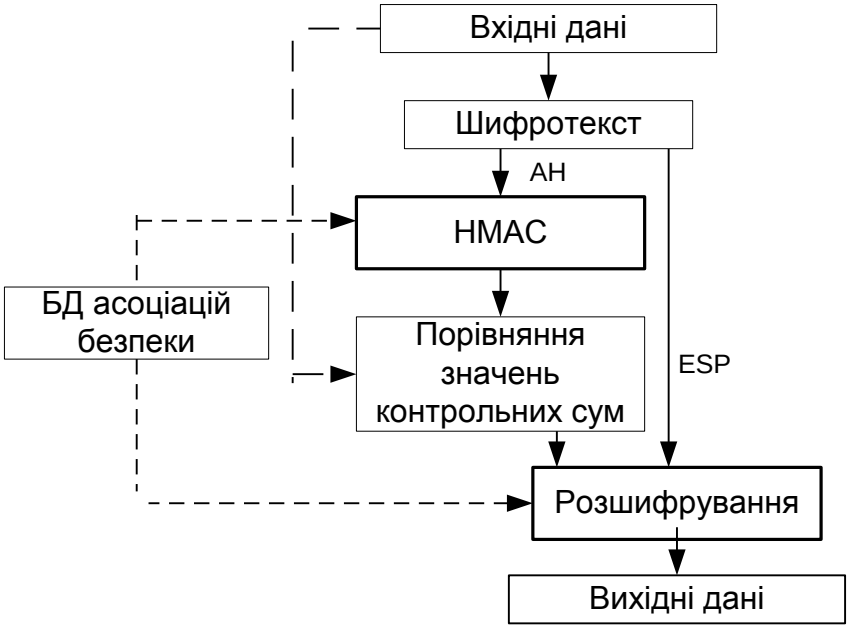


Рис. 4.2. Структура ОП виконання базових криптографічних операцій відповідно до протоколу IPSec в режимі приймання даних

Проведемо аналіз ОП IPSec під час передавання/приймання даних різними протоколами в транспортному режимі з використанням кадрів стандарту IPv4. Для транспортного режиму AH, дані AH розміщуються безпосередньо після оригінального IP заголовку (рис. 4.3). Автентифікуванню підлягає весь кадр, за виключенням змінних полів в заголовку IPv4, які обнуляються для обчислення значення HMAC [58,103]. Транспортний режим AH ідентифікує інкапсульовану в кадр інформацію про МС, а також окремі частини IP заголовку.

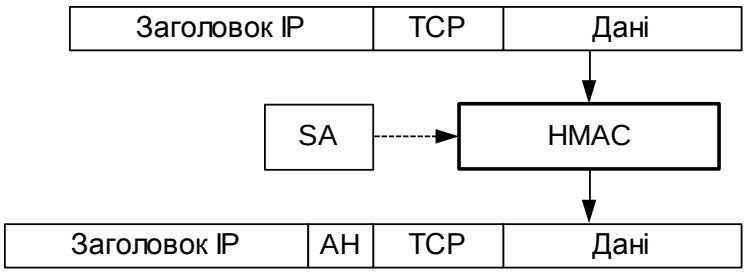


Рис. 4.3. Використання протоколу AH для оброблення кадру

Такт ОП виконання протоколу AH– t_{IPSec}, та час оброблення кадру з інформацією про МС T_{IPSec} складають відповідно:

$$t_{IPSec} = max(t_{SNx}, t_{SNx}^{1}), \tag{4.1}$$

$$T_{IPSec} = T_{SNx} + T_{SNx}^{1}. \tag{4.2}$$

де t_{SNx}, t_{SNx}^{1} – такти роботи ОП функції автентифікування, за якою обчислюється HMAC, визначені в [58,103]; T_{SNx}, T_{SNx}^{1} – часи роботи ОП функції автентифікування, за якою обчислюється HMAC, визначені в [58,103].

Транспортний режим передачі даних ESP забезпечує шифрування даних. При цьому заголовок ESP розміщується безпосередньо перед заголовком транспортного рівня, а трейлер (містить поля заповнювача, довжини заповнювача і наступного заголовку) розміщується після кадру з інформацією про МС (рис. 4.4). Весь кадр транспортного рівня разом з трейлером шифруються.

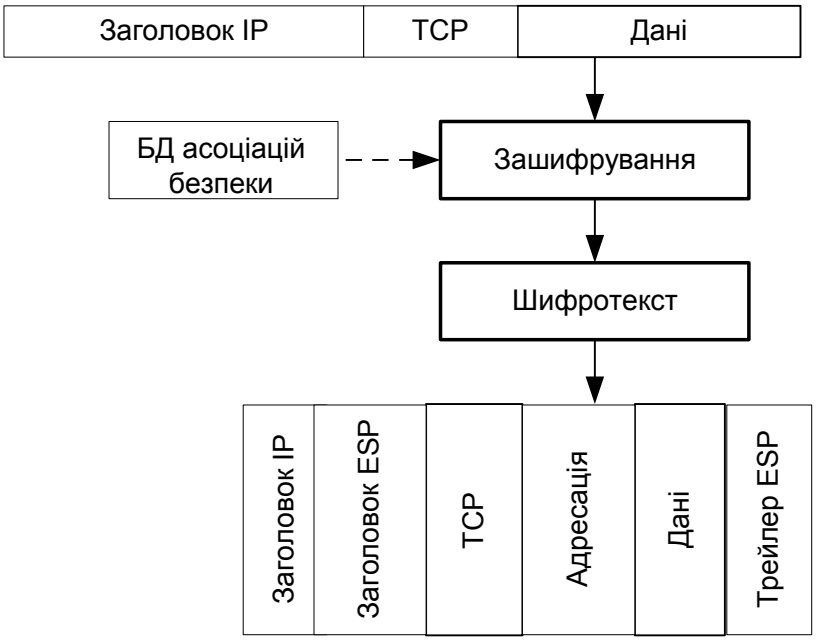


Рис. 4.4. Використання протоколу ESP для оброблення кадрів

Такт роботи для виконання протоколу ESP у транспортному режимі – t_{IPSec} та час оброблення кадру із інформацією про МС T_{IPSec} складають відповідно:

$$t_{IPSec} = t_{SNш}, \tag{4.3}$$

$$Tt_{IPSec} = T_{SNш}. \tag{4.4}$$

де $t_{SNш}$ – такт роботи ОП шифрування із структурою *SNш* визначені в [58], $T_{SNш}$ – час роботи ОП шифрування із структурою *SNш* визначені в [58].

ESP з автентифікуванням використовують для того, щоб забезпечити конфіденційність і автентифікування кадрів із блоками даних $L_{i,j}(c_x)$. У транспортному режимі передавання даних, сервіси автентифікування і конфіденційності застосовуються до даних кадру, не захищаючи при цьому заголовок кадру (рис. 4.5). У цьому режимі після трейлера додається поле даних автентифікування ESP, автентифікується весь шифрований текст і заголовок ESP.

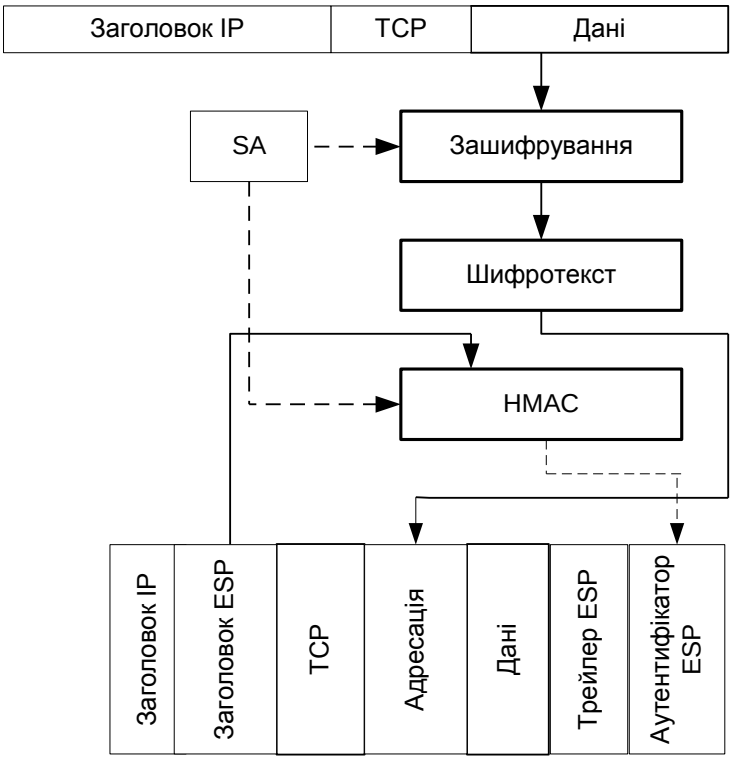


Рис. 4.5. Використання протоколу AH і ESP для оброблення кадрів

Такт роботи ОП для виконання протоколу ESP і AH у транспортному режимі – t_{IPSec} та час оброблення кадру з блоками даних $L_{i,j}(c_x)$ T_{IPSec} складають відповідно:

$$t_{IPSec}=max(t_{SNш}, t_{SNx}, t_{SNx}^{1}), \quad (4.5)$$

$$T_{IPSec}=T_{SNш}+(T_{SNx+}\ T_{SNx}^{1}). \quad (4.6)$$

Проведені дослідження структур ОП виконання алгоритмів хешування і шифрування [20,26], параметрів протоколу IPSec дали змогу побудувати математичну модель структури ОП процесора підтримки протоколу IPSec. Розглянемо взаємозв’язок параметрів ОП виконання протоколу IPSec (рис. 4.6).

Рис. 4.6. Взаємозв’язок параметрів для удосконалення ОП процесора підтримки протоколу IPSec

Вхідними параметрами для розробки структури ОП процесора підтримки протоколу IPSec є:

- криптографічні модулі процесора підтримки протоколу IPSec: алгоритм хешування, алгоритм шифрування;
- сервіси протоколу IPSec: AH, ESP, ESP+AH;
- розміри кадрів із блоками даних $L_{i,j}(c_x)$;
- часові характеристики ОП;

- виконання криптографічних модулів протоколу IPSec.

Шуканими параметрами моделі ОП процесора IPSec є такі параметри структур ОП шифрування і хешування, які при заданих алгоритмах оброблення, сервісах протоколу IPSec, розмірах кадрів і технологічних умов реалізації процесора, приймають найменші значення.

Для побудови моделі ОП пристрою процесора IPSec спочатку визначимо вектори характеристик його складових ОП. Характеристики ОП шифрування опишемо множиною:

$$P_{Ш} = \{\vec{P}_{Ші} \mid \vec{P}_{Ші} = [T_{Ші}, t_{Ші}]\}, \tag{4.7}$$

де $\vec{P}_{Ші}$ – вектор характеристик ОП шифрування *i*-ої структури, $T_{Ші}$ – час оброблення одного кадру, $t_{Ші}$ – такт роботи ОП.

Структура ОП шифрування задається параметрами: $Npp_ш$ - кількість конвеєрних регістрів алгоритму шифрування, $Nksr_ш$ - кількість комбінаційних схем (КС) алгоритму шифрування, $Nr_ш$ - кількість послідовно з'єднаних КС, що реалізують відповідні раунди алгоритму шифрування. Ці параметри разом з технологічними умовами виготовлення ОП визначають часові характеристики цього ОП. До технологічних умов належать часи: $Tk_ш$ – час оброблення даних в комутаторі даних алгоритму шифрування, $Tкс_ш$ – час оброблення даних в КС алгоритму шифрування, $Tрг_ш$ – час запису даних у вихідний регістр алгоритму шифрування.

Виходячи з цього та виразів запропонованих в [26], які описують часові характеристики пристроїв шифрування залежно від структури (ітераційна, конвеєрна, ітераційно-конвеєрна) ОП, представимо $t_{Ші}$ і $T_{Ші}$ у вигляді:

$$t_{Шi}(Npp_ш, Nksr_ш, Nr_ш) = \begin{cases} T_k + T_{KC}*(Ct_1 * Ct_1^{ОШ} + ... + Ct_{Nksr} * Ct_{Nksr}^{ОШ}) + T_{Pг} \\ T_{KC}*(Ct_1^{ОШ} + ... + Ct_{Nr}^{ОШ})/Npp + T_{Pг} \\ T_k + T_{KC}*(Ct_1 * Ct_1^{ОШ} + ... + Ct_{Nksr} * Ct_{Nksr}^{ОШ})/Npp + T_{Pг}, \end{cases} \quad (4.8)$$

$$T_{Шi}(Npp_ш, Nksr_ш, Nr_ш) = \begin{cases} t_{Шi} * (Nr/Nksr) \\ t_{Шi} * Npp \\ t_{Шi} * (Nr/(Nksr*Npp)) \end{cases} \quad (4.9)$$

де Ct_1 – коефіцієнт зростання часу спрацювання *i*-тої КС, що реалізує проекцію раундів алгоритму шифрування; $Ct_1^{ОШ}$ – коефіцієнт зростання часу спрацювання *i*-тої КС при виконанні різних операцій шифрування. Діапазон значень *i* для алгоритму DES складає 15 структур ОП (додаток А).

Характеристики ОП для обчислення HMAC визначаються характеристиками двох ОП хешування з однаковими структурами ОП, які входять до його складу.

Характеристики ОП хешування опишемо множиною:

$$P_X^A = \{\vec{P}_{Xj}^A \mid \vec{P}_{Xj}^A = [T_{Xj}^A, t_{Xj}^A]\}, \quad (4.10)$$

де $\vec{P}_X^A$ – вектор характеристик ОП хешування *j*-ої структури для виконання алгоритму хешування $A=\{A_m \mid A_m = [SHA\text{-}1, MD\text{-}5]\}$, $m=1,2$, T_{Xj}^A – час виконання хешування згідно з алгоритмом *A*, t_{Xi}^A – такт роботи ОП хешування згідно з алгоритмом *A*.

Структура ОП хешування задається алгоритмом *A* та параметрами: Npp_X - кількість конвеєрних регістрів алгоритму хешування, $Nksr_X$ - кількість КС алгоритму хешування, Nr_X - кількість послідовно з'єднаних КС, що реалізують відповідні раунди алгоритму хешування. До технологічних умов належать часи: Tk_X^A– час оброблення даних в комутаторі даних алгоритму хешування *A*, $Tкс_X^A$ - час оброблення даних в КС алгоритму хешування *A*,

$Tрг_X^A$ – час запису даних у вихідний регістр алгоритму хешування *A*, $Tкс_{NrX}^A$ $Tкс_X^A$ - час оброблення даних в *Nr*-тій КС алгоритму хешування *A*.

Виходячи з цього, та виразів, запропонованих в [20], які описують часові характеристики ОП хешування залежно від структури (ітераційна, конвеєрна, ітераційно-конвеєрна) пристрою, представимо t_{XAj} та T_{XAj} у вигляді:

$$t_{Xj}^A(Npp_x,Nksr_x,Nr_x)=\begin{cases} Nksr*T_{KC1}+T_k+T_{Pг} \\ Nksr*T_{KC1}+T_{Pг} \\ Nksr*T_{KC1}+T_k+T_{Pг} \end{cases} \quad (4.11)$$

$$T_{Xj}^A(Npp_x,Nksr_x,Nr_x)=\begin{cases} (Nr-1)*t_{Xj}^A/Nksr+T_{KCNr}+T_{Pг} \\ Npp*t_{Xj}^A+T_{KCNr}+T_{Pг} \\ (Nr-1)*t_{Aj}^A/Npp+T_{KCNr}+T_{Pг} \end{cases} \quad (4.12)$$

Характеристики ОП виконання алгоритму HMAC (A) визначається відповідними часовими характеристиками ОП хешування згідно з цим же алгоритмом. Враховуючи, що перший ОП обробляє повідомлення довільної довжини, а другий лише один буфер, отримано такі вирази:

$$T_{HMAC}^A(Npp_x,Nksr_x,Nr_x)=N*T_{Xj1}^A(Npp_x,Nksr_x,Nr_x)+T_{Xj2}^A(Npp_x,Nksr_x,Nr_x), \quad (4.13)$$

$$t_{HMAC}^A(Npp_x,Nksr_x,Nr_x)=max(t_{Xj1}^A(Npp_x,Nksr_x,Nr_x), t_{Xj2}^A(Npp_x,Nksr_x,Nr_x)). \quad (4.14)$$

де T_{HMAC}^A – час виконання алгоритму HMAC (A); t_{HMAC}^A – такт виконання алгоритму HMAC (A); *N* – кількість буферів для оброблення повідомлення.

Враховуючи, що перший і другий ОП хешування можуть мати різні структури, введено розподілення індексів *j1*, *j2*. За кількістю структур ОП HMAC для SHA-1 є 55, а для MD5 – 28 (додаток А) [116].

На базі виразів для оцінки характеристик ОП хешування, шифрування і обчислення HMAC, побудовано вирази для обчислення характеристик ОП процесора IPSec. При цьому враховано варіанти сервісів протоколу IPSec, які включають сервіси AH, ESP і AH+ESP.

Для сервісу AH з використанням алгоритму хешування *A* і алгоритму шифрування DES отримано такий вираз:

$$T_{IPSec}^{AH,A} = T_{HMAC}^{A}(Npp_x, Nksr_x, Nr_x). \quad (4.15)$$

де $T_{IPSec}^{AH,A}$ - час виконання криптографічних функцій IPSec відповідно до сервісу AH згідно з A.

Для сервісу ESP з використанням алгоритму шифрування DES отримано:

$$T_{IPSec}^{ESP,DES} = T_{Ш}(Npp_ш, Nksr_ш, 16). \quad (4.16)$$

де $T_{IPSec}^{ESP,DES}$ - час виконання криптографічних функцій IPSec згідно сервісу ESP згідно з алгоритмом DES.

Для сервісу AH+ESP при передачі та приймання даних з використанням алгоритму *A* отримаємо:

$$T_{IPSec}^{ESP+AH} = T_{Ш}(Npp_ш, Nksr_ш, 16) + T_{IPSec}^{AH,A} \quad (4.17)$$

де T_{IPSec}^{ESP+AH} - час виконання криптографічних функцій IPSec згідно сервісу ESP та AH згідно з алгоритмом хешування A та шифрування DES.

4.3. Удосконалення програмно-апаратних реалізацій протоколу IPSec

4.3.1. Напрямки розвитку реалізацій протоколу IPSec

Розглянемо детальніше можливі напрямки удосконалення реалізацій протоколу IPSec. Віртуальні приватні мережі під керуванням протоколу IPSec базуються на технологіях, що використовують верхні рівні багаторівневої моделі протоколів розподілених інформаційних систем. Тому При постановці задачі удосконалення програмно-апаратних реалізацій IPSec слід звертати увагу на принцип та структуру роботи протоколів, що впливають на роботу віртуальної приватної мережі.

Відповідно до вище сказаного можна виділити наступні напрямки розвитку реалізацій протоколу IPSec:

- мінімізація вартості складових протоколу IPSec;
- збільшення продуктивності сервера, клієнтів віртуальної приватної мережі, або кабельних комунікацій між ними, з метою підвищення швидкості передачі та обчислень в межах мережної взаємодії;
- мінімізація розміру кадру з інформацією про МС, що опрацьовується протоколом IPSec;
- мінімізація часу виконання перетворень алгоритмами протоколу IPSec.

Перші три напрямки розвитку протоколу IPSec обмежені в засобах досягнення оптимальних рішень внаслідок того, що віртуальні приватні мережі будуються на вже визначеній інфраструктурі. Одним з головних інструментів удосконалення протоколу IPSec є мінімізація часу виконання перетворень алгоритмами протоколу IPSec, зокрема, криптографічних перетворень. Шляхом мінімізації часу виконання криптографічних перетворень можна одержати оптимальні параметри моделі протоколу IPSec. Мінімізації часу виконання криптографічних алгоритмів для процесорів підтримки протоколу IPSec вимагає дослідження структури цих алгоритмів з метою виявлення вузьких місць. У роботі [23] доведено, що вузьким місцем при апаратній реалізації криптографічних модулів є структура ОП. Тому задача ефективного розвитку протоколу IPSec зводиться до удосконалення структури ОП криптографічних модулів для процесорів підтримки протоколу IPSec за критеріями часу опрацювання кадрів з блокими даних $L_{i,j}(c_x)$, затратами обладнання, ефективністю використання обладнання, тощо. Серед перелічених критеріїв виділимо удосконалення за часом опрацювання кадрів. Удосконалення за затратами обладнання розглядати не будемо, оскільки, у [25] доведено, що сучасний розвиток мікросхем дає змогу збільшити затрати обладнання з метою зменшення часу операційної задачі.

Для удосконалення часових параметрів ОП криптографічних модулів для процесорів підтримки протоколу IPSec виділимо в математичній моделі

змінних параметрів складові структури ОП, які в загальному вигляді записуються, як *SN(Nksr,Npp)* (рис. 4.7):

- $SN_{HMAC}^{A}(Nksr,Npp)$ – структури ОП хешування згідно з алгоритмами MD5 та SHA-1,
- *SNш(Nksr,Npp)* – структури ОП шифрування згідно з алгоритмом DES.

Додатково виділимо параметри, які визначають область використання процесора підтримки протоколу IPSec:

- розмір кадру визначається мережними протоколами та задає кількість буферів *N* для опрацювання;
- сервісні параметри протоколу IPSec визначають тип сервісу, який буде обробляти дані. До сервісних параметрів протоколу IPSec віднесемо протоколи AH, ESP та їх комбінацію, транспортний і тунельний режими передачі даних;
- часові параметри криптографічних модулів (час спрацювання регістра, час спрацювання КС, час спрацювання комутатора).

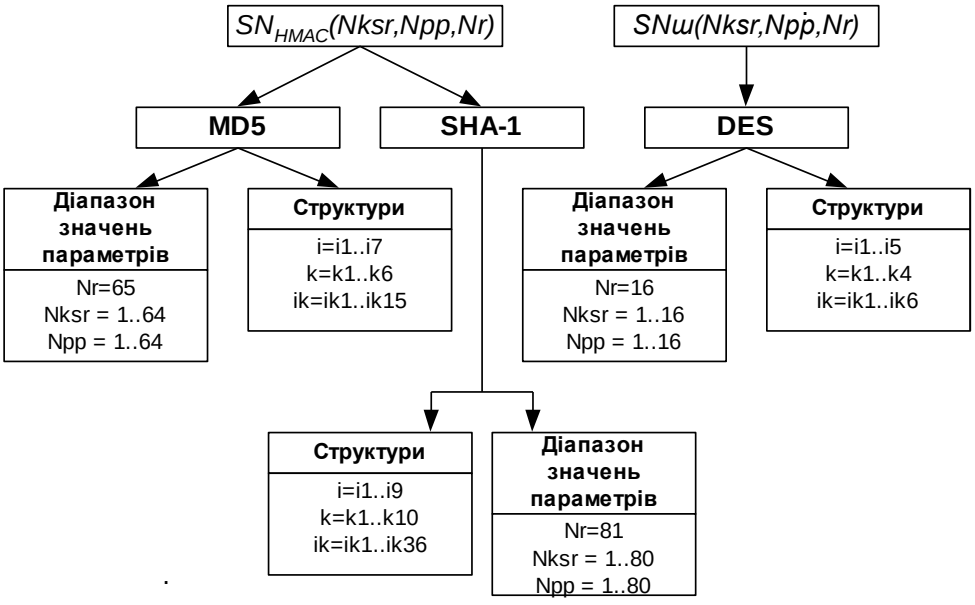


Рис. 4.7. Розподіл множини структур ОП криптографічних модулів для процесорів підтримки протоколу IPSec

4.3.2. Синтез структур операційних пристроїв криптографічних модулів для процесорів підтримки протоколу IPSec

Вхідною інформацією для синтезу структур ОП криптографічних модулів для процесорів підтримки протоколу IPSec є:

- алгоритм шифрування і хешування;
- перелік сервісів, які повинен забезпечувати процесор підтримки протоколу IPSec;
- параметри компонентного базису, в якому буде реалізовано процесор підтримки протоколу IPSec;
- розмір кадрів з блоками даних $L_{i,j}(c_x)$.

Задача проектування ОП криптографічних модулів для процесорів підтримки протоколу IPSec формулюється на основі приведеної інформації:

- вибрати структуру ОП криптографічного модуля, яка забезпечує оброблення заданої кількості буферів даних за мінімальний час;
- синтезувати структуру ОП криптографічного модуля на базі отриманої на попередньому етапі структури;
- побудувати схему ОП криптографічного модуля.

Метод синтезу спеціалізованих комп'ютерних систем складається з двох етапів [33]. На першому етапі виконується представлення криптографічних модулів, реалізованих в протоколі IPSec, у вигляді проекцій конкретизованих потокових графів, що реалізується заданим компонентним базисом. Далі проводиться аналіз отриманих структур за часом оброблення заданого числа кадрів з блоками даних $L_{i,j}(c_x)$ і такту роботи для різних сервісів протоколу. Результатом виконання цього етапу є набір ОП криптографічних модулів з відповідними множинами параметрів (вираз 4.7, 4.10). На другому етапі синтезуються структури ОП криптографічних модулів процесора підтримки протоколу IPSec, оцінюються їх параметри згідно з (4.15 – 4.17) та вибирається така структура, яка забезпечує найменший час оброблення кадрів відповідного розміру.

Опишемо послідовність виконання поставленого завдання, через таку послідовність етапів:

- визначаються сервіси, які повинен реалізувати процесор підтримки протоколу IPSec;
- згідно вибраного сервісу обчислюється час оброблення кадру з блоком даних $L_{i,j}(c_x)$ для кожної структури ОП процесора;
- задається умова порівняння часових характеристик масиву сформованих структур з поточною структурою;
- встановлюються умови визначення найкращих характеристик структур криптографічних модулів;
- виконується пошук параметрів найкращої структури за встановленими умовами та критеріями;
- виконується виведення інформації про характеристики криптографічних модулів апаратно-орієнтованих процесорів підтримки протоколу IPSec.

Блок-схема запропонованого алгоритму наведена в додатку Б. Як видно із блок-схеми, задача удосконалення програмно-апаратних реалізацій протоколу IPSec представляється у вигляді послідовності підзадач. При рішенні кожної із підзадач виконується пошук параметрів ОП за встановленими критеріями. При цьому на кожному наступному етапі коло пошуків звужується, що забезпечує зменшення часу перебору параметрів для найкращих характеристик структур ОП криптографічних модулів апаратно-орієнтованих процесорів підтримки протоколу IPSec для різних сервісів оброблення даних за різних технологічних характеристик компонентного базису.

4.3.3 Програмне забезпечення для вибору параметрів структури операційного пристрою криптографічних алгоритмів для процесорів підтримки протоколу IPSec

На основі запропонованого в п. 4.3.2 алгоритму та математичної моделі ОП криптографічних алгоритмів для процесорів підтримки протоколу IPSec розроблено програмне забезпечення для знаходження набору параметрів структури ОП з обчисленням її продуктивності.

Для роботи програмного забезпечення необхідно ввести наступні параметри математичної моделі:

- криптографічні алгоритми (алгоритми хешування: MD-5, SHA-1 та алгоритм шифрування: DES);
- технологічні характеристики реалізації: $Tk_{ш}$, $Tкс_{ш}$, $Tрг_{ш}$, Tk_{x}^{A}, $Tкс_{x}^{A}$, $Tрг_{x}^{A}$, $Tкс_{Nrx}^{A}$;
- сервіси протоколу IPSec: AH, ESP, комбінація AH та ESP в режимі прийому та передачі даних;
- розмір кадру з блоком даних $L_{i,j}(c_x)$.

Текст програми наведено в додатку В, а блок-схема алгоритму роботи програми в додатку Г.

На першому етапі при запуску програми з’являється діалогове вікно між комп’ютером та користувачем (рис. 4.8).

Для виконання даної програми існують функціональні кнопки:

- очистити – очищує область даних та оптимізації;
- обчислити – визначає максимальну структуру ОП хешування чи шифрування (залежно від вхідних даних);
- графік – зображає в окремому вікні графічну залежність між структурами ОП криптографічних алгоритмів процесора підтримки протоколу IPSec;
- вихід – вихід з програми.

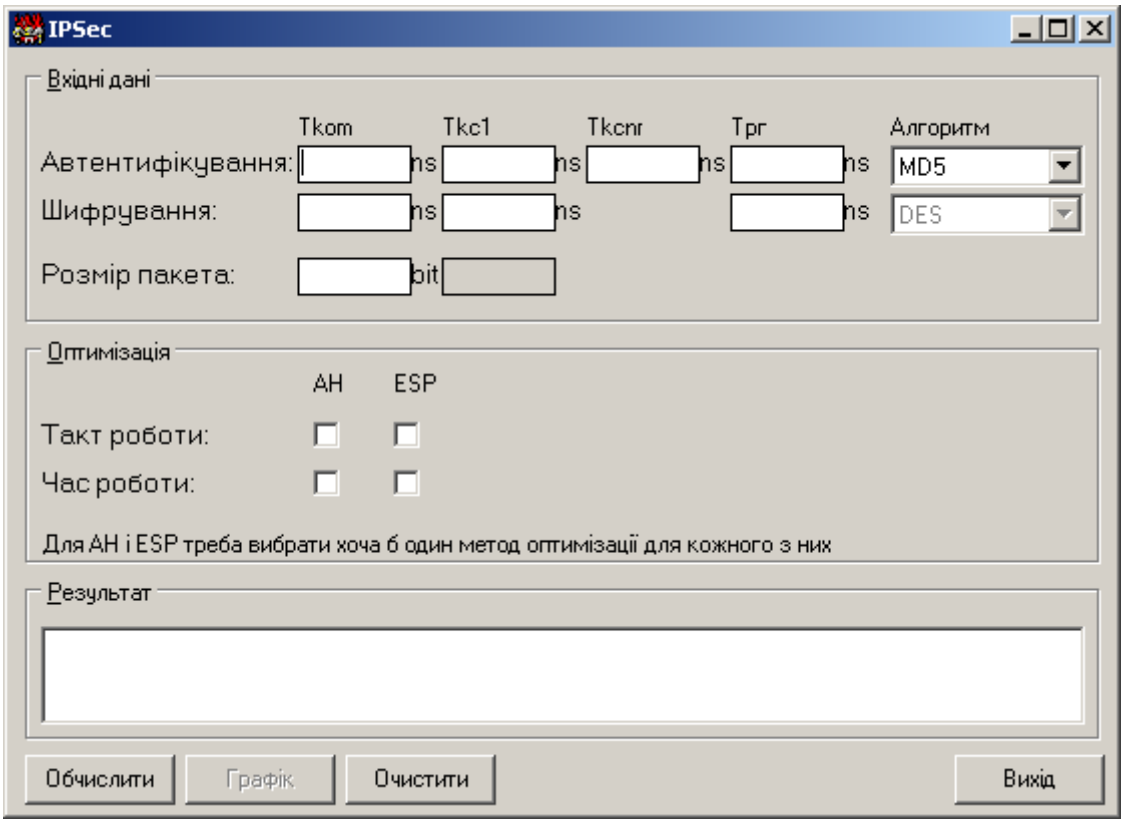


Рис. 4.8. Головне вікно розробленого програмного забезпечення

Вікно розробленого програмного забезпечення (рис. 4.8) містить три області:

1) Область для введення вхідних даних, в яку включено такі поля:
   - T_{kom} – час спрацювання комутатора шифрування/хешування;
   - T_{kc1} – час спрацювання КС 1 хешування/шифрування;
   - T_{kcnr} – час спрацювання КС *Nr*, яка здійснює додавання за модулем 2^{32} (використовується тільки для алгоритму хешування);
   - T_{pr} – час спрацювання регістра;
   - алгоритм задає алгоритм хешування;
   - розмір задає бітове значення розміру кадру з інформацією про МС.
2) Область оптимізації дає змогу встановити параметри для визначення удосконалених характеристик структур ОП. Критерієм вибору параметрів ОП є час або такт роботи для сервісів AH чи ESP або ж їхньої комбінації.

3) Область видачі результатів – на основі аналітичних формул для визначення часових операційних характеристик алгоритмів протоколу IPSec виконує оброблення інформації з попередніх областей.

Результатом роботи програмного забезпечення є набір параметрів пристроїв хешування і шифрування та їх часові характеристики (рис. 4.9).

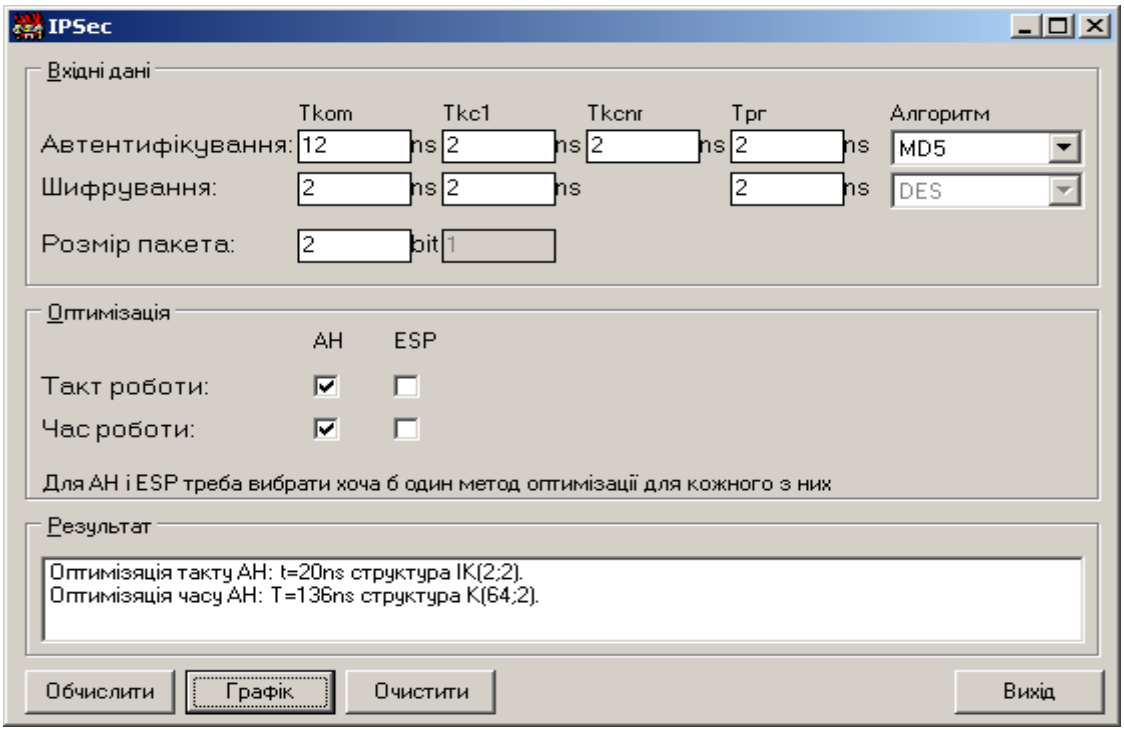


Рис. 4.9 – Результати визначення такту та часу роботи протоколу AH

На основі проведених обчислень будуються графіки залежності часу оброблення кадрів з інформацією про МС комп'ютерних мереж, від структур ОП шифрування, хешування та їх комбінації (рис. 4.10).

Для ідентифікації структур ОП криптографічних алгоритмів для процесорів підтримки протоколу IPSec на рис. 4.10 використано умовне позначення, запропоноване в [20]: *SN*(*Nksr*, *Npp*), де *SN* – код назви структури ОП ("і" – ітераційний граф-алгоритмічний ОП, "к" – конвеєрний граф-алгоритмічний ОП, "ік" – ітераційно-конвеєрний граф-алгоритмічний ОП),

Nksr, *Npp* – параметри структур ОП: кількість реалізованих КС і конвеєрних регістрів відповідно.

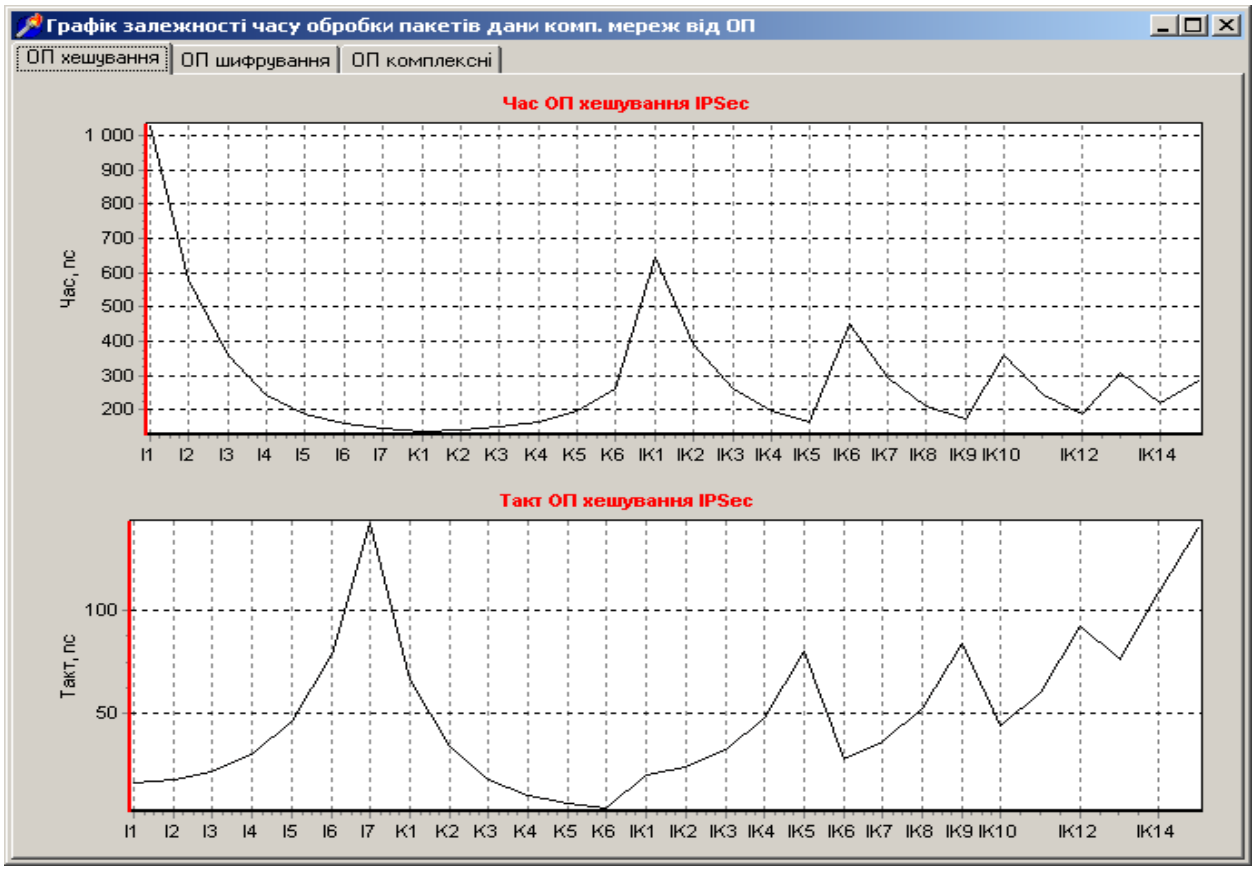


Рис. 4.10. Графік залежності часу оброблення даних комп'ютерної мережі від структур ОП базових криптографічних алгоритмів протоколу IPSec

Аналіз графіків показує, що вибір структури ОП криптографічних модулів для процесорів підтримки протоколу IPSec залежить від розміру та структури кадрів з блоками даних $L_{i,j}(c_x)$ та сервісів протоколу IPSec. Встановлено, що для забезпечення сервісів протоколу AH доцільно використовувати ітераційні структури ОП, для забезпечення сервісів протоколу ESP – ітераційно-конвеєрні. Для комбінованого протоколу шифрування/автентифікування та ESP – ітераційно-конвеєрні, та ітераційні - для AH.

4.3.4. Удосконалені характеристики структур операційного пристрою криптографічних модулів протоколу IPSec

Відомо, що затримку та джітер можна зменшити за рахунок зменшення довжини кадру з інформацією про МС. У цьому випадку кадри великої довжини не затримують відправку інших кадрів. Цього ефекту можна досягти засобами сегментації всіх вихідних кадрів, які перевищують відповідну довжину на пристроях генерування кадрів. Враховуючи сучасні тенденції розвитку комп'ютерних систем та їх компонентів, у яких довжина кадру пов'язана із швидкістю його поступлення [87], та використовуючи розроблене програмне забезпечення для пошуку оптимальних параметрів структур ОП криптографічних модулів для процесорів підтримки протоколу IPSec, удосконалено характеристики структур ОП криптографічних алгоритмів для кадрів різного розміру при різних сервісах протоколу IPSec (таблиця 4.1).

Таблиця 4.1

Удосконалені характеристики структур ОП криптографічних модулів процесорів підтримки протоколу IPSec

Розмір кадру, біт	Швидкість поступлення кадрів, Кбіт/с	Сервіси протоколу IPSec				
		AH (MD5)	AH (SHA-1)	ESP (DES)	AH+ESP (MD5,DES)	AH+ESP (SHA-1,DES)
256	64	i(64;1)	i(80;1)	ik(2;8)	i(64;1), ik(8;2)	i(80;1), ik(8;2)
512	128	i(64;1)	i(80;1)	ik(8;8)	i(64;1), ik(8;2)	i(64;1), ik(8;2)
2048	512	i(64;1)	i(80;1)	ik(2;8)	i(64;1), ik(8;2)	i(64;1), ik(8;2)
6144	1576	i(64;1)	i(80;1)	ik(8;8)	i(64;1), ik(16;8)	i(64;1), ik(16;2)

При побудові табл. 4.1 використано результатати синтезу ОП хешування [20] та ОП пристроїв шифрування [26] на програмовану логічну інтегральну схему типу ALTERA EPF10K50-3, які дали змогу визначити

технологічні характеристики для знаходження параметрів структур ОП криптографічних модулів процесора підтримки протоколу IPSec. Технологічні характеристики елементів ОП наступні: $Tk_{ш}=1,2$ *нс,* $Tкс_{ш}=14,3$ *нс,* $Tрг_{ш}=0,9$ *нс,* $Tk_{x}=0,7$ *нс,* $Tкс_{x}=9$ *нс,* $Tрг_{x}=0,3$ *нс,* $Tкс_{Nrx}=2,1$ *нс*.

На рис. 4.11 подано графік залежності часу оброблення кадрів різного розміру від кількості операцій сервісів AH, ESP та AH+ESP протоколу IPSec.

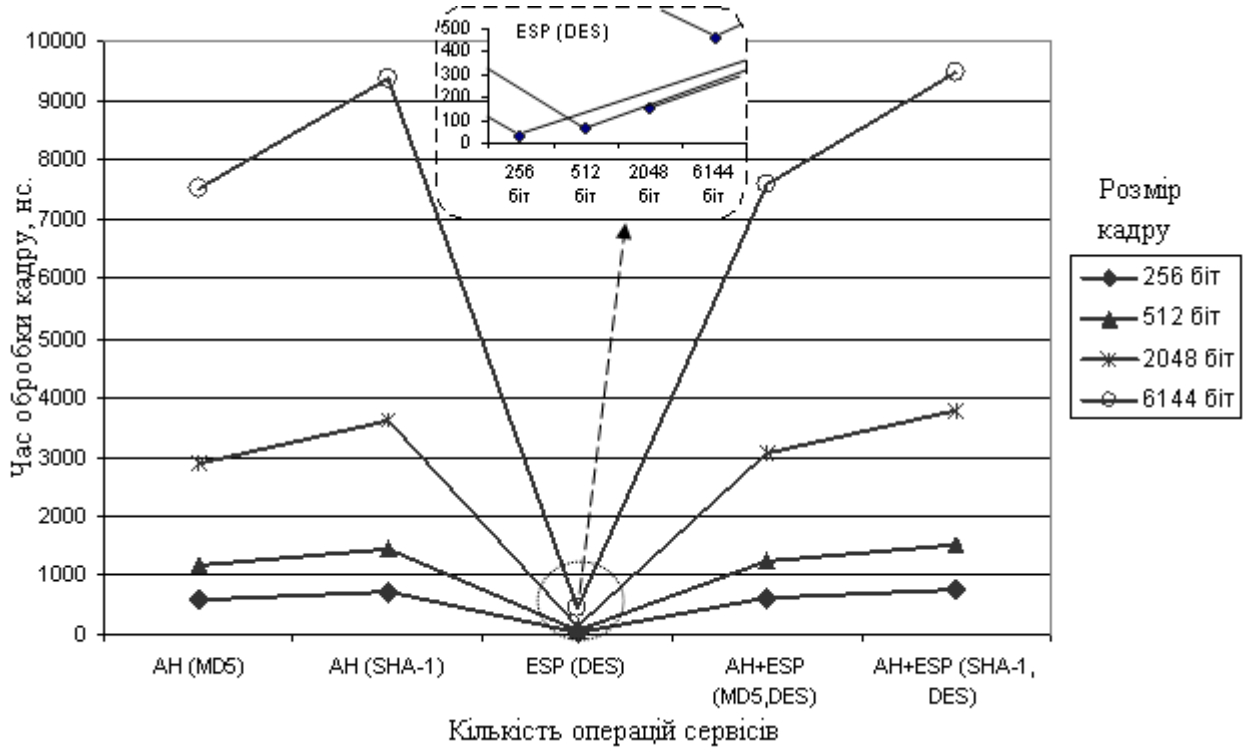


Рис. 4.11 – Графік залежності часу оброблення кадрів різного розміру від кількості операцій сервісів протоколу IPSec

Аналіз графіку дозволяє зробити висновок, що найменша тривалість опрацювання кадрів спостерігається при використанні сервісу ESP відповідно до алгоритму DES. Із збільшенням розміру кадру значення часу його опрацювання зростає прямопропорційно. Найбільший час оброблення кадрів спостерігається при автентифікуванні, відповідно до алгоритму SHA-1 та суміщенні сервісів протоколу IPSec (AH+ESP).

На рис. 4.12 наведено комплексну структуру ОП криптографічних алгоритмів для процесорів підтримки протоколу IPSec (К1ІК3).

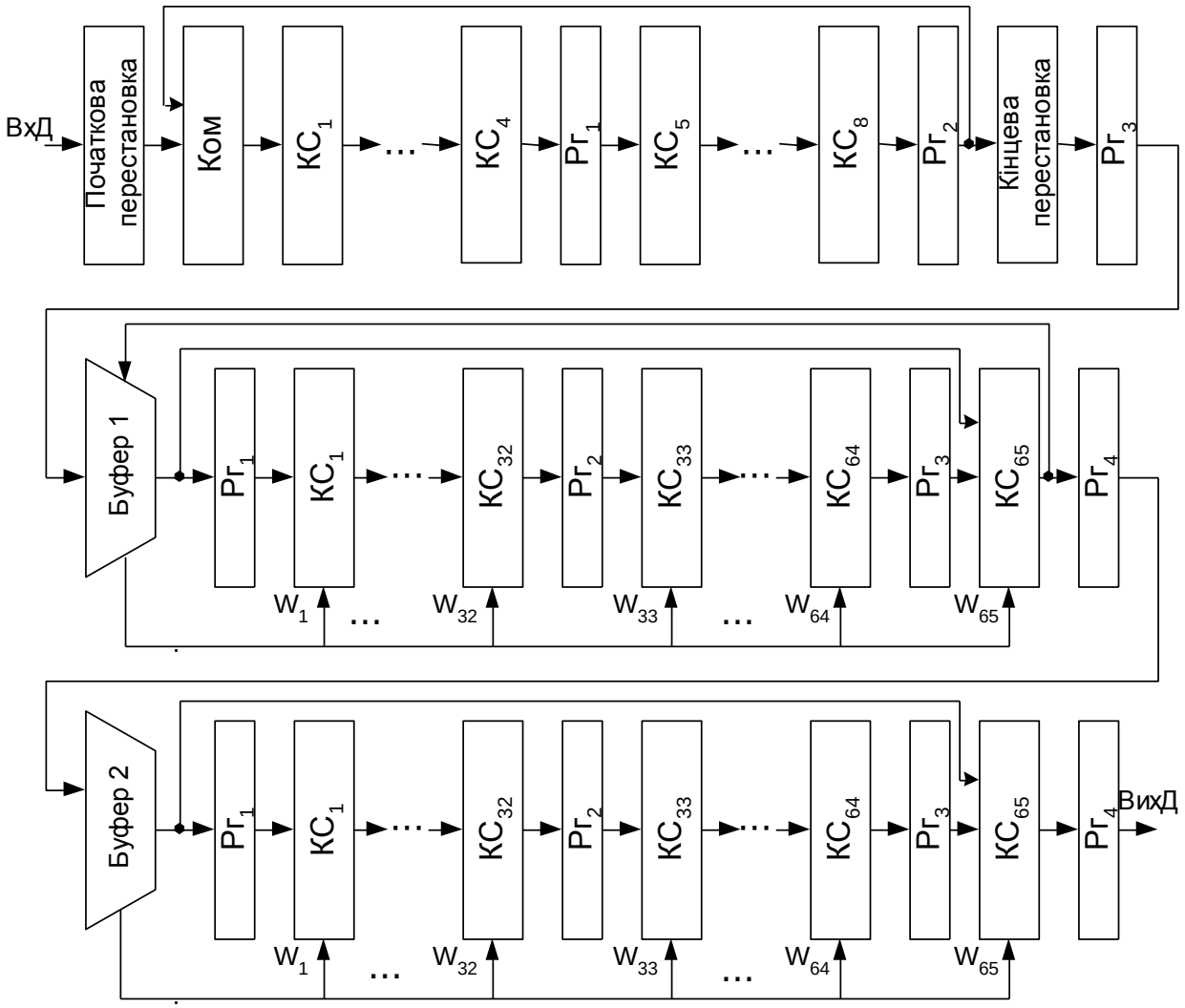


Рис. 4.12. Комплексна структура ОП криптографічних алгоритмів для процесорів підтримки протоколу IPSec (К1ІК3)

На рисунку зображено структури: К1(Nksr,Npp)=К1(64,2) – структура хешування, ІК3(Nksr,Npp)=ІК3(8,2) – структура шифрування.

Аналіз отриманих результатів дозволив встановити, що в більшості випадків найменший час опрацювання кадрів досягається при ітераційній та ітераційно-конвеєрній реалізації криптографічних модулів апаратно-орієнтованих процесорів підтримки протоколу IPSec.

ВИСНОВКИ

1. Досліджено базові структури ОП криптографічних модулів процесорів підтримки протоколу IPSec, що дало змогу побудувати аналітичні вирази, які описують час оброблення кадрів із блоками даних $L_{i,j}(c_x)$, залежно від параметрів структури ОП. На основі побудованих аналітичних виразів запропоновано математичну модель ОП процесора підтримки протоколу IPSec, параметром якої є значення відображення потокових графів базових криптографічних алгоритмів. Аргументами математичної моделі ОП є алгоритми оброблення кадрів із блоками даних $L_{i,j}(c_x)$, технологічні характеристики компонентного базису реалізації та перелік сервісів протоколу IPSec.
2. Розроблено програмне забезпечення для пошуку оптимальних параметрів структур ОП криптографічних модулів процесорів підтримки протоколу IPSec, що дало змогу удосконалити характеристики структур ОП для різних сервісів оброблення даних за різних технологічних характеристик компонентного базису.
3. Доведено, що найменша тривалість часу оброблення кадрів з блоками даних $L_{i,j}(c_x)$ спостерігається при використанні сервісу ESP відповідно до алгоритму DES. Із збільшенням розміру кадру значення часу його опрацювання зростає прямопропорційно. Найбільший час оброблення кадрів спостерігається при автентифікуванні, відповідно до алгоритму SHA-1 та суміщенні сервісів протоколу IPSec (AH+ESP).
4. Обґрунтовано, що в більшості випадків найменший час опрацювання кадрів досягається при ітераційній та ітераційно-конвеєрній реалізації криптографічних модулів апаратно-орієнтованих процесорів підтримки протоколу IPSec.

РОЗДІЛ 5
РЕАЛІЗАЦІЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ БАГАТОКАНАЛЬНИХ ТРАНСКОДЕРІВ СТИСНЕНИХ МОВНИХ СИГНАЛІВ

5.1. Програмне забезпечення транскодування стиснених мовних та звукових сигналів

Програмне забезпечення транскодування стиснених мовних та звукових сигналів працює на універсальних процесорах та використовуються лише у системах, де показник реального часу не є важливим. Мінімальний набір функцій, що мають бути реалізовані в програмних транскодерах стиснених МС:

- ідентифікація вхідних форматів;
- перетворення форматів;
- збереження вихідних файлів.

Середовищем розробки програмного транскодера стиснених мовних та звукових сигналів обрано візуальну мову програмування Visual C++ з пакету Microsoft Visual Studio 6.0. Visual C++ 6.0 представляє собою потужний та складний інструмент для створення прикладних програм з графічним інтерфейсом для операційних систем типу Windows [45,47,56].

На рис. 5.1 наведена узагальнена структура спроектованого програмного забезпечення.

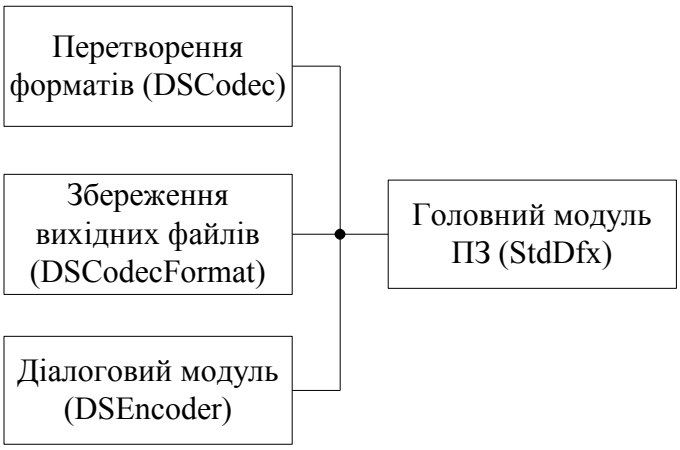


Рис. 5.1. Структура програмного забезпечення транскодування стиснених мовних та звукових сигналів

У модулі DSCodecFormat реалізовано функцію зберігання вихідних файлів. Якщо на диску на який проводиться запис файлу достатньо місця, то керування транскодуванням передається модулю “Збереження вихідних файлів”. Блок-схема алгоритму роботи даного модуля наведена на рис. 5.2.

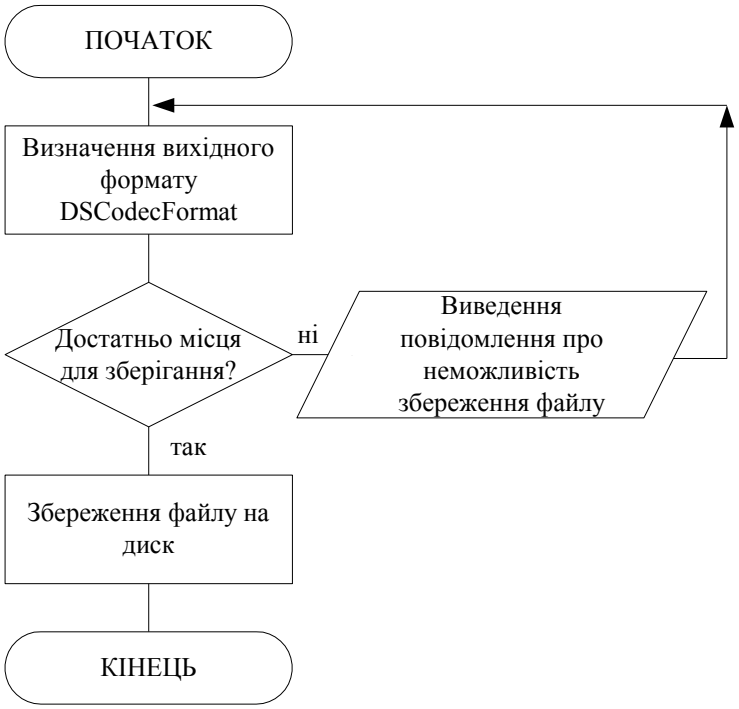


Рис. 5.2. Блок-схема алгоритму роботи модуля “Зберігання вихідних файлів”

У модулі DSCodec, методами об’єктно-орієнтованого програмування за допомогою конструктора CDSCodec::CDSCodec() із використанням стандартної функції BuildCodecFormat, реалізована функція перетворення форматів стиснених мовних сигналів. Для реалізації даної функції підключено бібліотеку STL, яка містить набір функцій для роботи із стрічками, математичними виразами та для перетворення форматів файлів. Алгоритм роботи даного модуля виконує перевірку вхідної інформації, і якщо та перетворює стисненні мовні та звукові сигнали, що містяться у файлах. Блок-схема алгоритму роботи модуля “Перетворення форматів” наведена на рис. 5.3.

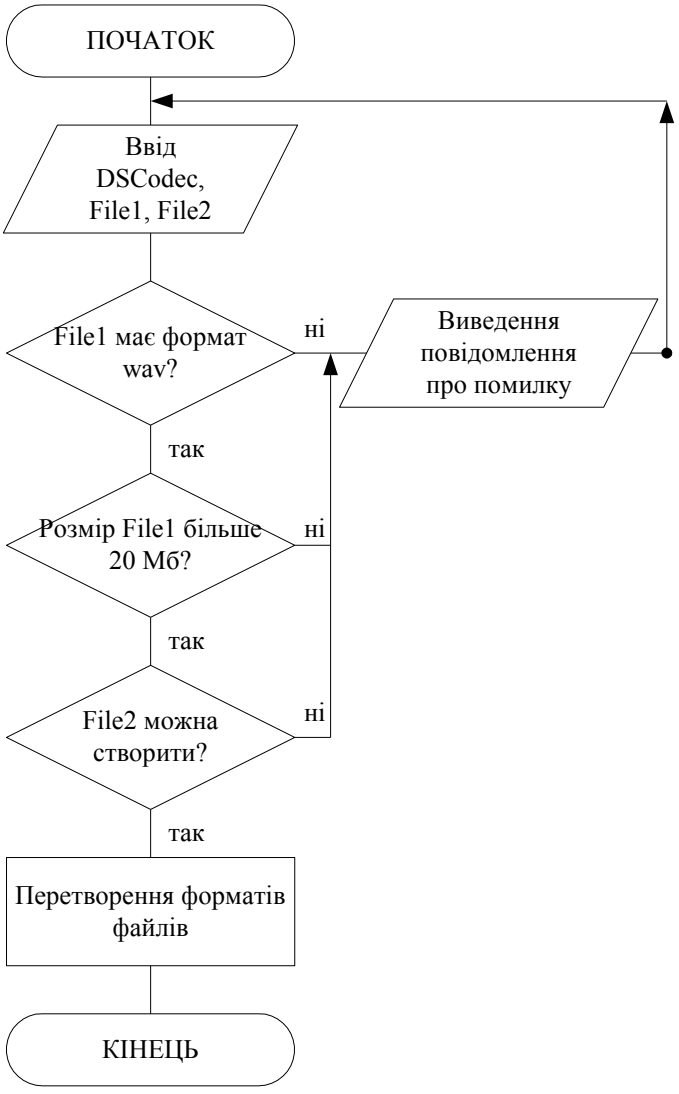


Рис. 5.3. Блок-схема алгоритму роботи модуля “Перетворення форматів”

На рис. 5.4 наведено головне вікно розробленого ПЗ.

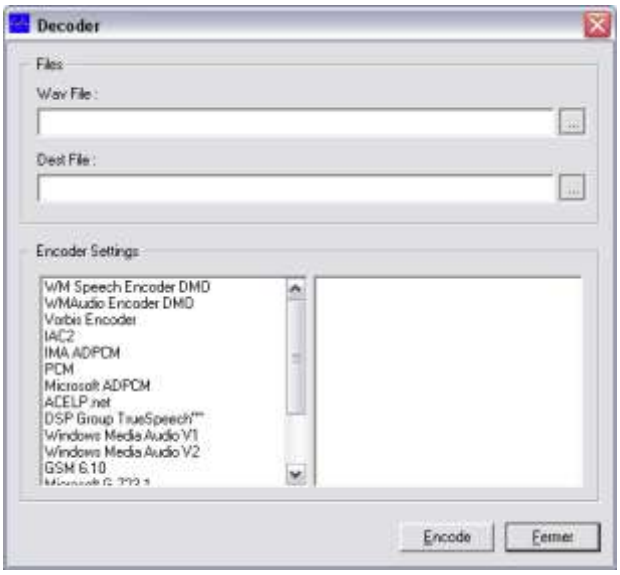


Рис. 5.4. Головне вікно ПЗ

Розроблене ПЗ працює з такими кодеками: WMA Voice Encoder DMO, WM Speech Encoder DMO, WM Audio Encoder DMO, 3ivx D4 Audio Encoder, Indeo Audio Software, Pinnacle AC3 Encoder, Pinnacle AC3 Encoder, Pinnacle MP3 Encoder, Pinnacle MPEG Layer-2 Audio Encoder, Vorbis Encoder, IMC, IAC2, IMA ADPCM, PCM, Ogg Vorbis, Microsoft ADPCM, ACELP.net, DSP Group TrueSpeech, Windows Media Audio, GSM 06.10, G.723.1, CCITT A-Law, CCITT u-Law, AC-3 ACM Codec, MPEG Layer-3. Частоти дискретизації використаних кодеків змінюються від 8 до 96 КГц, швидкість від 0,1 до 768 Кб/с, кількість каналів від 1 до 5. Розмір ПЗ - 32 Кб. Розроблене ПЗ функціонує під керуванням операційних систем типу Windows.

5.2. Синтез алгоритмів стиснення мовних сигналів

5.2.1. Цифрові процесори оброблення сигналів

Вузькосмуговому стисненню МС на ринок комерційних додатків відкрив розвиток мікроелектроніки і, зокрема, поява дешевих процесорів цифрового оброблення сигналів (ЦОС) в інтегральному виконанні [14,15,51].

Процесори ЦОС мають архітектуру, оптимізовану для виконання операцій, які характерні для типових алгоритмів оброблення сигналів. Прикладами таких операцій є множення з накопиченням, а також вибірка операндів з біт-інверсною адресацією, необхідна для виконання швидкого перетворення Фур'є [1,32,33].

Архітектура процесорів ЦОС містить декілька обчислювальних блоків, які одночасно забезпечують виконання операції в одному такті роботи процесора. Для завантаження обчислювальних блоків інформацією використовується декілька шин передачі даних та багатопортова пам'ять даних. Для збільшення продуктивності процесорів ЦОС пам'ять інструкцій та пам'ять даних розподілені, а доступ до них здійснюється по різних шинах. Для процесорів ЦОС характерним є використання інструкцій, що містять поля для керування всіма обчислювальними блоками.

Процесори ЦОС виготовляють у вигляді інтегральних мікросхем, які в одному кристалі містять ядро процесора, пам'ять і периферійні пристрої для обміну інформацією. Наявність вбудованої пам'яті забезпечує швидкий доступ ядра до її вмісту для отримання максимальної продуктивності [140].

Існує безліч модифікацій процесорів ЦОС, що різняться продуктивністю, обсягом пам'яті, споживаною потужністю, тощо. У обладнанні мультисервісних мереж зв'язку використовуються дешеві процесори з середньою продуктивністю і малою споживаною потужністю, орієнтовані на реалізацію одноканального оброблення інформації у складі термінальних пристроїв, або потужні високопродуктивні процесори, орієнтовані на багатоканальні (десятки каналів) додатки, що використовуються у складі таких групових пристроїв, як багатоканальні шлюзи IP-телефонії чи контролери багатоточкових конференцій підключені до комунікаційних мереж по цифрових трактах E1 [6,18,41,67,99].

Найвідоміших виробниками ЦОС є фірми Motorola (www.motorola.com), Texas Instruments (www.ti.com), Analog Devices (www.analog.com).

Система Code Composer Studio розроблена спеціально для проектування процесорів ЦОС фірми Texas Instrument [147]. Code Composer Studio об'єднує в собі можливості редактора, відлагоджувача, компонувальника та компілятора, а також дає змогу в масштабі реального часу проводити аналіз даних та виконувати їх візуальний перегляд.

Для реалізації алгоритмів стиснення МС із процесорів типу TMS320C6000 обрано процесор TMS320C6201-200 [18,67,99,146]. На тактовій частоті 200 Мгц мікропроцесор має продуктивність до 1,6 млрд операцій в секунду.

Структура мікропроцесора TMS320C6201 наведена на рис. 5.5.

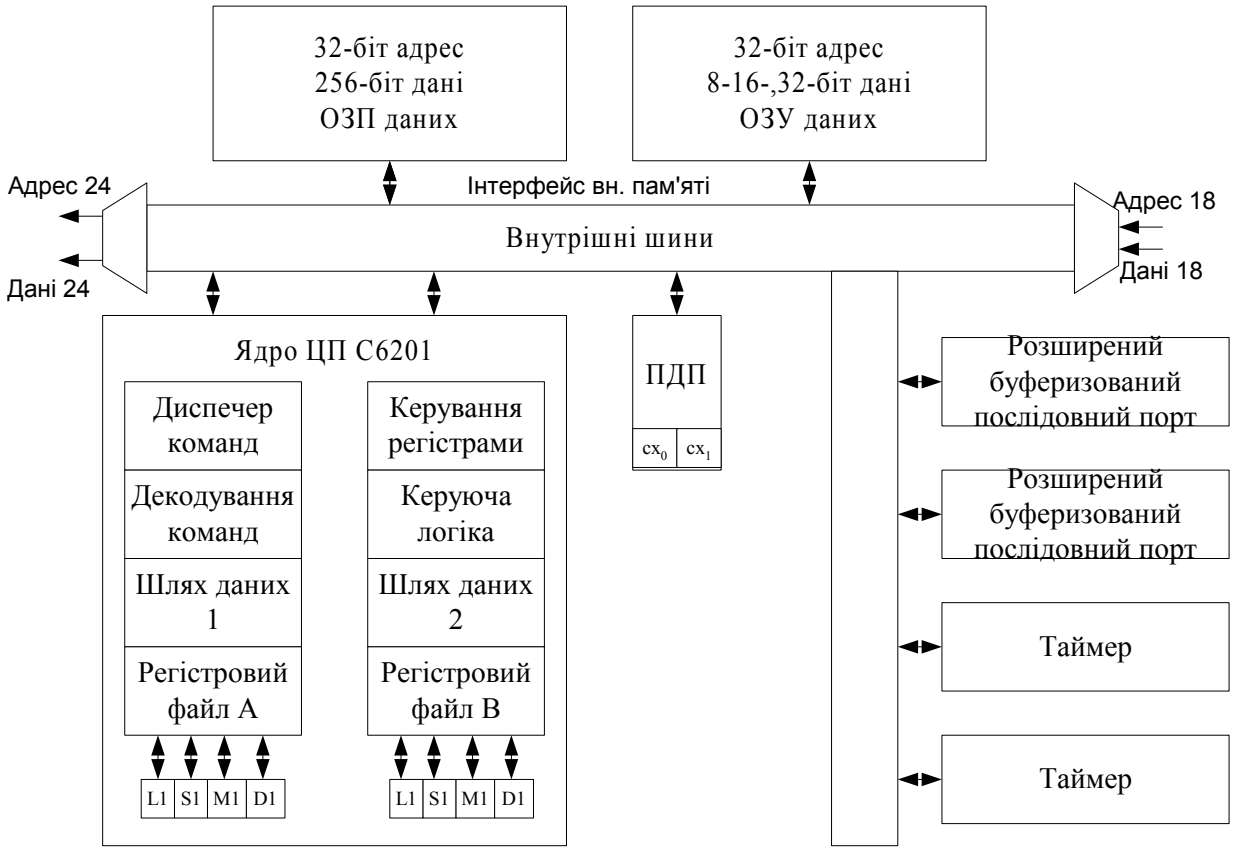


Рис. 5.5. Структура мікропроцесора TMS320C6201

Процесор TMS320C6201 складається з трьох основних частин: центрального процесора (ядро), периферійних пристроїв і пам'яті.

5.2.2. Реалізація алгоритму багатоімпульсного квантування з максимальною достовірністю

Алгоритм БКМД використовується в мультимедійному обладнанні, що працює відповідно до стандарту Н.324 [148]. Алгоритм працює на швидкостях 5.3 і 6.3 Кбіт/с. Висока швидкість передачі забезпечує кращу якість мови. Нижча швидкість забезпечує хорошу якість мови і надає розробникам додаткові можливості при побудові систем. У будь-який момент на межі кадру можливе перемикання швидкості передачі.

Вхідний МС з частотою дискретизації 8 КГц розбивається на кадри завдовжки 30мс, що відповідає 240 16-бітовим відлікам в лінійному законі. Додатково існує затримка (look ahead), яка складає 7.5мс, що визначає сумарну алгоритмічну затримку рівну 37.5мс. Додаткові затримки виникають з наступних причин:

- процеси стиснення та декомпресії вимагають деякого часу;
- час передачі по каналу;
- затримка мультиплексування при комбінуванні МС з іншими видами даних.

Схема кодера алгоритму БКМД наведена на рис. 3.7 [74,95,118]. Кожен кадр, що надходить на вхід кодера подається на фільтр верхніх частот для видалення постійної складової, а потім ділиться на 4 підкадри. Для кожного підкадру обчислюються параметри фільтру лінійного прогнозування 10-го порядку. Для останнього підкадру ці параметри квантуються з використанням Predictive Split Vector Quantizer (PSVQ). Для передачі декодеру виконується перетворення КЛП у вектор ЛСП і його подальше квантування. Неквантовані КЛП використовуються для побудови короткочастотного фільтру на який подається МС. Для двох підкадрів по схемі з розімкненою петлею обчислюється ВТ, яка знаходиться в діапазоні від 18 до 142 відліків [95].

Подальше опрацювання МС відбувається по підкадрах. Ґрунтуючись на раніше обчисленій оцінці ВТ, будується фільтр гармонічного шуму. Для отримання імпульсного відгуку використовується комбінований фільтр, що складається з синтезуючого фільтру КЛП, формантного зважуючого фільтру і фільтру гармонічного шуму. На підставі оцінки ВТ і імпульсного відгуку обчислюється спрогнозоване значення ВТ 5-го порядку в схемі із замкнутою петлею. Диференціал обчислюється у інтервалі отриманої раніше оцінки ВТ. Значення прогнозувальника ВТ віднімається від первинного цільового вектора. Значення ВТ і диференціал передаються від кодера до декодера. Потім апроксимується неперіодична компонента збудження. Для більшої швидкості використовується збудження, отримане по схемі MP-MLQ, а для меншої швидкості - по схемі ACELP [95,118].

Схема декодера алгоритму наведена на рис. 3.3 [74,95]. Робота декодера побудована на покадровому принципі. Спочатку декодуються індекси квантованих КЛП, потім будується синтезуючий фільтр КЛП. Для кожного підкадру декодується збудження АКК і ФКК та подається на синтезуючий фільтр. Адаптивний постфільтр складається з формантного постфільтру і реверсивного (forward-backward) постфільтру ВТ. Сигнал збудження передається на постфільтр ВТ, потім на синтезуючий фільтр, а вихід синтезуючого фільтру подається на вхід формантного постфільтру. Блок масштабування підсилення зберігає рівень енергії на вході формантного постфільтру. У декодері також існує механізм відновлення втрачених кадрів, який вмикається у разі неспівпадання контрольного біта. Відновлення виконується на базі останнього отриманого кадру і збереженому контексті декодера [95].

Розроблено програмне забезпечення реалізації алгоритму для процесорів типу TMS320C6201 фірми Texas Instruments. Програмна реалізація задовольняє наступні вимоги:

- повністю сумісна з стандартом ITU-T G.723.1 (досягнута побітова відповідність тестовим векторам);

- функціонує в режимі реального часу;
- сумісна зі всіма цифровими процесорами типу TMS320C6201.

В таблиці 5.1 наведено результати аналізу апаратної складності реалізації кодеків алгоритму БКМД.

Таблиця 5.1

Апаранта складність кодеків алгоритму БКМД

	G.723.1 6.3кбіт/с кодер	G.723.1 6.3кбіт/с декодер	G.723.1 5.3кбіт/с кодер	G.723.1 5.3кбіт/с декодер
Обчислювальний ресурс, МГц	19.70	1.63	14.5	1.55
Пам’ять				
програм, К байт	67.72			
таблиці, К байт	37.6			
даних, К байт	4.24 + 1.86* N			
Всього, К байт	109.56 + 1.86 * N			

Примітка: N – кількість одночасно реалізованих каналів

На рис.5.6 проілюстровано використання обчислювального ресурсу кодеком алгоритму БКМД.

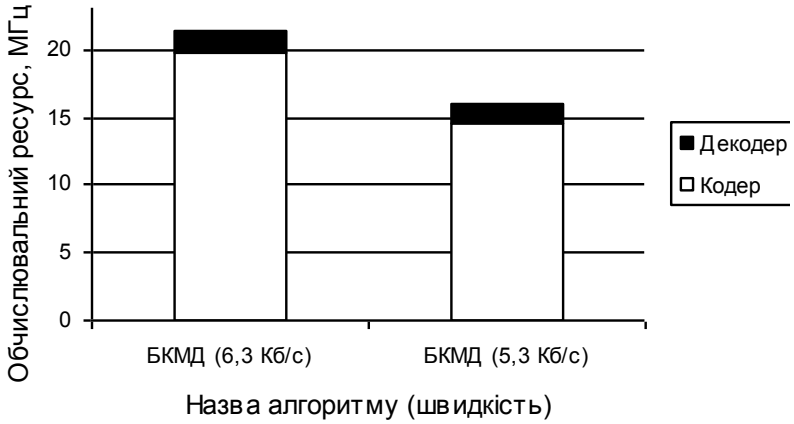


Рис. 5.6 – Частота кодеку алгоритму БКМД

На рис. 5.7 проілюстровано використання пам’яті восьми канальним кодеком алгоритму БКМД.

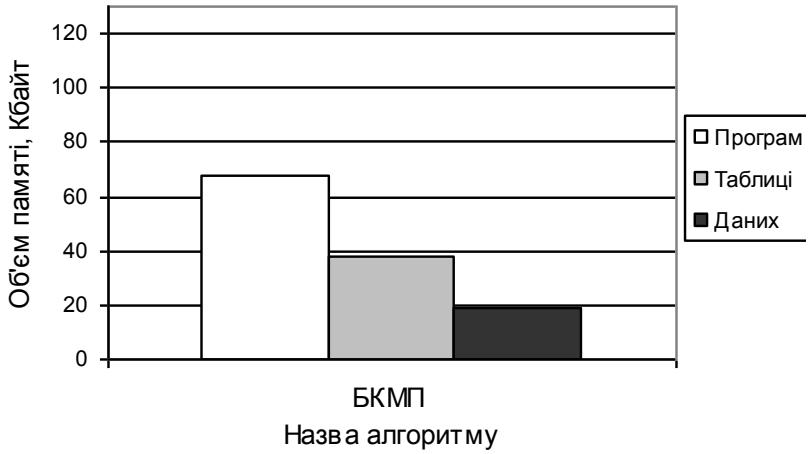


Рис.5.7 – Використання пам’яті кодеком алгоритму БКМП

Всі виміри проводились на TMS320C6201 EVM з використанням Code Composer Studio 1.0 для восьми канального кодека [146,148].

5.2.3. Реалізація алгоритму лінійного прогнозування, що генерується алгебраїчним кодом спряженої структури

Алгоритм ЛПАКСС базується на моделі кодування з використанням лінійного прогнозувальника із збудженням по кодовій алгебраїчній книзі (CELP-модель). Кодер оперує кадрами МС завдовжки 10мс, дискретизованими з частотою 8КГц, що відповідає 80 16-бітовим відлікам в лінійному законі ІКМ. Для кожного кадру проводиться аналіз МС і виділяються параметри моделі (КЛП, індекси і коефіцієнти підсилення в АКК і ФКК). Далі ці параметри кодуються і передаються в канал. Схема кодера алгоритму наведена на рис. 3.9 [96,131].

У декодері даны використовується для відновлення параметрів сигналу збудження і коефіцієнтів синтезуючого фільтру. МС відновлюється шляхом подачі сигналу збудження на короткочастотний синтезуючий фільтр. Синтезуючий фільтр має полюсну передавальну функцію 10-го порядку. Для роботи синтезатора ВТ використовується АКК. У подальшому, якість мови покращується адаптивним постфільтром. У разі втрати кодером потоку даних

під час передачі, початкові дані для мовного синтезатора отримується через інтерполяцію даних з попередніх "хороших" кадрів, але при цьому енергія інтерпольованого МС поступово зменшується, що не створює особливого дискомфорту у слухача. Схема декодера алгоритму наведена на рис. 3.5 [96,131].

Розроблене програмне забезпечення реалізації для процесорів типу TMS320C6201 фірми Texas Instruments. Програмна реалізації задовольняє наступні вимоги:

- повністю сумісна з стандартом ITU-T G.729A (досягнута побітова відповідність тестовим векторам);
- функціонує в режимі реального часу;
- сумісна зі всіма цифровими процесорами типу TMS320C6201.

В таблиці 5.2 наведено результати аналізу апаратної складності реалізації кодеків алгоритму ЛПАКСС.

Таблиця 5.2

Апаратна складність кодеків алгоритму ЛПАКСС

	G.729A кодер	G.729A декодер
Обчислювальний ресурс, МГц	10.72	2.24
	12.96	
Пам’ять		
програм, К байт	6.52	
таблиці, К байт	3.14+0.05* N	
даних, К байт	1.17 + 1.56* N	
Всього, К байт	10.83 + 1.61 * N	

Примітка: N – кількість одночасно реалізованих каналів

На рис. 5.8 проілюстровано використання обчислювального ресурсу кодеком алгоритму ЛПАКСС.

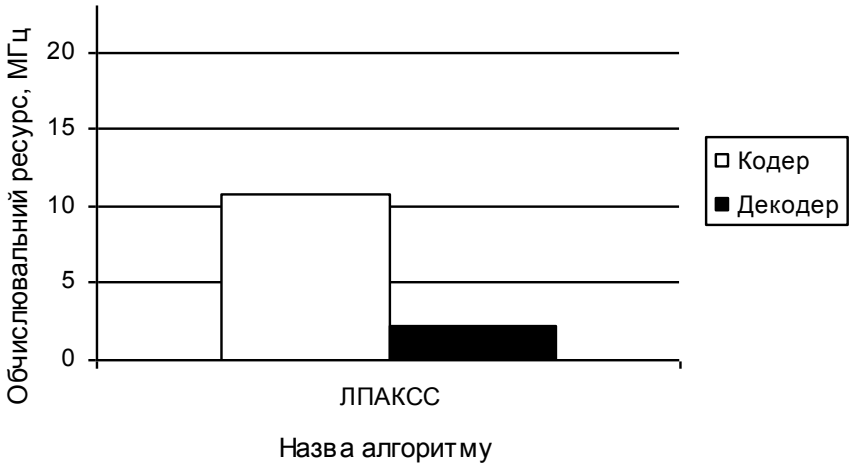


Рис. 5.8 – Частота кодеку алгоритму ЛПАКСС

На рис. 5.9 проілюстровано використання пам’яті восьми канальним кодеком алгоритму ЛПАКСС.

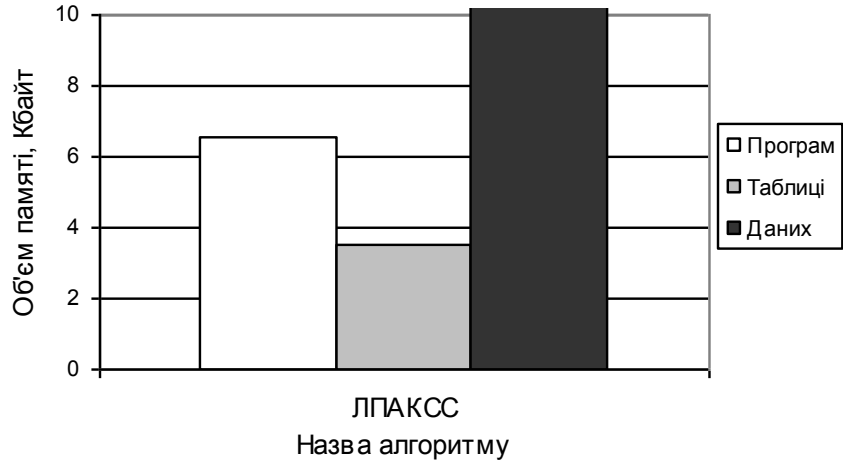


Рис. 5.9 – Використання пам’яті кодеком алгоритму ЛПАКСС

По аналогії з реалізацією БКМД всі виміри проводились на TMS320C6201 EVM з використанням Code Composer Studio 1.0 над восьми канальним кодеком алгоритму ЛПАКСС [146,148].

Схема ініціалізації структури багатоканального алгоритму ЛПАКСС наведена на рис. 5.10.

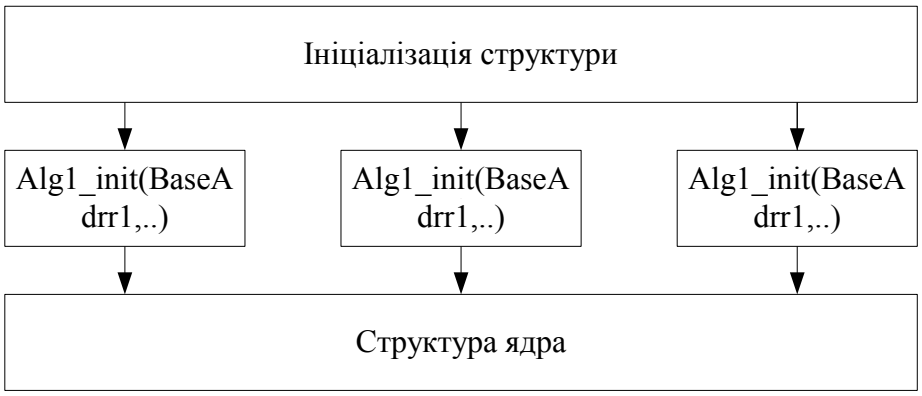


Рис. 5.10. Структурна схема ініціалізації алгоритму ЛПАКСС

Структура ядра багатоканального алгоритму ЛПАКСС наведена на рис. 5.11.

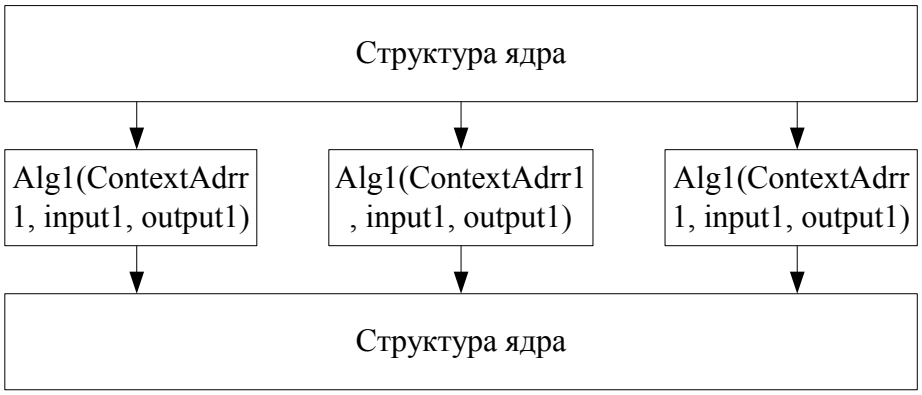


Рис. 5.11. Структура ядра багатоканального алгоритму ЛПАКСС

У додатку Е наведено інтерфейс ядра багатоканального процесора виконання алгоритмів ЛПАКСС та БКМД.

5.3. Реалізація та експериментальне дослідження транскодера між G.729A та G.723.1

5.3.1. Особливості реалізації транскодера між G.729A та G.723.1

Вхідними даними для реалізації транскодера між G.729A та G.723.1 є [60]:

-метод транскодування між G.729A та G.723.1 (див. п.2.2.1 та п.2.2.2);

-структура кодера алгоритму ЛПАКСС (рис. 3.9);

-структура кодера алгоритму БКМД (рис. 3.7);

-структура декодера алгоритму ЛПАКСС (рис. 3.5);

-структура декодера алгоритму БКМД (рис. 3.3);

-схема транскодування з G.723.1 до G.729A (рис. 3.12);

-схема транскодування з G.723.1 до G.729A (рис. 3.13);

Керування роботою транскодера виконується відповідно до алгоритму наведеного в п. 3.2.1. На рис. 5.12 наведена структурна схема ядра транскодера між G.723.1 та G.729A, який програмно реалізований на базі цифрового сигнального процесора типу TMS320C6201 [60].

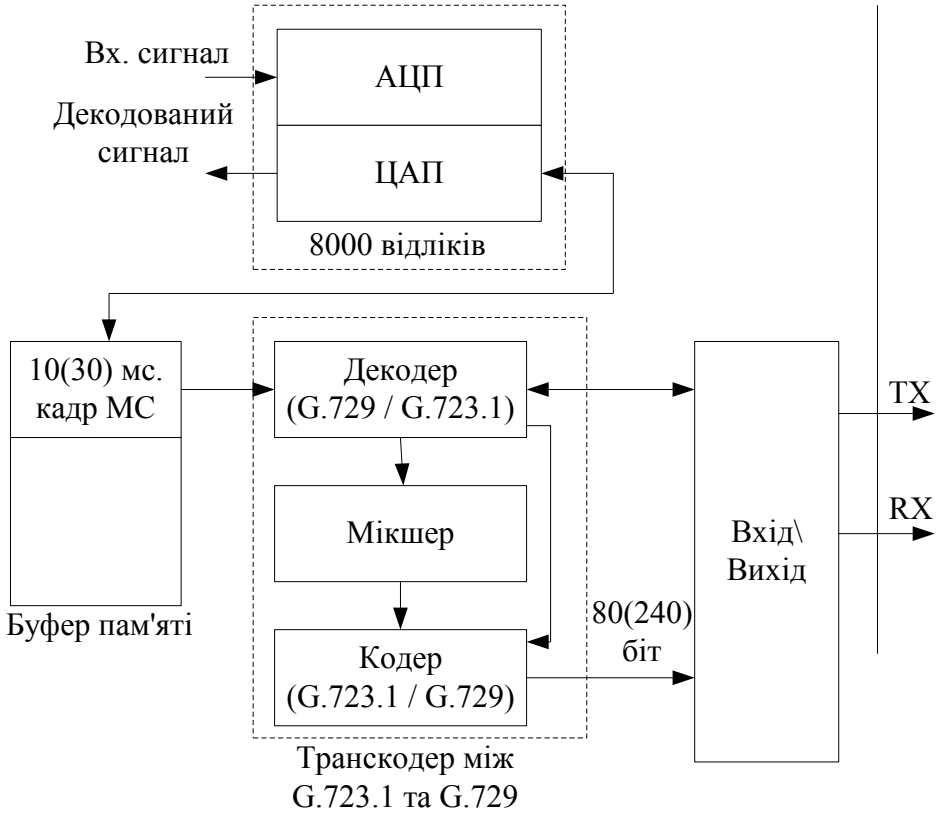


Рис 5.12. Інтерфейс ядра транскодера між G.729A та G.723.1

5.3.2. Експериментальне дослідження транскодера між G.729A та G.723.1

Для експериментальної оцінки реалізованого транскодера стиснених МС між G.729A та G.723.1 виконано тести оцінки якості мовлення PESQ та часової складності транскодування [60].

Для об’єктивної оцінки якості мови використано МС з бази фрагментів “ISABASE” [5,38]. Елементами цієї бази даних є сукупність оцифрованих звукових хвиль вимовлених російською мовою, а також додаткова інформація про ці хвилі. Додаткова інформація містить відомості про диктора та саме речення (його текст, фонетичну транскрипцію і результати ручної сегментації на слова і фонеми). Мовні матеріали записано 36 дикторами, з них 20 чоловіки і 16 жінки. Всі вони не були професійними дикторами і не мали досвіду в мистецтві мовного читання. Текстові прообрази мовних фрагментів наговорювалися в режимі дискретної читаної мови, в якій відповідний мовний фрагмент тексту вимовлявся з короткими паузами, що виразно виділялися між окремими словами. Словосполучення для оцінки взяті з розряду нейтральної лексики. Приведемо декілька прикладів таких словосполучень:

- звонок раздался совершенно неожиданно;
- руководитель разрешил произвести маневр;
- химия и физика - интересные науки.

Кожне речення мало довжину 8 секунд з частотою дискретизації 8 КГц. Дикторами взято 4 чоловіки та 4 жінки і по 24 речення на кожного. Таким чином, використано 96 речень для суб’єктивної оцінки якості мовлення (PESQ - Perceptual Evaluation of Speech Quality) [90]. У таблиці 5.3 та 5.4 наведено порівняльні результати оцінки якості мовлення відповідно до проведених тестів.

Таблиця 5.3

Оцінка якості мовлення (формат G.723.1, 5.3 Кб/с)

	Диктор	
	Чоловік	Жінка
Тандем (з G.723.1 до G.729A)	3,018	3,342
Запропонований метод (з G.723.1 до G.729A)	3,041	3,354
Тандем (з G.729A до G.723.1)	3,012	3,384
Запропонований метод (з G.729A до G.723.1)	3,065	3,412

Таблиця 5.4

Аналіз якості мовлення (формат G.723.1, 6.3 Кб/с)

	Диктор	
	Чоловік	Жінка
Тандем (з G.723.1 до G.729A)	3,106	3,453
Запропонований метод (з G.723.1 до G.729A)	3,118	3,464
Тандем (з G.729A до G.723.1)	3,101	3,456
Запропонований метод (з G.729A до G.723.1)	3,302	3,498

На рис. 5.13 та 5.14 наведено графічне відображення таблиць 5.3 та 5.4.

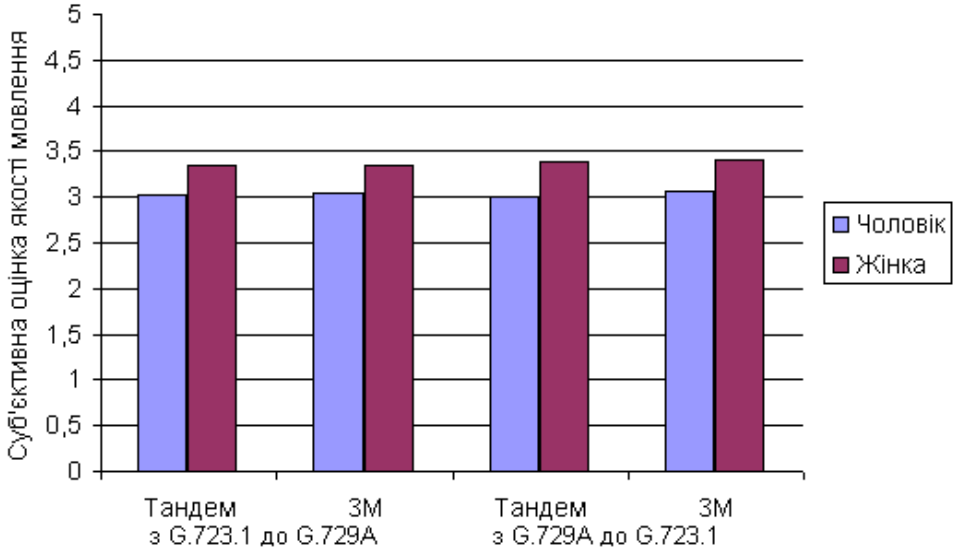


Рисунок 5.13 – Аналіз якості мовлення (формат G.723.1, 5.3 Кб/с)

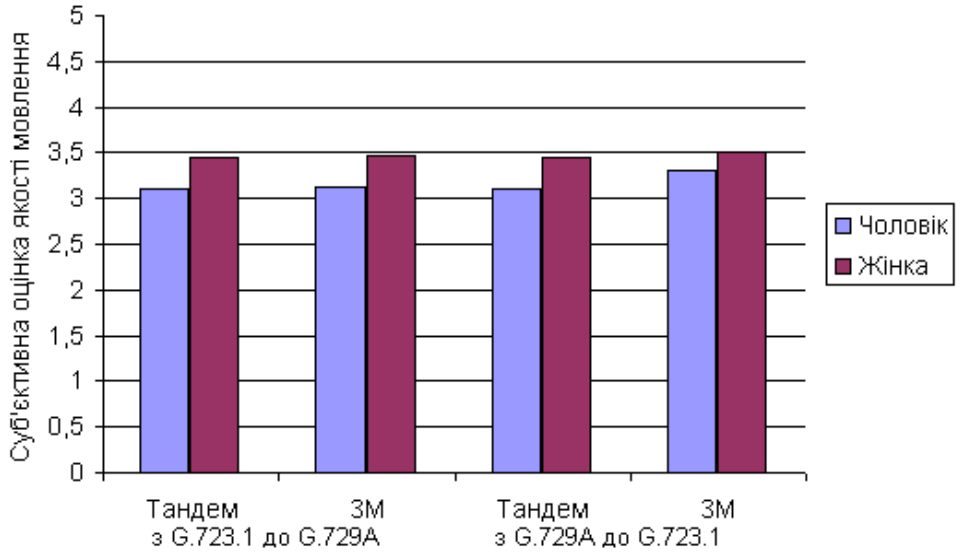


Рисунок 5.14 – Аналіз якості мовлення (формат G.723.1, 6.3 Кб/с)

Дослідження показали, що запропонований метод забезпечує на 0,3%-1,7% кращу якість мовлення під час транскодування між G.729A та G.723.1 (5,3 Кб/с) та 0,3%-6,4% під час транскодування між G.729A та G.723.1 (6,3 Кб/с) [60].

Для оцінки апаратної складності [57] запропонованого методу проведено серію обчислювальних експериментів з реалізованим транскодером на базі процесора типу TMS320C6201 (див. пункт 5.3.1). Одержані результати порівняні з даними реалізації класичного методу на цій же елементній базі, без виконання процесу оптимізації коду [60]. Результати досліджень показали, що апаратна складність стиснення МС змінюється, оскільки діапазон пошуку залежить від вхідного МС. Однак діапазон апаратної складності в модулі декодування є незначним. Результати експериментів удосконаленого транскодера стиснених МС наведені в таблиці 5.5 та на рисунку 5.15.

Таблиця 5.5

Аналіз апаратної складності використання декодерів алгоритмів БКМД і ЛПАКСС під час транскодування стиснених МС

MIPS	з G.723.1 до G.729A		з G.729A до G.723.1 (5,3 Кб/с)		з G.729A до G.723.1 (6,3 Кб/с)	
	Тандем	ЗМ*	Тандем	ЗМ	Тандем	ЗМ
КЛП і ЛСП	6,45	2,38	7,02	5,69	6,99	5, 63
Відкритий цикл	0,97	0,31	1,58	1,24	1,59	1,26
АКК	2,47	2,47	10,18	6,37	8,91	4,78
ФКК	4,32	4,32	10,54	2,22	17,32	6,68
Інші	4,06	4,06	8,0,7	8,07	8,07	8,07
Сума (декодування)	18,27	13,54	29,32	23,59	42,88	20,79
Виграш ЗМ, %	25,9		19,5		51,6	

Примітка: * ЗМ – запропонований метод

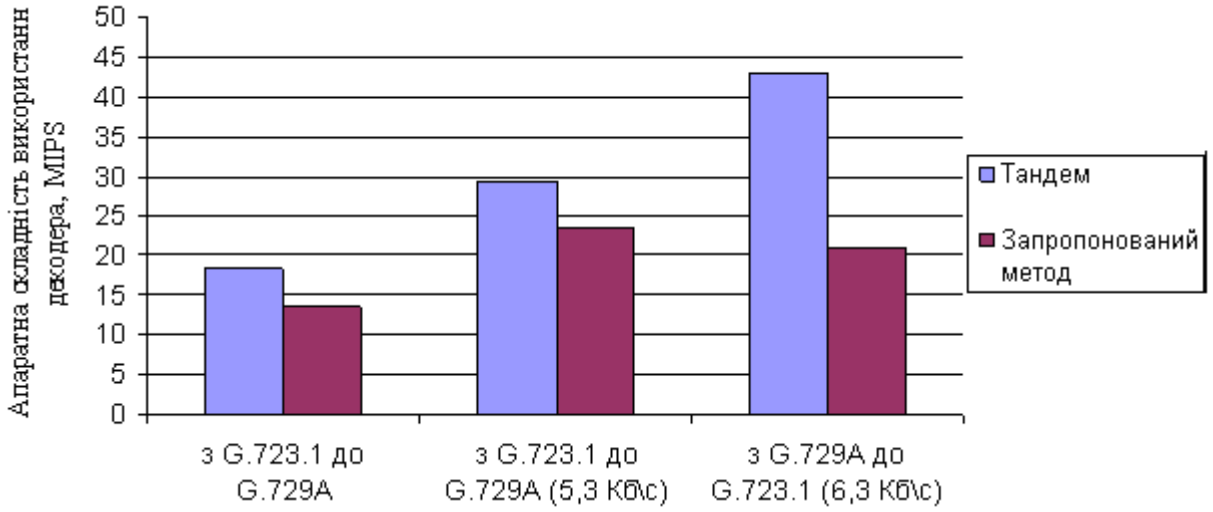


Рисунок 5.15 – Порівняльний аналіз апаратної складності використання декодерів алгоритмів БКМД та ЛПАКСС під час транскодування

Результати досліджень показали, що програмна реалізація удосконаленого транскодера на основі використання запропонованого методу транскодування між форматами G.729A та G.723.1 забезпечує кращу якість мовлення (0,3%-1,7% - під час транскодування з G.729A до G.723.1 5,3 Кб/с; 0,3%-6,4% - під час транскодування з G.729A до G.723.1 6,3 Кб/с) та меншу апаратну складність (25,9%-51,6%) у порівнянні з класичним методом.

ВИСНОВКИ

1. Розроблено програмне забезпечення для транскодування стиснених звукових та мовних сигналів, що працює з такими кодеками: WMA Voice Encoder DMO, WM Speech Encoder DMO, WM Audio Encoder DMO, 3ivx D4 Audio Encoder, Indeo Audio Software, Pinnacle AC3 Encoder, Pinnacle AC3 Encoder, Pinnacle MP3 Encoder, Pinnacle MPEG Layer-2 Audio Encoder, Vorbis Encoder, IMC, IAC2, IMA ADPCM, PCM, Ogg Vorbis, Microsoft ADPCM, ACELP.net, DSP Group TrueSpeech, Windows Media Audio, GSM 06.10, G.723.1, CCITT A-Law, CCITT u-Law, AC-3 ACM Codec, MPEG Layer-3. Частоти дискретизації використаних кодеків змінюються від 8 до 96 КГц, швидкість від 0,1 до 768 Кб/с, кількість каналів від 1 до 5. Транскодування виконуються з допомогою функцій відображення бібліотеки Audio Compression Manager.
2. Розроблено програмне забезпечення виконання алгоритмів БКМД та ЛПАКСС для процесорів типу TMS320C6201. Програмні реалізації відповідають вимогами стандартів ITU-T G.723.1, ITU-T G.729A. Проведено дослідженно апаратної складності реалізації кодеків розроблених алгоритмів.
3. На базі реалізацій кодеків, розроблено програмне забезпечення транскодування стиснених МС між форматами G.723.1 та G.729A на базі цифрового сигнального процесора типу TMS320C6201. Доведено, що удосконалений транскодер на основі використання запропонованого методу транскодування між форматами G.729A та G.723.1 забезпечує кращу якість мовлення (0,3%-1,7% - під час транскодування з G.729A до G.723.1 5,3 Кб/с; 0,3%-6,4% - під час транскодування з G.729A до G.723.1 6,3 Кб/с) та меншу апаратну складність (25,9%-51,6%) у порівнянні з класичним методом.

ЗАГАЛЬНІ ВИСНОВКИ ПО РОБОТІ

У дисертаційній роботі розв'язано наукову задачу дослідження та розробки методів і багатоканальних комп'ютерних засобів перетворення та криптографічного захисту форматів стиснених мовних сигналів, вирішення якої дає змогу розвантажити канали зв'язку, зменшити результуючі часові затримки між територіально віддаленими учасниками сеансу зв’язку, підвищити захищеність та ефективність функціонування систем зв'язку. При цьому отримано такі результати:

1. Проведено аналіз алгоритмів стиснення та мікшування МС, який дозволив їх класифікувати, виділити переваги і недоліки та окреслити перспективні напрями їх розвитку. Обґрунтовано, що найперспективнішими є алгоритми класу гібридного стиснення МС та багатоступінчастого множинного мікшування МС. Проведено порівняльний аналіз комп’ютерних засобів транскодування стиснених МС, визначено області їх доцільного використання. Доведено, що для комп’ютерних систем реального часу найбільш перспективними є апаратно реалізовані транскодери стиснених МС.

2. Запропоновано метод перетворення форматів стиснених МС між GSM 06.20 та G.729A, що враховує структурну схожість модулів короткотермінової фільтрації, довготермінової фільтрації та випадкового збудження алгоритмів ЛПГВС та ЛПАКСС, яка дає можливість провести пряме перетворення параметрів, згенерованих даними модулями. Розроблений метод дозволяє зменшити часову затримку і апаратну складність у порівнянні з класичним методом.

3. Запропоновано метод перетворення форматів стиснених МС між G.723.1 та G.729A, який дає можливість виконувати пряме перетворення ряду параметрів кадру одного формату у параметри кадру іншого та передбачає виконання чотирьох етапів: перетворення ЛСП, перетворення ВТ, пошук у

АКК та ФКК. Розроблений метод дозволяє зменшити часову затримку, апаратну складність декодера та покращити якість мовлення.

4. Вперше запропоновано метод багатоступінчастого мікшування МС на основі пам’яті з довільним доступом, який дає можливість опрацьовувати значення відліків з блоків даних, що були одержані шляхом декомпресії стиснених МС різних форматів. Процес мікшування починається при надходженні хоча б двох блоків даних у цільову комірку пам’яті з довільним доступом, що дозволяє уникати черг у буфері БМ та зменшити затримки пов’язані з часом очікування блоків даних.

5. Удосконалено структури багатоканальних транскодерів стиснених МС орієнтовані на використання в мережному обладнанні багатоабоненських конвергентних мереж. Дослідження показали, що використання удосконалених структур транскодерів дозволяє підвищити продуктивність оброблення блоків даних $L_{i,j}(c_x)$.

6. Запропоновано принципи побудови ОП криптографічних модулів процесорів підтримки протоколу IPSec. Дані принципи дозволили удосконалити характеристики структур ОП криптографічних модулів апаратно-орієнтованих процесорів підтримки протоколу IPSec для різних сервісів оброблення даних за різних технологічних характеристик компонентного базису, що забезпечило зменшення затрат обладнання на їх реалізацію.

7. Створено реалізації багатоканальних комп’ютерних засобів транскодування стиснених МС. Результати комп’ютерного моделювання показали, що реалізація удосконаленого транскодера на основі використання запропонованого методу транскодування між G.729A та G.723.1, забезпечує кращу якість мови (до 6,4%) та використовує до 51,6% менше апаратних ресурсів у порівнянні з класичним методом.

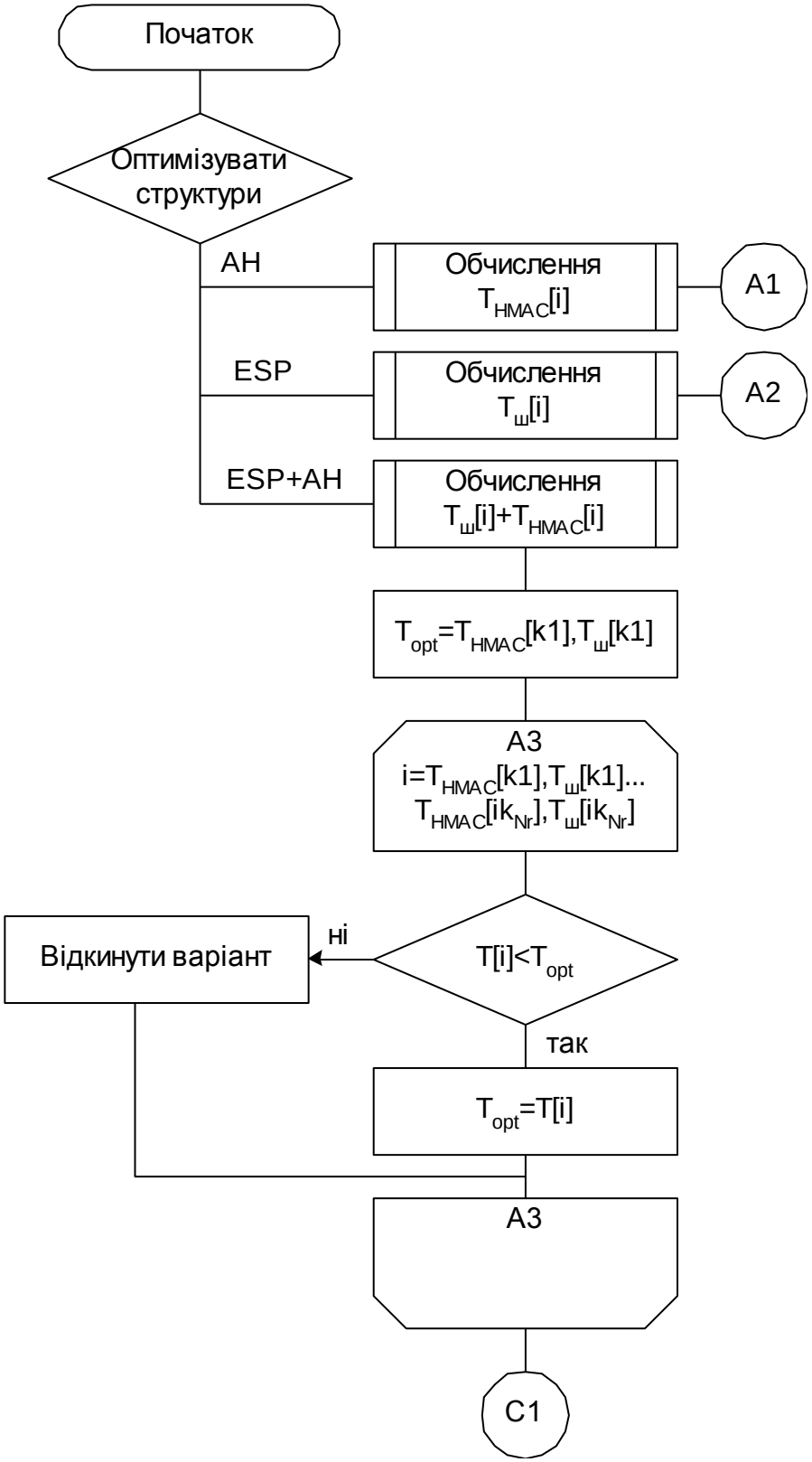
Додаток А

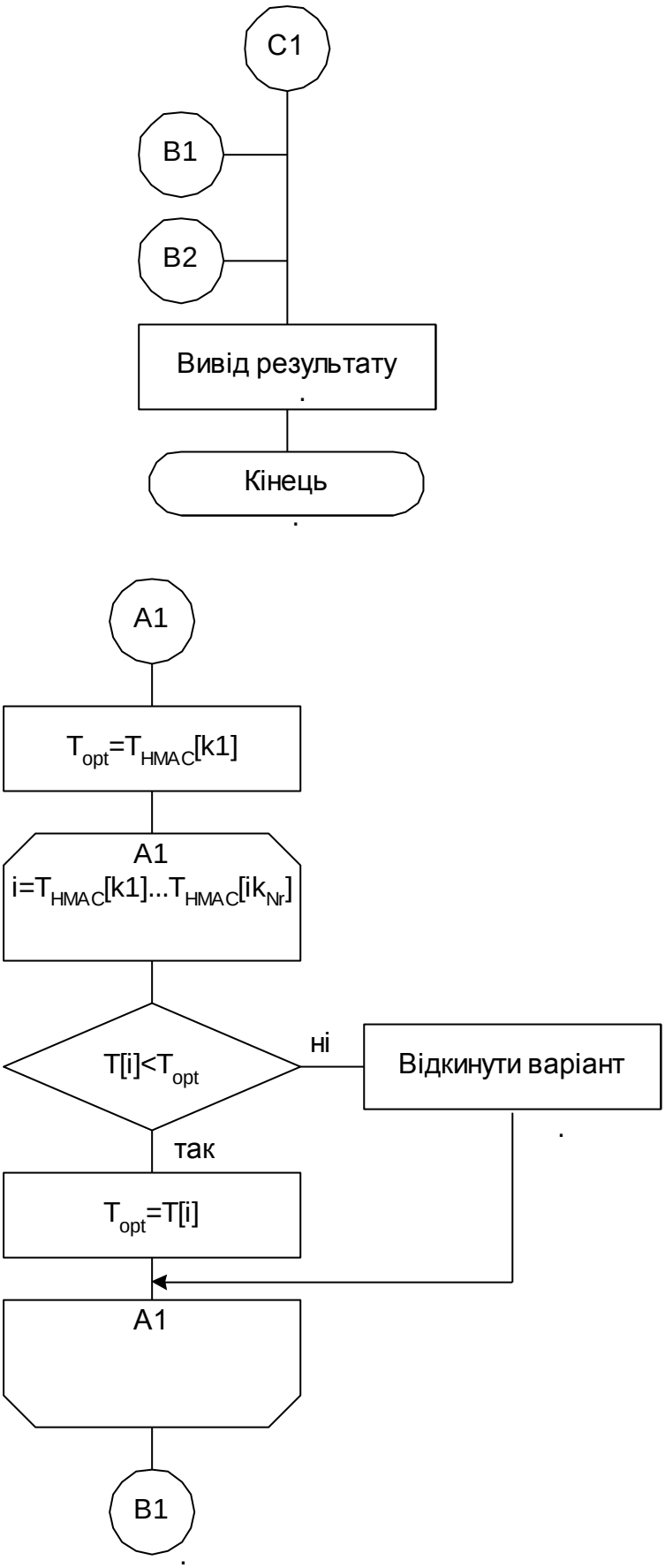
Кодування структур ОП базових криптографічних алгоритмів IPSec

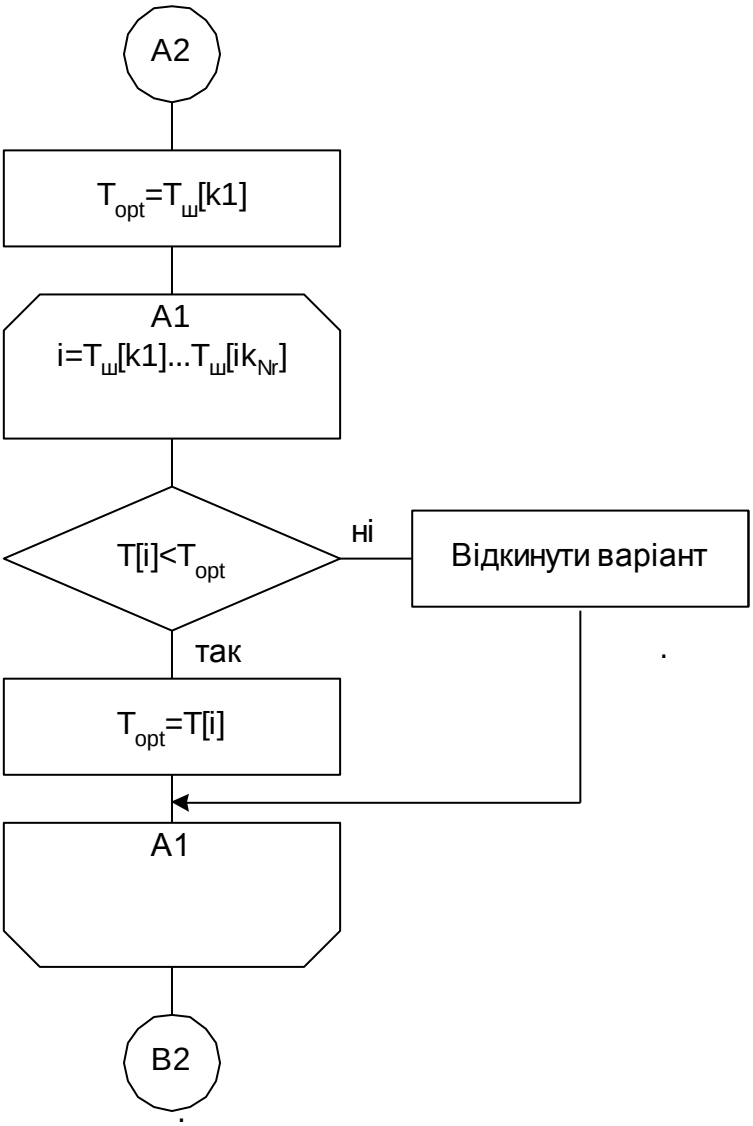
DES			MD5			SHA-1		
Код	**Nksr**	**Npp**	**Код**	**Nksr**	**Npp**	**Код**	**Nksr**	**Npp**
к1	16	2	к1	64	2	к1	80	2
к2	16	4	к2	64	4	к2	80	4
к3	16	8	к3	64	8	к3	80	5
к4	16	16	к4	64	16	к4	80	8
і1	1	1	к5	64	32	к5	80	10
і2	2	1	к6	64	64	к6	80	16
і3	4	1	і1	1	1	к7	80	20
і4	8	1	і2	2	1	к8	80	40
і5	16	1	і3	4	1	к9	80	80
ік1	2	2	і4	8	1	і1	1	1
ік2	4	2	і5	16	1	і2	2	1
ік3	8	2	і6	32	1	і3	4	1
ік4	4	4	і7	64	1	і4	5	1
ік5	8	4	ік1	2	2	і5	8	1
ік6	8	8	ік2	4	2	і6	10	1
			ік3	8	2	і7	16	1
			ік4	16	2	і8	20	1
			ік5	32	2	і9	40	1
			ік6	4	4	і10	80	1
			ік7	8	4	ік1	2	2
			ік8	16	4	ік2	4	2
			ік9	32	4	ік3	5	2
			ік10	8	8	ік4	8	2
			ік11	16	8	ік5	10	2
			ік12	32	8	ік6	16	2
			ік13	16	16	ік7	20	2
			ік14	32	16	ік8	40	2
			ік15	32	32	ік9	4	4
						ік10	5	4
						ік11	8	4
						ік12	10	4
						ік13	16	4
						ік14	20	4
						ік15	40	4
						ік16	5	5
						ік17	8	5
						ік18	10	5
						ік19	16	5
						ік20	20	5
						ік21	40	5
						ік22	8	8
						ік23	10	8
						ік24	16	8
						ік25	20	8
						ік26	40	8
						ік27	10	10
						ік28	16	10
						ік29	20	10
						ік30	40	10
						ік31	16	16
						ік32	20	16
						ік33	40	16
						ік34	20	20
						ік35	40	20
						ік36	40	40

Додаток Б

Блок схема алгоритму оптимізації структур ОП базових криптографічних алгоритмів IPSec




C1
B1
B2
Вивід результату
Кінець
A1
$T_{opt}=T_{HMAC}[k1]$
A1
$i=T_{HMAC}[k1]...T_{HMAC}[ik_{Nr}]$
$T[i]<T_{opt}$
ні
Відкинути варіант
так
$T_{opt}=T[i]$
A1
B1


A2
$T_{opt}=T_ш[k1]$
A1
$i=T_ш[k1]...T_ш[ik_{Nr}]$
$T[i]<T_{opt}$
ні
Відкинути варіант
так
$T_{opt}=T[i]$
A1
B2

Додаток В

Текст програми для знаходження оптимізованої структури операційного пристрою процесора IPSec

```
program Project;
uses
  Forms,
  Main in 'Main.pas' {frmMain},
  Graph in 'Graph.pas' {frmGraph};
{$R *.RES}
{$E IPSec.exe}
begin
  Application.Initialize;
  Application.Title := 'IPSec';
  Application.CreateForm(TfrmMain, frmMain);
  Application.CreateForm(TfrmGraph, frmGraph);
  Application.Run;
end.

unit Graph;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  ExtCtrls, TeeProcs, TeEngine, Chart, Series, OleCtrls, vcfi, raphado,
  StdCtrls, ComCtrls;
type
  TfrmGraph = class(TForm)
    PageControl1: TPageControl;
    tabAH: TTabSheet;
    tabESP: TTabSheet;
    tabKomp: TTabSheet;
    chrESP: TChart;
    Series2: TLineSeries;
    chrAHT: TChart;
    LineSeries1: TLineSeries;
    chrAH: TChart;
    Series1: TLineSeries;
    chrKom: TChart;
    LineSeries3: TLineSeries;
    chrESPT: TChart;
    LineSeries2: TLineSeries;
    procedure FormResize(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  frmGraph: TfrmGraph;
```

```
implementation
{$R *.DFM}
procedure TfrmGraph.FormResize(Sender: TObject);
begin
  if Height<500 then height:=500 ;
  if width<600 then width:=600;
  chrAH.Height:=height div 2-20;
  chrESP.Height:=Height div 2-20;
end;
end.
unit Main;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, Mask, Spin, ExtCtrls, graph;
// User Objects
type
  TInterCon = record
    Nkrs:integer;
    Npp:integer;
    Takt:real;
    Time:real;

end;
```

// Реалізація інтерфейсу програмного забезпечення

```
type
  TfrmMain = class(TForm)
    grbInput: TGroupBox;
    labTitle: TLabel;
    labTkci: TLabel;
    labTkom: TLabel;
    labTkcnr: TLabel;
    labTpg: TLabel;
    labAlgoritm: TLabel;
    cmbAlgo: TComboBox;
    edtTkom: TEdit;
    edtTkc1: TEdit;
    edtTkcnr: TEdit;
    edtTpg: TEdit;
    labShifr: TLabel;
    edtSTkom: TEdit;
    edtSTkc1: TEdit;
    edtSTpg: TEdit;
    cmbSAlgo: TComboBox;
    labPaketSize: TLabel;
    edtPakSize: TEdit;
    btnClear: TButton;
    btnCalc: TButton;
    btnCancel: TButton;
    grbOpt: TGroupBox;
```

```
    labTakt: TLabel;
    labTime: TLabel;
    labAH: TLabel;
    labESP: TLabel;
    chbTaktAH: TCheckBox;
    chbTimeAH: TCheckBox;
    chbTaktESP: TCheckBox;
    chbTimeESP: TCheckBox;
    grbResult: TGroupBox;
    lsbResult: TListBox;
    edtPaketSize: TEdit;
    labOptInfo: TLabel;
    btnGraph: TButton;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label7: TLabel;
    Label6: TLabel;
    Label8: TLabel;
    procedure FormCreate(Sender: TObject);
    procedure edtTkomExit(Sender: TObject);
    procedure edtTkc1Exit(Sender: TObject);
    procedure edtTkcnrExit(Sender: TObject);
    procedure edtTpgExit(Sender: TObject);
    procedure cmbAlgoChange(Sender: TObject);
    procedure edtPakSizeExit(Sender: TObject);
    procedure btnClearClick(Sender: TObject);
    procedure btnCancelClick(Sender: TObject);
    procedure btnCalcClick(Sender: TObject);
    procedure edtSTkomExit(Sender: TObject);
    procedure edtSTkc1Exit(Sender: TObject);
    procedure edtSTpgExit(Sender: TObject);
    procedure btnGraphClick(Sender: TObject);
    procedure chbTaktESPClick(Sender: TObject);
    procedure chbTaktAHClick(Sender: TObject);
    procedure chbTimeAHClick(Sender: TObject);
    procedure chbTimeESPClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmMain: TfrmMain;

  // Оголошення масивів вхідних даних
  Algo:integer;
  tmp:integer;
  Inter:array [1..10] of TInterCon;
```

```
  Conv:array [1..9] of TInterCon;
  InterCon:array [1..36] of TInterCon;
  SInter:array [1..4] of TInterCon;
  SConv:array [1..4] of TInterCon;
  SInterCon:array [1..6] of TInterCon;
  Compl:array [1..55,1..14] of real;
  si,sj:string;

  // Оголошення змінних, що будуть використовуватись при оптимізації
  minTaktAH:integer;
  minTaktESP:integer;
  minTimeAH:integer;
  minTimeESP:integer;
  minTaktAH_ESP:integer;
  minTimeAH_ESP:integer;
  minCompli:integer;
  minComplj:integer;
  scompli:integer;
  scomplj:integer;
  TimeM:integer;
  TaktM:integer;
  TimeN:integer;
  TaktN:integer;

function TrueData:boolean;

// Оголошення процедур для створення структур даних
procedure SetInter;
procedure SetConv;
procedure SetInterCon;
procedure SetSInter;
procedure SetSConv;
procedure SetSInterCon;

// Оголошення процедур для обчислень
procedure CalcInter;
procedure CalcConv;
procedure CalcInterCon;
procedure CalcSInter;
procedure CalcSConv;

procedure CalcOpt;

// Обчислення параметрів структур

implementationProcedure CalcOpt;
var maxi:integer;
    i,j:integer;
    p:integer;
begin
  // Initilize optimiz
  if Algo=80 then maxi:=10 else maxi:=7;
```

```
minTimeAH:=1;
minTimeESP:=1;
minTimeAH_ESP:=1;
minTaktAH:=1;
minTaktESP:=1;
minTaktAH_ESP:=1;

TimeM:=1;  // 1-Inter 2-Conv 3-InterCon
TaktM:=1;  // 1-Inter 2-Conv 3-InterCon

// Знаходження часу і такту сервісу АН
for i:=2 to maxi do
begin
  if Inter[minTimeAH].time>Inter[i].Time then MinTimeAH:=i;
  if Inter[minTaktAH].takt>Inter[i].Takt then MinTaktAH:=i;
end;
if inter[minTimeAH].time>Conv[1].Time then
begin
  minTimeAH:=1;
  TimeM:=2;
end;
if Inter[minTaktAH].takt>Conv[1].Takt then
begin
  minTaktAH:=1;
  TaktM:=2;
end;
for i:=2 to maxi-1 do
begin
  if TimeM=1 then
  begin
    if Inter[minTimeAH].time>Conv[i].Time then
    begin
      minTimeAH:=i;
      TimeM:=2;
    end;
  end else
  begin
    if Conv[minTimeAH].time>Conv[i].Time then
    begin
      minTimeAH:=i;
      TimeM:=2;
    end;
  end;
  if Conv[minTaktAH].takt>Conv[i].Takt then
  if TaktM=1 then
  begin
    if Inter[minTaktAH].takt>Conv[i].Takt then
    begin
      minTaktAH:=i;
      TaktM:=2;
    end;
  end else
```

```
      if Conv[minTaktAH].takt>Conv[i].Takt then
      begin
        minTaktAH:=i;
        TaktM:=2;
      end;
    begin
    end;
  end;

  // Знаходження мінімального такту та часу при суміщенні сервісів
  if TimeM=1 then
  begin
    if TimeN=1 then if Inter[minTimeAH].time>SInter[minTimeESP].time then
minTimeAH_ESP:=2 else minTimeAH_ESP:=1;
    if TimeN=2 then if Inter[minTimeAH].time>SConv[minTimeESP].time then
minTimeAH_ESP:=2 else minTimeAH_ESP:=1;
    if TimeN=3 then if Inter[minTimeAH].time>SInterCon[minTimeESP].time then
minTimeAH_ESP:=2 else minTimeAH_ESP:=1;
  end;
  if TimeM=2 then
  begin
    if TimeN=1 then if Conv[minTimeAH].time>SInter[minTimeESP].time then
minTimeAH_ESP:=2 else minTimeAH_ESP:=1;
    if TimeN=2 then if Conv[minTimeAH].time>SConv[minTimeESP].time then
minTimeAH_ESP:=2 else minTimeAH_ESP:=1;
    if TimeN=3 then if Conv[minTimeAH].time>SInterCon[minTimeESP].time then
minTimeAH_ESP:=2 else minTimeAH_ESP:=1;
  end;
  if TimeM=3 then
  begin
    if TimeN=1 then if InterCon[minTimeAH].time>SInter[minTimeESP].time then
minTimeAH_ESP:=2 else minTimeAH_ESP:=1;
    if TimeN=2 then if InterCon[minTimeAH].time>SConv[minTimeESP].time then
minTimeAH_ESP:=2 else minTimeAH_ESP:=1;
    if TimeN=3 then if InterCon[minTimeAH].time>SInterCon[minTimeESP].time then
minTimeAH_ESP:=2 else minTimeAH_ESP:=1;
  end;
  if TaktM=1 then
  begin
    if TaktN=1 then if Inter[minTaktAH].takt>SInter[minTaktESP].takt then minTaktAH_ESP:=2
else minTaktAH_ESP:=1;
    if TaktN=2 then if Inter[minTaktAH].takt>SConv[minTaktESP].takt then
minTaktAH_ESP:=2 else minTaktAH_ESP:=1;
    if TaktN=3 then if Inter[minTaktAH].takt>SInterCon[minTaktESP].takt then
minTaktAH_ESP:=2 else minTaktAH_ESP:=1;
  end;
  if TaktM=2 then
  begin
    if TaktN=1 then if Conv[minTaktAH].takt>SInter[minTaktESP].takt then
minTaktAH_ESP:=2 else minTaktAH_ESP:=1;
    if TaktN=2 then if Conv[minTaktAH].takt>SConv[minTaktESP].takt then
minTaktAH_ESP:=2 else minTaktAH_ESP:=1;
```

```
   if TaktN=3 then if Conv[minTaktAH].takt>SInterCon[minTaktESP].takt then
minTaktAH_ESP:=2 else minTaktAH_ESP:=1;
  end;
  if TimeM=3 then
  begin
   if TaktN=1 then if InterCon[minTaktAH].takt>SInter[minTaktESP].takt then
minTaktAH_ESP:=2 else minTaktAH_ESP:=1;
   if TaktN=2 then if InterCon[minTaktAH].takt>SConv[minTaktESP].takt then
minTaktAH_ESP:=2 else minTaktAH_ESP:=1;
   if TaktN=3 then if InterCon[minTaktAH].takt>SInterCon[minTaktESP].takt then
minTaktAH_ESP:=2 else minTaktAH_ESP:=1;
  end;
```

// Знаходження параметрів ітераційно-конвеєрної структури

```
procedure CalcInterCon;
var i:integer;
   maxi:integer;
begin
   if Algo=80 then maxi:=36 else maxi:=15;
   for i:=1 to maxi do
   begin

InterCon[i].takt:=InterCon[i].Npp*strtofloat(frmMain.edtTpg.text)+InterCon[i].Nkrs*strtofloat(f
rmMain.edtTkc1.text)+strtofloat(frmMain.edtTkom.text);

InterCon[i].time:=Algo/InterCon[i].Nkrs*strtoint(frmMain.edtPaketSize.text)*InterCon[i].takt+s
trtofloat(frmMain.edtTkcnr.text)+strtofloat(frmMain.edtTpg.text);
     frmGraph.chrAH.Series[0].add(InterCon[i].time,'IK'+inttostr(i),1);
     frmGraph.chrAHT.Series[0].add(InterCon[i].takt,'IK'+inttostr(i),1);
   end;
end;
```

// Процедура визначення параметрів оптимальної структури операційного пристрою

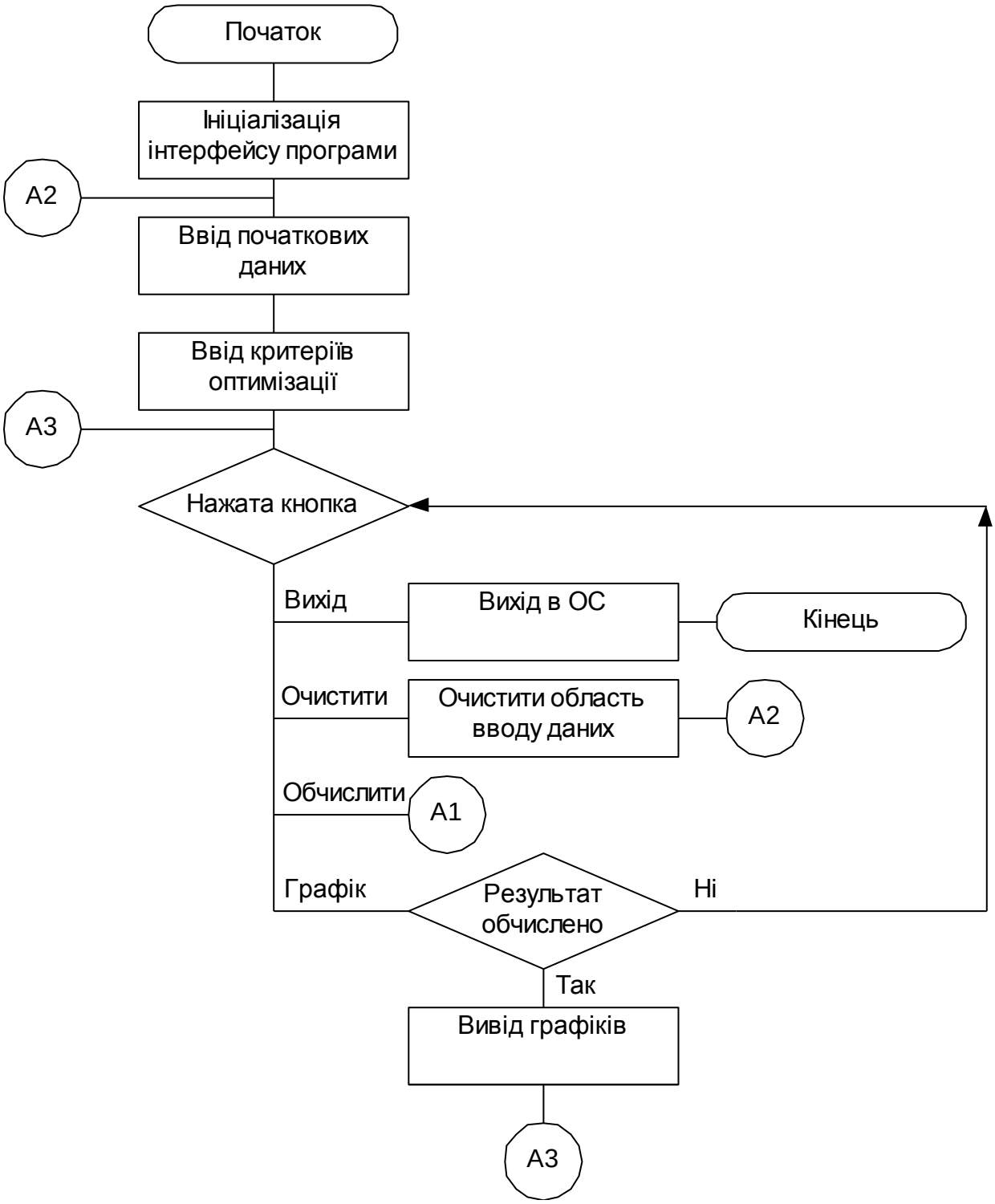
```
procedure TfrmMain.btnCalcClick(Sender: TObject);
var opt:String;
   sesptemp:string;
   sahtemp:string;
begin
  if TrueData then
  begin
   btnGraph.Enabled:=true;
   frmGraph.chrAH.Series[0].Clear;
   frmGraph.chrESP.Series[0].Clear;
   frmGraph.chrKom.Series[0].Clear;
   frmGraph.chrESPT.Series[0].Clear;
   frmGraph.chrAHT.Series[0].Clear;

   CalcSInter;
   CalcSConv;
   CalcSInterCon;
   CalcInter;
   CalcConv;
```

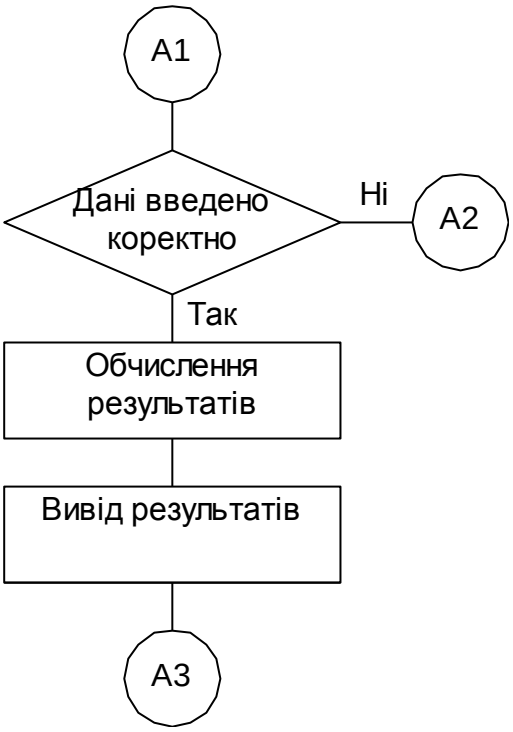
```
   CalcInterCon;
   CalcOpt;

   lsbResult.Clear;
   if(chbTaktAH.Checked)and(not chbTaktESP.Checked) then
   begin
     opt:='Оптимізяція такту АН: t=';
     if TaktM=1 then opt:=opt+floattostr(Inter[minTaktAH].Takt)+'ns структура
I('+floattostr(Inter[minTaktAH].Nkrs)+';'+floattostr(Inter[minTaktAH].Npp)+').';
     if TaktM=2 then opt:=opt+floattostr(Conv[minTaktAH].Takt)+'ns структура
K('+floattostr(Conv[minTaktAH].Nkrs)+';'+floattostr(Conv[minTaktAH].Npp)+').';
     if TaktM=3 then opt:=opt+floattostr(InterCon[minTaktAH].Takt)+'ns структура
IK('+floattostr(InterCon[minTaktAH].Nkrs)+';'+floattostr(InterCon[minTaktAH].Npp)+').';
   end;
   if(not chbTaktAH.Checked)and(chbTaktESP.Checked) then
   begin
     opt:='Оптимізяція такту ESP: t=';
     if TaktN=1 then opt:=opt+floattostr(SInter[minTaktESP].Takt)+'ns структура
I('+floattostr(SInter[minTaktESP].Nkrs)+';'+floattostr(SInter[minTaktESP].Npp)+').';
     if TaktN=2 then opt:=opt+floattostr(SConv[minTaktESP].Takt)+'ns структура
K('+floattostr(SConv[minTaktESP].Nkrs)+';'+floattostr(SConv[minTaktESP].Npp)+').';
     if TaktN=3 then opt:=opt+floattostr(SInterCon[minTaktESP].Takt)+'ns структура
IK('+floattostr(SInterCon[minTaktESP].Nkrs)+';'+floattostr(SInterCon[minTaktESP].Npp)+').';
   end;
   if(chbTaktAH.Checked)and(chbTaktESP.Checked) then
   begin
     if TaktN=1 then
sesptemp:='I('+floattostr(SInter[minTaktESP].Nkrs)+','+floattostr(SInter[MinTaktESP].Npp)+').';
     if TaktN=2 then
sesptemp:='K('+floattostr(SConv[minTaktESP].Nkrs)+','+floattostr(SConv[MinTaktESP].Npp)+'
).';
     if TaktN=3 then
sesptemp:='IK('+floattostr(SInterCon[minTaktESP].Nkrs)+','+floattostr(SInterCon[MinTaktESP]
.Npp)+').';

     if TaktM=1 then
sahtemp:='I('+floattostr(Inter[minTaktESP].Nkrs)+','+floattostr(Inter[MinTaktESP].Npp)+').';
     if TaktM=2 then
sahtemp:='K('+floattostr(Conv[minTaktESP].Nkrs)+','+floattostr(Conv[MinTaktESP].Npp)+').';
     if TaktM=3 then
sahtemp:='IK('+floattostr(InterCon[minTaktESP].Nkrs)+','+floattostr(InterCon[MinTaktESP].Np
p)+').';

     if minTaktAH_ESP=1 then
     begin
       opt:='Оптимізяція такту АН: t=';
       if TaktM=1 then opt:=opt+floattostr(Inter[minTaktAH].Takt)+'ns структура
AH.I('+floattostr(Inter[minTaktAH].Nkrs)+';'+floattostr(Inter[minTaktAH].Npp)+'),
'+'структура ESP.'+sesptemp;
       if TaktM=2 then opt:=opt+floattostr(Conv[minTaktAH].Takt)+'ns структура
AH.K('+floattostr(Conv[minTaktAH].Nkrs)+';'+floattostr(Conv[minTaktAH].Npp)+'),
'+'структура ESP.'+sesptemp;
```

```
      if TaktM=3 then opt:=opt+floattostr(InterCon[minTaktAH].Takt)+'ns структура
AH.IK('+floattostr(InterCon[minTaktAH].Nkrs)+';'+floattostr(InterCon[minTaktAH].Npp)+'),
'+'структура ESP.'+sesptemp;
     end else
     begin
      opt:='Оптимізяція такту ESP: t=';
      if TaktN=1 then opt:=opt+floattostr(SInter[minTaktESP].Takt)+'ns структура
ESP.I('+floattostr(SInter[minTaktESP].Nkrs)+';'+floattostr(SInter[minTaktESP].Npp)+'),
'+'структура AH.'+sahtemp;
      if TaktN=2 then opt:=opt+floattostr(SConv[minTaktESP].Takt)+'ns структура
ESP.K('+floattostr(SConv[minTaktESP].Nkrs)+';'+floattostr(SConv[minTaktESP].Npp)+'),
'+'структура AH.'+sahtemp;
      if TaktN=3 then opt:=opt+floattostr(SInterCon[minTaktESP].Takt)+'ns структура
ESP.IK('+floattostr(SInterCon[minTaktESP].Nkrs)+';'+floattostr(SInterCon[minTaktESP].Npp)+
'), '+'структура AH.'+sahtemp;
     end;

   end;
   lsbResult.Items.Add(opt);
 //************************************
   if(chbTimeAH.Checked)and(not chbTimeESP.Checked) then
   begin
     opt:='Оптимізяція часу AH: T=';
     if TimeM=1 then opt:=opt+floattostr(Inter[minTimeAH].Time)+'ns структура
I('+floattostr(Inter[minTimeAH].Nkrs)+';'+floattostr(Inter[minTimeAH].Npp)+').';
     if TimeM=2 then opt:=opt+floattostr(Conv[minTimeAH].Time)+'ns структура
K('+floattostr(Conv[minTimeAH].Nkrs)+';'+floattostr(Conv[minTimeAH].Npp)+').';
     if TimeM=3 then opt:=opt+floattostr(InterCon[minTimeAH].Time)+'ns структура
IK('+floattostr(InterCon[minTimeAH].Nkrs)+';'+floattostr(InterCon[minTimeAH].Npp)+').';
   end;
   if(not chbTimeAH.Checked)and(chbTimeESP.Checked) then
   begin
     opt:='Оптимізяція часу ESP: T=';
     if TimeN=1 then opt:=opt+floattostr(SInter[minTimeESP].Time)+'ns структура
I('+floattostr(SInter[minTimeESP].Nkrs)+';'+floattostr(SInter[minTimeESP].Npp)+').';
     if TimeN=2 then opt:=opt+floattostr(SConv[minTimeESP].Time)+'ns структура
K('+floattostr(SConv[minTimeESP].Nkrs)+';'+floattostr(SConv[minTimeESP].Npp)+').';
     if TimeN=3 then opt:=opt+floattostr(SInterCon[minTimeESP].Time)+'ns структура
IK('+floattostr(SInterCon[minTimeESP].Nkrs)+';'+floattostr(SInterCon[minTimeESP].Npp)+').';
   end;
   if(chbTimeAH.Checked)and(chbTimeESP.Checked) then
   begin
     opt:='Оптимізація часу AH+ESP: T='+floattostr(Compl[minCompli,minComplj])+'
структура '+si+sj+'.';
   end;
   lsbResult.Items.Add(opt);
 end;
end;
```

Додаток Г

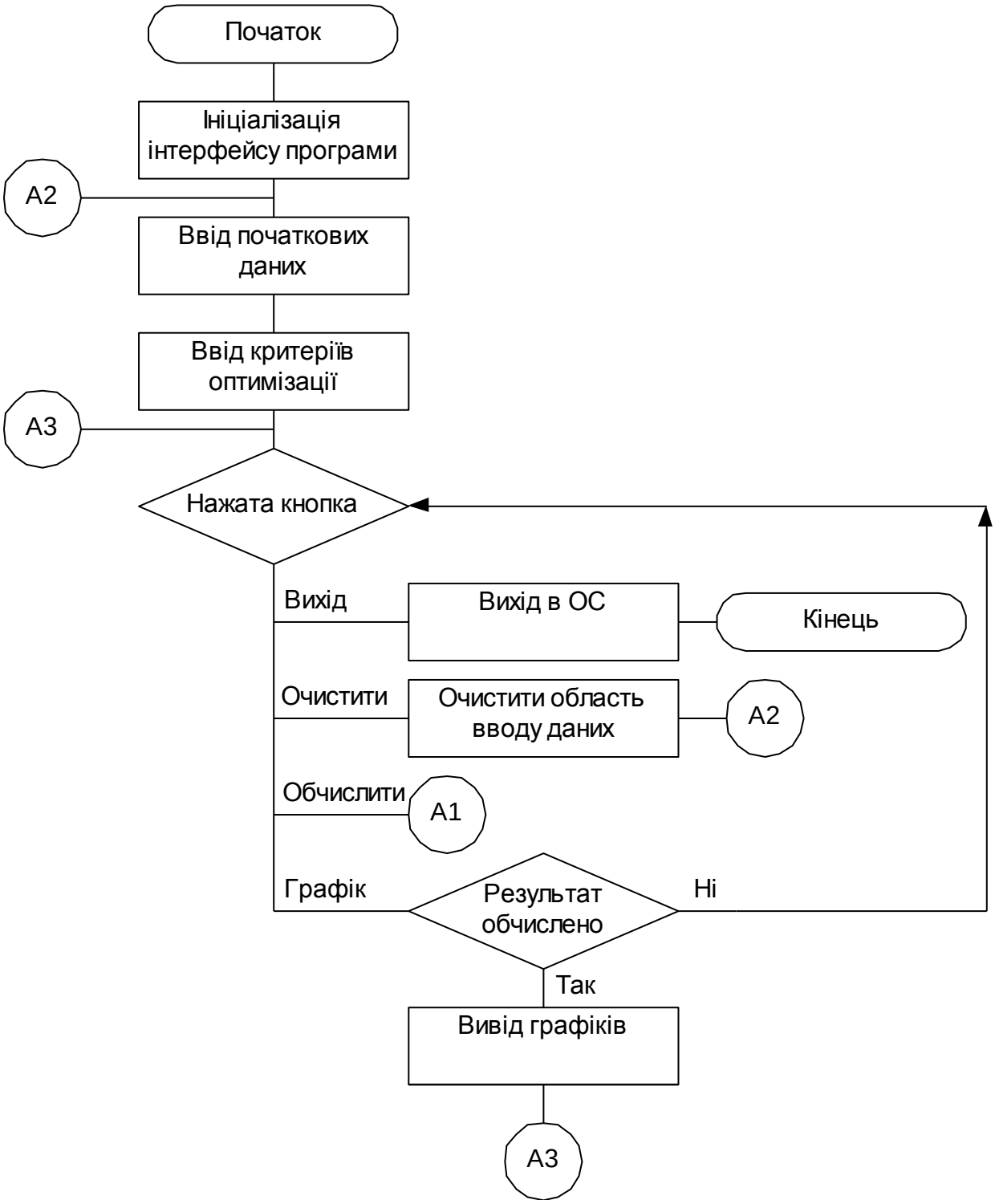
Блок схема програми оптимізації структур ОП базових криптографічних алгоритмів IPSec

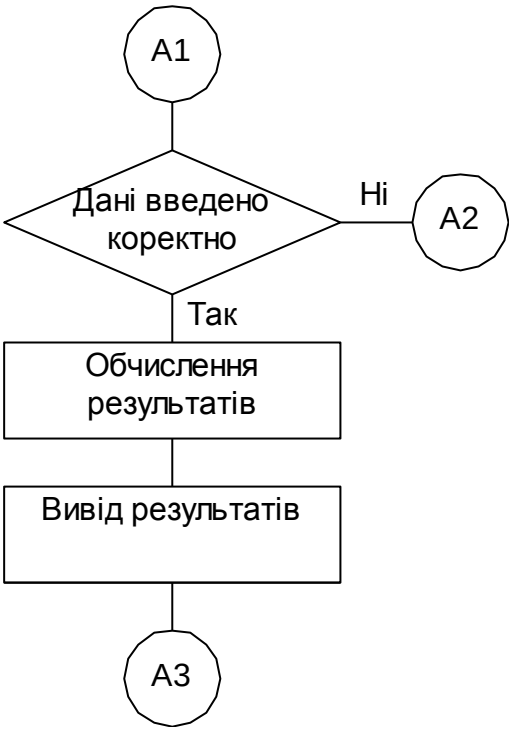



A1
Дані введено коректно
Ні
A2
Так
Обчислення результатів
Вивід результатів
A3

Додаток Д

Блок схема програми оптимізації структур ОП базових криптографічних алгоритмів IPSec




A1
Дані введено коректно
Ні
A2
Так
Обчислення результатів
Вивід результатів
A3

Додаток Е

Лістинг програми перетворення форматів стиснених звукових та мовних сигналів

```
// DSEncoder.cpp: implementation of the CDSEncoder class.
//////////////////////////////////////////////////////////////////////
#include "stdafx.h"
#include "DShowEncoder.h"
#include "DSEncoder.h"
#ifdef _DEBUG
#undef THIS_FILE
static char THIS_FILE[]=__FILE__;
#define new DEBUG_NEW
#endif
//////////////////////////////////////////////////////////////////////
// Construction/Destruction
//////////////////////////////////////////////////////////////////////
CDSEncoder::CDSEncoder()
{
	m_pGraphBuilder = NULL;
	BuildCodecArray();
}

CDSEncoder::~CDSEncoder()
{
	// Creal the codec collection
	int nNumberOfCodec = GetSize();
	for (int i=0; i<nNumberOfCodec; i++) {
		delete GetAt(i);
	}
	RemoveAll();
}
void CDSEncoder::BuildCodecArray()
{
	HRESULT			hr;
	ICreateDevEnum*		pSysDevEnum = NULL;
	IEnumMoniker*		pEnum = NULL;
	IMoniker*		pMoniker = NULL;
	// System Device Enumerator
	hr = CoCreateInstance(CLSID_SystemDeviceEnum, NULL,
CLSCTX_INPROC_SERVER, IID_ICreateDevEnum, (void**) &pSysDevEnum);
	if (FAILED(hr)) {
		// ERROR HERE
```

```
                return;
        }
        // Moniker Enumerator
        hr = pSysDevEnum-
>CreateClassEnumerator(CLSID_AudioCompressorCategory, &pEnum, 0);
        if (FAILED(hr)) {
                // ERROR HERE
                pSysDevEnum->Release();
                return;
        }
        // Cycle throught IMoniker collection
        while (pEnum->Next(1, &pMoniker, NULL) == S_OK) {
                // New instance of CDSCodec
                CDSCodec *pCodec = new CDSCodec();
                pCodec->m_pMoniker = pMoniker;
                // Retrieve codec name
                IPropertyBag *pPropertyBag = NULL;
                hr = pMoniker->BindToStorage(NULL, NULL, IID_IPropertyBag,
(void**) &pPropertyBag);
                if (SUCCEEDED(hr)) {
                        VARIANT var;
                        VariantInit(&var);
                        pPropertyBag->Read(L"FriendlyName", &var, NULL);
                        CString szTempName(var.bstrVal);
                        pCodec->m_szCodecName = szTempName;
                        VariantClear(&var);
                        // Add new instance to collection
                        Add(pCodec);
                        pCodec->BuildCodecFormatArray();
                } else {
                        delete pCodec;
                }
        }
        // Libération des enumerators
        pEnum->Release();
        pSysDevEnum->Release();
}
HRESULT CDSEncoder::AddFilterByClsid(IGraphBuilder *pGraph, LPCWSTR
wszName, const GUID& clsid, IBaseFilter **ppF)
{
   *ppF = NULL;
   HRESULT hr = CoCreateInstance(clsid, NULL, CLSCTX_INPROC_SERVER,
     IID_IBaseFilter, (void**)ppF);
   if (SUCCEEDED(hr))
   {
```

```
        hr = pGraph->AddFilter((*ppF), wszName);
    }
    return hr;
}
BOOL CDSEncoder::SetFilterFormat(AM_MEDIA_TYPE* pStreamFormat,
IBaseFilter* pBaseFilter)
{
        HRESULT hr;
        BOOL retVal = FALSE;

        // Pin enumeration
        IEnumPins* pEnumPins = NULL;
        hr = pBaseFilter->EnumPins(&pEnumPins);
        if (FAILED(hr)) {
                // ERROR HERE
                return FALSE;
        }
        IPin* pPin = NULL;
        while (pEnumPins->Next(1, &pPin, 0) == S_OK) {
                PIN_DIRECTION sDirection;
                pPin->QueryDirection(&sDirection);
                // Output Pin ?
                if (sDirection == PINDIR_OUTPUT) {
                        IAMStreamConfig* pStreamConfig = NULL;
                        hr = pPin->QueryInterface(IID_IAMStreamConfig, (void**)
&pStreamConfig);
                        if (SUCCEEDED(hr)) {
                                hr = pStreamConfig->SetFormat(pStreamFormat);
                                if (SUCCEEDED(hr)) {
                                        retVal = TRUE;
                                }
                                pStreamConfig->Release();
                        }
                }
                pPin->Release();
        }
        // Free memory
        pEnumPins->Release();

        return retVal;
}
void CDSEncoder::BuildGraph(CString szSrcFileName, CString szDestFileName,
int nCodec, int nFormat)
{
        HRESULT hr;
```

```
	IBaseFilter *pParser = NULL, *pCodec = NULL, *pMux = NULL, *pDest
= NULL;
	IFileSinkFilter* pSink = NULL;
	IFileSourceFilter* pSourceFilter = NULL;
	GUID CLSID_WavParser;
	UuidFromString((unsigned char*)"3C78B8E2-6C4D-11D1-ADE2-
0000F8754B99", &CLSID_WavParser);
	GUID CLSID_WavDest;
	UuidFromString((unsigned char*)"D51BD5A1-7548-11CF-A520-
0080C77EF58A", &CLSID_WavDest);
	// GraphBuilder construction
	hr = CoCreateInstance(CLSID_FilterGraph, NULL,
CLSCTX_INPROC_SERVER, IID_IGraphBuilder, (void**)
&m_pGraphBuilder);
	if (SUCCEEDED(hr)) {
		// Parse filter
		hr = AddFilterByClsid(m_pGraphBuilder, L"Parser",
CLSID_WavParser, &pParser);
		// ACM codec filter
		IMoniker* pMoniker = GetAt(nCodec)->m_pMoniker;;
		pMoniker->BindToObject(NULL, NULL, IID_IBaseFilter, (void**)
&pCodec);
		hr = m_pGraphBuilder->AddFilter(pCodec, L"ACM Codec");
		// Mux filter
		hr = AddFilterByClsid(m_pGraphBuilder, L"WavDest",
CLSID_WavDest, &pMux);
		// Output file filter
		hr = AddFilterByClsid(m_pGraphBuilder, L"File Writer",
CLSID_FileWriter, &pDest);
		pDest->QueryInterface(IID_IFileSinkFilter, (void**) &pSink);
		pSink->SetFileName(szDestFileName.AllocSysString(), NULL);
		pSink->Release();
		// Calculate output file size
		CFileStatus fileStatus;
		CFile::GetStatus(szSrcFileName, fileStatus);
		int nInputFileSize = fileStatus.m_size;
		// Assuming 44kHz, 16bits, stereo
		int nMediaTime = (int) (nInputFileSize / (44000 * 2 * 2));
		WAVEFORMATEX *pWav = (WAVEFORMATEX *)
GetAt(nCodec)->GetAt(nFormat)->m_pMediaType->pbFormat;
		int nOutputSize = nMediaTime * (pWav->nAvgBytesPerSec);

		// Check for output File
		try {
			CFile::Remove(szDestFileName);
```

```
            } catch(...) {
                  // nothing to do
            }
            // Render Graph
            hr = m_pGraphBuilder->RenderFile(szSrcFileName.AllocSysString(),
NULL);
            if (SUCCEEDED(hr)) {
                  // Set Codec property
                  hr = SetFilterFormat(GetAt(nCodec)->GetAt(nFormat)-
>m_pMediaType, pCodec);
                  // Retrieve control interfaces
                  IMediaControl* pMediaControl = NULL;
                  hr = m_pGraphBuilder->QueryInterface(IID_IMediaControl,
(void**) &pMediaControl);
                  if (SUCCEEDED(hr)) {
                        hr = pMediaControl->Run();
                        // start encoding
                        if (SUCCEEDED(hr)) {
                              long nCode = 0;
                              IMediaEvent *pMediaEvent = NULL;
                              m_pGraphBuilder-
>QueryInterface(IID_IMediaEvent, (void**) &pMediaEvent);
                              int nPercentComplete = 0;

                              // Wait until job complete
                              while (nCode != EC_COMPLETE) {
                                    pMediaEvent->WaitForCompletion(1000,
&nCode);
                                    // Report Progress
                                    CFile::GetStatus(szDestFileName,
fileStatus);
                                    CString szPercent;
                                    szPercent.Format("%d %%",
(fileStatus.m_size*100)/nOutputSize);
                                    AfxGetMainWnd()-
>SetWindowText(szPercent);
                              }
                              pMediaControl->Stop();
                              AfxGetMainWnd()-
>SetWindowText("Complete");
                              pMediaEvent->Release();
                        } else {
                              char szError[1024];
                              AMGetErrorText(hr, szError, 1024);
                              CString szDesc(szError);
```

```
                                AfxMessageBox(szDesc);
                        }

                        pMediaControl->Release();
                }
        }

        // Free interfaces
        pCodec->Release();
        pParser->Release();
        pMux->Release();
        pDest->Release();

        m_pGraphBuilder->Release();
        m_pGraphBuilder = NULL;
    }
}

// DSCodec.cpp: implementation of the CDSCodec class.
//////////////////////////////////////////////////////////////////////
#include "stdafx.h"
#include "DShowEncoder.h"
#include "DSCodec.h"
#ifdef _DEBUG
#undef THIS_FILE
static char THIS_FILE[]=__FILE__;
#define new DEBUG_NEW
#endif
//////////////////////////////////////////////////////////////////////
// Construction/Destruction
//////////////////////////////////////////////////////////////////////
CDSCodec::CDSCodec()
{
    m_pMoniker = NULL;
    m_szCodecName.Empty();
}
CDSCodec::~CDSCodec()
{
    int nNumberOfFormat = GetSize();
    for (int i=0; i<nNumberOfFormat; i++) {
        delete GetAt(i);
    }
    RemoveAll();

    if (m_pMoniker != NULL) {
```

```
            m_pMoniker->Release();
            m_pMoniker = NULL;
        }
}
void CDSCodec::BuildCodecFormatArray()
{
        if (m_pMoniker == NULL) return;

        HRESULT                 hr;
        IBaseFilter         *pBaseFilter = NULL;

        // Retrieve the IBaseFilter
        hr = m_pMoniker->BindToObject(NULL, NULL, IID_IBaseFilter, (void**)
&pBaseFilter);
        if (FAILED(hr)) {
            // ERROR HERE
            return;
        }
        // Enumerate Pin
        IEnumPins  *pEnumPins = NULL;
        hr = pBaseFilter->EnumPins(&pEnumPins);
        if (FAILED(hr)) {
            // ERROR HERE
            pBaseFilter->Release();
            return;
        }
        // Find the output Pin
        IPin   * pPin = NULL;
        while (pEnumPins->Next(1, &pPin, 0) == S_OK) {
            PIN_DIRECTION direction;
            pPin->QueryDirection(&direction);
            if (direction == PINDIR_OUTPUT) {
                // Retrieve the IAMStreamConfig
                IAMStreamConfig*pStreamConfig = NULL;
                hr = pPin->QueryInterface(IID_IAMStreamConfig, (void**)
&pStreamConfig);
                if (SUCCEEDED(hr)) {
                    int nCount = 0, nSize = 0;
                    pStreamConfig->GetNumberOfCapabilities(&nCount,
&nSize);
                    for (int i=0; i<nCount; i++) {
                        AM_MEDIA_TYPE* pMediaType = NULL;
                        AUDIO_STREAM_CONFIG_CAPS confCaps;
                        hr = pStreamConfig->GetStreamCaps(i,
&pMediaType, (BYTE*)&confCaps);
```

```
                                        if (SUCCEEDED(hr)) {
                                                CDSCodecFormat *pCodecFormat = new
CDSCodecFormat();
                                                pCodecFormat->m_pMediaType =
pMediaType;
                                                Add(pCodecFormat);
                                        }
                                }
                                pStreamConfig->Release();
                        }
                }
                pPin->Release();
        }
        pEnumPins->Release();
        pBaseFilter->Release();
}
```

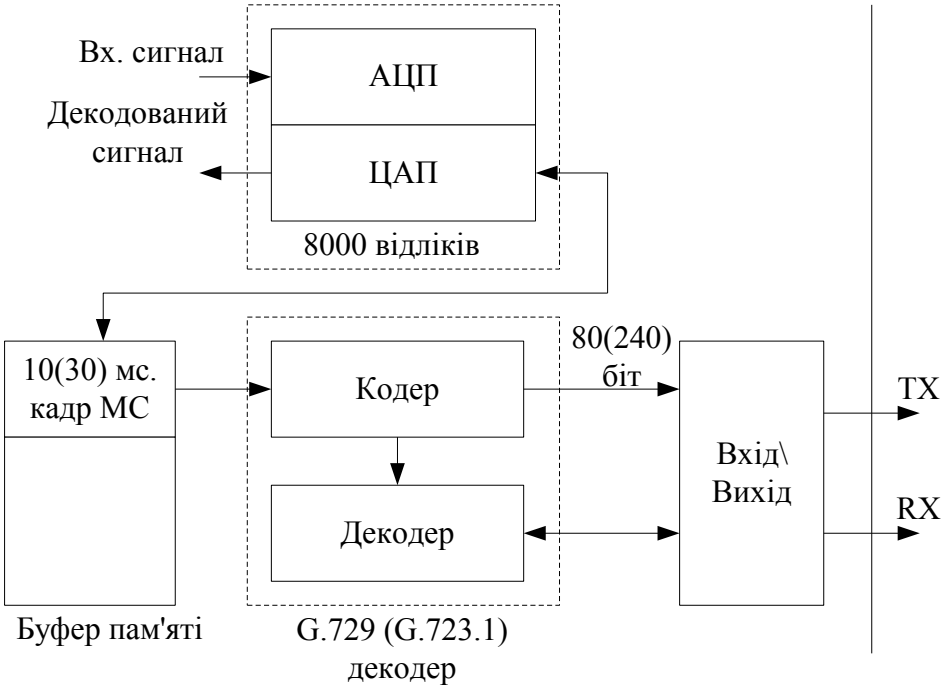
Код програмного модуля DSCodecFormat виглядає так:

```
// DSCodecFormat.cpp: implementation of the CDSCodecFormat class.
//////////////////////////////////////////////////////////////////////
#include "stdafx.h"
#include "DShowEncoder.h"
#include "DSCodecFormat.h"
#ifdef _DEBUG
#undef THIS_FILE
static char THIS_FILE[]=__FILE__;
#define new DEBUG_NEW
#endif
//////////////////////////////////////////////////////////////////////
// Construction/Destruction
//////////////////////////////////////////////////////////////////////
CDSCodecFormat::CDSCodecFormat()
{
        m_pMediaType = NULL;
}
CDSCodecFormat::~CDSCodecFormat()
{
        if (m_pMediaType != NULL) {
                // Free pMediaType
                if (m_pMediaType->cbFormat != 0) {
                        CoTaskMemFree((PVOID)m_pMediaType->pbFormat);

                        // Strictly unnecessary but tidier
                        m_pMediaType->cbFormat = 0;
```

```
                m_pMediaType->pbFormat = NULL;
            }
            if (m_pMediaType->pUnk != NULL) {
                m_pMediaType->pUnk->Release();
                m_pMediaType->pUnk = NULL;
            }
            CoTaskMemFree((PVOID)m_pMediaType);
        }
}
WORD CDSCodecFormat::NumberOfChannels()
{
        if (m_pMediaType != NULL) {
            WAVEFORMATEX* pFormat = (WAVEFORMATEX*)
m_pMediaType->pbFormat;
            return pFormat->nChannels;
        }
        return -1;
}
DWORD CDSCodecFormat::SamplesPerSecond()
{
        if (m_pMediaType != NULL) {
            WAVEFORMATEX* pFormat = (WAVEFORMATEX*)
m_pMediaType->pbFormat;
            return pFormat->nSamplesPerSec;
        }
        return -1;
}
DWORD CDSCodecFormat::BytesPerSec()
{
        if (m_pMediaType != NULL) {
            WAVEFORMATEX* pFormat = (WAVEFORMATEX*)
m_pMediaType->pbFormat;
            return pFormat->nAvgBytesPerSec;
        }
        return -1;
}
WORD CDSCodecFormat::BitsPerSample()
{
        if (m_pMediaType != NULL) {
            WAVEFORMATEX* pFormat = (WAVEFORMATEX*)
m_pMediaType->pbFormat;
            return pFormat->wBitsPerSample;
        }
        return -1;
}
```

Додаток Ж

Інтерфейс ядра багатоканальних процесорів алгоритмів ЛПАКСС та БКМД



Додаток И

Акти про впровадження результатів дисертаційної роботи

“ЗАТВЕРДЖУЮ”
Проректор з наукової роботи
Тернопільського
національного
економічного університету
д.е.н., проф. Мельник А.Ф.
_____

„____” _____ 200__
р.

АКТ
про використання результатів кандидатської дисертаційної роботи
Шевчука Руслана Петровича
„Багатоканальні комп’ютерні засоби перетворення форматів та
криптографічного захисту стиснених мовних сигналів”

Комісія у складі голови – завідувача кафедри комп’ютерних наук, керівника науково-дослідної роботи, д.т.н., проф. Дивака М.П. та членів: начальника науково-дослідної частини Письменного В.І. і начальника відділу прогнозування і маркетингу Лучка А.В. склали цей акт про те, що дослідження та результати дисертаційної роботи Шевчука Р.П. використані під час виконання науково-дослідних робіт на кафедрі комп’ютерних наук факультету комп’ютерних інформаційних технологій з безпосередньою участю автора, а саме:
- науково-дослідної роботи «Співпраця між Україною та Румунією в галузі розподілених систем (CobURDiS)» за другим етапом (03.2006-12.2006 р.) (номер державної реєстрації 0106U005307), у якій автором на основі розроблених принципів функціонування багатоканальних комп’ютерних засобів транскодування узагальнив особливості функціонування нестаціонарних розподілених об’єктів;
- науково-дослідної роботи “Розробка теоретичних засад, алгоритмічного та програмного забезпечення для моделювання технічних, екологічних та економічних систем на основі аналізу інтервальних даних” (номер державної реєстрації 0102U002565), у якій автором. розроблено методи для перетворення форматів стиснених мовних сигналів.

Голова комісії
завідувач кафедри комп’ютерних наук,
керівник НДР, д.т.н., проф. Дивак М.П.

Члени комісії:
начальник НДЧ Письменний В.І.

начальник відділу ПМ Лучка А.В.

„ЗАТВЕРДЖУЮ”
Перший проректор
Тернопільського національного
економічного університету
проф. Журавель Г.П.
_____
„____” _____ 200_ р.

АКТ

про впровадження в навчальний процес Тернопільського національного економічного університету результатів дисертаційної роботи Шевчука Руслана Петровича „Багатоканальні комп’ютерні засоби перетворення форматів та криптографічного захисту стиснених мовних сигналів”

Даний акт складений про те, що результати дисертаційної роботи аспіранта кафедри комп’ютерних наук Шевчука Руслана Петровича на тему „Багатоканальні комп’ютерні засоби перетворення форматів та криптографічного захисту стиснених мовних сигналів” використані в навчальному процесі факультету комп’ютерних інформаційних технологій Тернопільського національного економічного університету для студентів напряму підготовки 0804 – „Комп’ютерні науки” спеціальності 7.080403 „Програмне забезпечення автоматизованих систем”.

Впровадження результатів дисертаційної роботи полягає у наступному:

- розроблено методичні вказівки до проведення лекційних та лабораторних занять з дисципліни «Програмне забезпечення мультимедіа»;
- розроблено методичні вказівки до проведення лекційних занять з дисципліни «Методи та засоби захисту програмного забезпечення»
- розроблено методичні вказівки до проведення лабораторних занять з дисципліни «Методи та засоби вимірювання та цифрової обробки інформації»

Декан факультету комп’ютерних
інформаційних технологій,
зав. кафедри комп’ютерних наук,
д.т.н., проф. М.П. Дивак

Доцент кафедри комп’ютерних наук,
к.т.н. М.Я. Шпінталь

„ЗАТВЕРДЖУЮ”
Голова правління ВАТ ТКБР
„Стріла”
_____ О.О. Рафалюк
„___” _____ 2008 р.

АКТ

про впровадження результатів дисертаційної роботи
Шевчука Руслана Петровича
„Багатоканальні комп’ютерні засоби перетворення форматів та криптографічного захисту стиснених мовних сигналів”

Ми, комісія у складі: Піскун Сергій Олександрович, Карпів Володимир Богданович, Кульпа Михайло Володимирович склали даний акт про те, що у розробці та виробництві комп’ютерних систем зв’язку реального часу використано такі результати дисертаційної роботи аспіранта кафедри комп’ютерних наук Тернопільського національного економічного університету Шевчука Руслана Петровича:

1. Метод перетворення стиснених мовних сигналів між форматами G.723.1 та G.729A для підвищення ефективності використання каналів зв’язку між територіально віддаленими джерелами формування сигналів багатоканальних комп’ютерних систем реального часу.
2. Метод багатоступінчастого мікшування на базі пам’яті з довільним доступом для мікшування мовних сигналів по мірі їх поступлення у багатоканальні засоби комп’ютерних систем.
3. Удосконалені структури операційних пристроїв криптографічних модулів апаратно-орієнтованих процесорів підтримки протоколу IPSec для забезпечення захисту сигналів, що передаються системами зв’язку реального часу.

Перераховані результати дали змогу підвищити ефективність та захищеність передачі сигналів каналами зв’язку, а також зменшити результуючі затримки між територіально віддаленими джерелами багатоканальних комп’ютерних систем реального часу.

Зам. Голови правління _____ Карпів В.Б.

Головний конструктор _____ Піскун С.О.

Ведучий інженер _____ Кульпа М.В.

СПИСОК ЛІТЕРАТУРИ

1. Айфичер К. Цифровая обработка сигналов: практический поход / Айфичер К., Эммануил С., Барри У. – М. : Вильямс, 2004. – 922 с.
2. Бабкин В. В. Проблемы построения современных систем цифровой речевой связи / В.В. Бабкин // Труды 9-той международной конференции и выставки “Цифровая Обработка Сигналов и ее Применение DSPA-2007”. – 2007. – С. 121–124.
3. Бабкин В.В. LPC вокодер 1000-1200 бит/с / В.В. Бабкин // Труды 3-ей международной конференции и выставки “Цифровая Обработка Сигналов и ее Применение DSPA-2000”. – 2000. – С.50–58.
4. Барсуков В.С. Безопасность связи в каналах телекоммуникаций / Барсуков В.С., Дворянкин С.В., Шеремет И.А. // – М. : Электронные знания, 1993. – 122 с.
5. База речевых фрагментов русского языка “ISABASE” / Д.С. Богданов, О.Ф. Кривнова, А.Я. Подрабинович [и др.] // Интеллектуальные технологии ввода и вывода информации. – 1998. – С. 71–83.
6. Гольдштейн В.С. IP-Телефония / Гольдштейн В.С., Пинчук А.В., Суховицкий А.Л. – М. : Радио и Связь, 2001. – 336 с.
7. Гольдштейн А.Б., Гольдштейн Б. С. Softswitch / А.Б. Гольдштейн, Б. С. Гольдштейн. – СПб. : БХВ, 2006. – 368 с.
8. Гольдштейн Б. С. Протокол SIP. Справочник / Гольдштейн Б. С., Зарубин А.А., Саморезов В.В. – СПб. : БХВ, 2005. – 456 с.
9. Гольдштейн Б.С. Сигнализация в сетях связи : в 2 т. / Б.С. Гольдштейн. – СПб. : БХВ, 2005. – .– Т.2: Протоколы сети доступа. – 2005 – 448 с.
10. Грушвицкий Р.И. Проектирование систем на микросхемах программируемой логики / Грушвицкий Р.И. – СПб. : БХВ, 2002. – 608 с.
11. Дворянкин С.В. Компьютерные технологии защиты речевых

сообщений в каналах электросвязи / Дворянкин С.В. – М. : РИО МТУСИ, 1999. – 52 с.

12. Дегтеренко А.Н. Кодирование речевых сигналов на основе систем с переменой структурой: дис. маг.: 8.090803 / Дегтеренко Анатолий Николаевич. – Чернигов, 2004. – 118 с.
13. Задірака В.К., Олексюк О.С. Методи захисту фінансової інформації : Навчальний посібник / В.К Задірака., О.С Олексюк. – К.: Вища шк., 2000. – 460 с.
14. Иванова Н.Ю. Применение DSP для оптимизации защиты информации в сетях VoIPoW [Електронный ресурс] / Н.Ю. Иванова, С.А. Лысенков, А.Г. Солодов // Электронный научный журнал “Исследовано в России”. – Режим доступу к журн. : http://zhurnal.ape.relarn
15. Иванова Н.Ю. Реализация алгоритмов кодирования речевого сигнала на DSP / Иванова Н.Ю., Лысенков С.А., Солодов А.Г. – Самара. : СамГТУ, 2004. – 48 с.
16. Карпінський М.П. Система безпеки комп’ютерної мережі з використанням пристроїв Cisco IDS і PIX / Карпінський М.П // Вісник Тернопільського державного технічного університету. – 2006. – Т. 11, № 3. – С. 101-108.
17. Кестер У. Цифровая обработка сигналов / У. Кестер – California : California Technical Publishing, 1999. – 650 с.
18. Комаров А.В. Цифровые сигнальные процессоры / А.В. Комаров – Обнинск, 2003. – 214 с.
19. Концепція технічного захисту інформації в Україні / Кабінет Міністрів України. – Офіц. вид. – К. : Урядовий кур’єр, 1997. – № 1126. – (Бібліотека офіційних видань).
20. Коркішко Т. Базові структури операційних пристроїв хешування для процесорів підтримки протоколу IPSec / Т. Коркішко, Л. Коркішко, Р. Шевчук // Комп’ютинг. – 2003. – Т. 2, № 1. – С. 41–47.
21. Коркішко Т. Алгоритми та процесори симетричного блокового

шифрування / Коркішко Т., Мельник А., Мельник В. – Львів: БаК, 2003. – 168 с.

22. Коркішко Т. Аналіз архітектур багатоабонентських мультимедіа-конференцій / Т. Коркішко, Р. Шевчук // Матеріали третьої міжнародної науково-технічної конференції “Проблеми інформатики і моделювання”. – Харків. – 2003. – С. 6.
23. Коркішко Т. Синтез структур операційних пристроїв виконання криптографічних алгоритмів IPSEC оптимізованих для обробки медіа пакетів / Т. Коркішко, Р. Шевчук // Комп’ютинг. – 2004. – Т. 3, № 3. – С. 100–109.
24. Коркішко Т. Часові характеристики паралельних багатоабонентських мультимедіа конференцій рекурсивної архітектури / Т. Коркішко, Р. Шевчук // Вісник Тернопільського державного технічного університету. – 2004. – № 2. – С. 109–116.
25. Коркішко Т.А. Багатоканальні апаратно-орієнтовані процесори симетричного блокового шифрування : дис. ... канд.тех.наук : 05.13.05 / Коркішко Тимур Анатолійович. – Львів, 2002. – 213 с.
26. Коркішко Т.А. Методика проектування багатоканальних процесорів симетричного блокового шифрування / Т.А. Коркішко, А.О. Мельник // Вісник Тернопільського державного технічного університету. – Тернопіль. – 2002. – Т. 7, № 2. – С. 100–109.
27. Круг П.Г. Процессоры цифровой обработки сигналов: Учебное пособие / Круг П.Г. – М. : МЭИ, 2001. – 128 с.
28. Маркел Дж. Линейное предсказание речи (Пер. с англ. Под ред. Ю.Н. Прохорова и В.С. Звездина) / Маркел Дж., Грэй А.Х. – М. : Связь, 1980. – 308 с.
29. Мельник А. Порівняльний аналіз алгоритмів стиснення мовних сигналів / А. Мельник, Р. Шевчук // Вісник національного університету “Львівська політехніка” Комп’ютерні системи і мережі. – 2004. № 523. – С. 109–117.

30. Мельник А. Особливості багатоканального транскодування форматів стиснених мовних сигналів / А. Мельник, Р.Шевчук // Вісник Тернопільського державного технічного університету. – 2005. – № 2. – С. 122–128.
31. Мельник А.О. Мікшування мовних сигналів у мультимедійних системах реального часу / А.О. Мельник, Р.П. Шевчук, Т.А. Коркішко // Комп'ютинг. – 2006. – Т.5, № 1. – С. 57–65.
32. Мельник А.А. Процессоры обработки сигналов / Мельник А.А. – Львов, 1989. – 63 с. – (Препринт / АН УССР. Ин-т прикл. Проблем механики и математики ; №29-89).
33. Мельник А.О. Спеціалізовані комп'ютерні системи реального часу / А. О. Мельник. – Львів, 1996. – 54 с.
34. Музыченко Е. Часто задаваемые вопросы по цифровому представлению звуковых сигналов [Электронный ресурс] / Е. Музыченко // neFormat – 1999. – № 3. – Режим доступа к журн.: http://dj.townnet.ru/magazine/
35. Николайчук Я.М. Захист даних в функціонально орієнтованих мережах зв'язку / Николайчук Я.М., Лях І.М., Притуляк Я.Г. // Вісник Хмельницького національного університету. – 2005. – Т. 1, № 4. –С. 259–263.
36. Пат. 5703794 США, МКИ G 06 K 15. Method and system for mixing audio streams in a computing systems: Пат. 5703794 США, МКИ G 06 K 15 Heddle R.M (США); Microsoft Corporation. № 492709, Заявл. 20.06.95, Опубл. 30.12.97, НКИ 364/512 R. – 20 с.
37. Пат. 7096181, МКИ G 10 L 19/14. Method for searching codebook: Пат. 7096181 США, МКИ G 10 L 19/14 Jung, Sung Kyo (Seoul, KR); LG Electronics Inc. № 277874, Заявл. 23.10.2002, Опубл. 22.08.2006. – 12 с.
38. Передача речи по трактам радиотелефонной связи. Требования к разборчивости речи и методы артикуляционных измерений : ГОСТ 16600-72. – [Дата документа: 1974-01-01]. – М. : Госстандарт СССР,

1973. – 93 с.

39. Петраков А.В. Утечка и защита информации в телефонных каналах / Петраков А.В., Лагутин В.С. – М. : Энергоатомиздат, 1998. – 317 с.
40. Прикладные решения Avaya. Руководство по развертыванию IP-телефонии. – 2004. – 474 с.
41. Прокис Дж. Цифровая связь (Пер с англ. Под ред. Д.Д Кловского) / Прокис Дж. – М. : Радио и связь, 2000. – 800 с.
42. Рабинер Л.Р. Цифровая обработка речевых сигналов / Рабинер Л.Р., Шафер Р.В. – М. : Радио и связь, 1981. – 784 с.
43. Росляков А.В. IP-телефония / Росляков А.В., Самсонов М. Ю., Шибаева И.В. – М. : Эко-Трендз, 2003. – 252 с.
44. Сапожков М.А. Вокодерная Связь / Сапожков М.А., Михайлов В.Г. –М. : Радио и Связь, 1983. – 398 с.
45. Секунов Н.Ю. Обработка звука на PC / Секунов Н.Ю. – СПб.: БХВ-Петербург, 2001. – 1248 с.
46. Семенов Ю. А. Сети Интернет. Архитектура и протоколы / Семенов Ю. А. – М. : Сиринъ, 1998. – 424 с.
47. Сергеев А. П. Программирование в Microsoft Visual C++ 2005. Самоучитель / Сергеев А. П., Терен А. Н. – М. : Издательский дом “Вильямс”, 2006. – 352 с.
48. Сергиенко А.Б. Цифровая обработка сигналов / Сергиенко А.Б. – СПб. : Питер, 2003. – 604 с.
49. Симоненков Д. Компрессия звуковых данных [Электронный ресурс] / Д. Симоненков // Компьютерра. – 1998. – № 32. – Режим доступа к журн: http://offline.computerra.ru/1998/260/1494/
50. Скляр Б. Цифровая связь. Теоретические основы и практическое применение. Изд. 2-е, испр / Б. Скляр. – М. : Издательский дом “Вильямс”, 2003. – 1004 с.
51. Солонина А.И. Алгоритмы и процессоры цифровой обработки сигналов / Солонина А.И., Уласович Д.А., Яковлєв Л.А. – СПб. : БХВ-

Петербург, 2002. – 464 с.

52. Фланаган Дж. Анализ, синтез и восприятие речи (Пер. с англ. Под ред. А.А. Пирогова) / Дж Фланаган. – М. : Связь, 1968. – 395 с.
53. Фролов А.В. Мультимедиа для Windows / А.В. Фролов. – М. : Диалог-МИФИ, 1995. – 284 с.
54. Хелд Г. Сокращение задержки голоса по IP [Электронный ресурс] / Г. Хелд // LAN – 2000. – № 07. – Режим доступа к журн.: http://www.osp.ru
55. Хорошко В. А. Методы и средства защиты информации / Хорошко В. А., Чекатков А. А. – К. : Юниор, 2003. – 504 с.
56. Хортон А. Microsoft Visual C++ 2005: базовый курс / А. Хортон. – М. : Диалектика, 2007. – 1152 с.
57. Черкаський М. Складність апаратно-програмних комп'ютерних засобів / М. Черкаський // Сучасні проблеми в комп'ютерних науках. – 2000. С. 58-67.
58. Шевчук Р.П. Оптимізація програмно-апаратних засобів реалізації IPSec: маг. роб. : 8.091501 / Шевчук Руслан Петрович. – Тернопіль, 2003. – 121 с.
59. Шевчук Р.П. Транскодування стиснених мовних сигналів між GSM 06.20 та G.729 / Р.П. Шевчук // Міжнародний науково-технічний журнал “Інформаційні технології та комп’ютерна інженерія” – 2007. – № 3. – С. 172–179.
60. Шевчук Р.П. Проектування багатоканального транскодера між G.723.1 та G.729A / Р.П. Шевчук, Л.І. Гончар // Вимірювальна та обчислювальна техніка в технологічних процесах. – 2007. – № 2. – С.124–129.
61. Шевчук Р. Особливості організації мультимедіа конференцій / Р. Шевчук, А. Чвиль // Матеріали VIII наукової конференції ТДТУ імені Івана Пулюя. – Тернопіль. – 2004. – С. 88.
62. Advanced WMA Workshop 2.1 [Electronic resources]. – Режим доступу :

http://www.magicvideo.net.

63. All To MP3 Converter 2.1 [Electronic resources]. – Режим доступу : http://www.wma-mp3.com.
64. Appelblom M. An Advanced Speech Codec For VoIP. Group Green [Electronic resources] / M. Appelblom, C. Omurcali C, S. Isoard [and other]. – 2003. – Режим доступу : http://www.s3.kth.se/kurser/2E1366/ students/03/green/download/
65. AudioCodes. User Manual and Library Reference [Electronic resources]. – 2005. – Режим доступу : www.audiocodes.com/
66. Benvenuto N. Algorithms for communication systems and their applications / Benvenuto N., Cherubini G. – England: John Wiley & Sons Ltd, 2002. – 1305 p.
67. Berkeley A. Choosing a DSP Processor / A. Berkeley. – California: Berkeley Design Technology Inc, 2000. – 8 p.
68. Bernard A.P. Source-channel coding of speech: PhD thesis / A.P. Bernard. – University of California, 1998. – 78 p.
69. Campos Neto A.F. Performance assessment of tandem connection of enhanced cellular coders / Campos Neto A.F., Crcoran F.L. // IEEE Proceedings of International Conference on Acoustics Speech Signal Processing. – 1999. P. 177–180.
70. CCITT Recommendation G.721. 32kb/s Adaptive Differential Pulse Code Modulation (ADPCM) / Blue Book, Vol. III. – Fascicle III.3. – 1988. Режим доступу: http://www.itu.int/rec/T-REC-G.721/e.
71. CCITT Recommendation G.723. Extensions of Recommendation G.721 ADPCM to 24 and 40 kbits/s for DCME Application [Electronic resources] / Blue Book, Vol. III. – Fascicle III. – 1988. Режим доступу: http://www.itu.int/rec/T-REC-G.723/e.
72. Chen J-H. A real-time full duplex 16/8 kbps CVSELP coder with integral echo canceler implemented on a single DSP56001 / Chen J-H., Danisewicz R.G., Kline R.B. [and other] // Advances in Speech Coding. – Kluwer,

Academic Publishers, 1990. – P. 32–41.

73. Choi Y. S. A Very Low Complexity VSELP Speech Coder Using Regular Pulse Basis Vectors / Choi Y. S., Hong-Goo, Kang J. H. [and other] // IEICE Trans. Fund. Elec. Comm. Comp. Sci. – 1997. – Vol. E80–A, № 6. –P. 996–1001.
74. Chu W.C. Speech coding algorithms Foundation and Evolution of Standardized Coders / W.C. Chu. – New Jersey : John Wiley & Sons Inc, 2003. – 578 p.
75. Cisco CallManager System Guide. Release 3.3(2) [Electronic resources]. – 2002. Режим доступу: http://www.cisco.com/application/pdf/en/us/guest/products/ps4153/c1696/ccmigration_09186a008011b480.pdf
76. Cox R.V. New Directions in Sub-band Coding / R.V. Cox // IEEE Trans. On Selected Areas in Communications, Special Issue on Voice Coding for Communications. –1988. – Vol. 6. – № 2. – P. 391–409.
77. Devalapalli S.K. Design of a CELP Speech Coder and Study of Complexity vs Quality Trade-offs for different Codebooks [Electronic resources] / Devalapalli S.K., Rangarajan R., Venkataramanan R. – Режим доступу: www-personal.umich.edu/~rvenkata/
78. Ercelens J.S. LPC interpolation by approximation of the sample of autocorrelation function / Ercelens J.S., Broersen P.M // Processing of IEEE Trans. Acoustic Speech Signal Process. – 1998. – Vol. 6. – № 6. – P.569–573.
79. European Telecommunication Standard. Digital Cellular Telecommunication System (Phase 2+). Half Rate Speech GSM 06.20 Version 5.1.1 [Electronic resources]. – May 1998. – Режим доступу: http://www.mobilein.com/gsm_standards.htm.
80. Foster A. B. Media Gateway Control Protocol (MGCP) Version 1.0 F. RFC3435 [Electronic resources] / A. B. Foster. – 2003. – Режим доступу: www.ietf.org/rfc/
81. Gersho A. Recent Trends and Techniques in Speech Coding / Gersho A.,

Wang S. // Processing 24-th Asilomar Conf. Circuits, Systems, and Computers. – 1990. – P. 634-638.

82. González A.J. Audio mixing for interactive multimedia communications / González A.J., Hussein A.W // Processing of the JCIS'98. – 1998. – P. 217–220.
83. Griffin D. Multiband Excitation Vocoder / Griffin D., Lim J // IEEE Trans. ASSP-36. – 1988. – № 8. – P. 1223–1235.
84. Hardwick J.C. The application of the IMBE speech coder to mobile communications / Hardwick J.C., Lim J.S // Processing of the IEEE International Conference on Acoustics, Speech, and Signal (ICASSP). – 1991. – P. 249-252.
85. Heath S. Multimedia & Communications Technology / S. Heath. – 1996. – 294 p.
86. Hersent O. IP Telephony Packet based Multimedia Communications Systems / Hersent O., Gurle D., Petit J. – Harlow : Addison Wesley, 2000. – 480 p.
87. Micom. User Manual and Library Reference [Electronic resources]. –Режим доступу : http://www.micom.com
88. Intelligent Voice Transcoding Technology [Electronic resources]. – Режим доступу http://www.dilithiumnetworks.com
89. International Telecommunication Union. Packet based multimedia communication systems. Recommendation H.323 [Electronic resources] / Telecommunication Standardization Sector of ITU. – 1998. – Режим доступу: http://dret.net/biblio/reference/h323
90. ITU P.862 (2000). Perceptual evaluation of speech quality (PESQ), and objective method for end-to-end speech quality assessment of narrowband telephone networks and speech codecs [Electronic resources] / ITU-T Recommendation P. 862. – Режим доступу: http://www.itu.int/
91. ITU Recommendation G.726. 40, 32, 24, 16 kbit/s Adaptive Differential Pulse Code Modulation [Electronic resources]. – 1991. – Режим доступу:

http://www.itu.int/rec/T-REC-G.726/e

92. ITU Recommendation G.728. Coding of Speech at 16 kbit/s Using Low-Delay Code Excited Linear Prediction [Electronic resources]. – 1994. – Режим доступу: http://www.itu.int/rec/T-REC-G.728/e
93. ITU-T H.323: Packet-based multimedia communications systems [Electronic resources]. – 1999. – Режим доступу: http://www.imtc.org/
94. ITU-T Recommendation G.722 (SB-ADPCM) 7 kHz audio-coding within 64 kbit/s [Electronic resources]. – 1988. – Режим доступу: http://www.itu.int/rec/T-REC-G.722/e.
95. ITU-T Recommendation G.723.1. Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kb/s. – 1996. – Режим доступу: http://www.itu.int/rec/T-REC-G.723.1/e
96. ITU-T Recommendation G.729. Coding of speech at 8 kbit/s using conjugate-structure algebraic code-excited linear prediction (CS-ACELP). – 1996. – Режим доступу: http://www.itu.int/rec/T-REC-G.729/e
97. Jayant N. Digital Coding of Waveforms: Principles and Applications to Speech and Video. Englewood Cliffs / Jayant N., Noll P. – NJ: Prentice-Hall, 1990. – 688 p.
98. Jayant N.S. Digital Coding of Speech Waveforms: PCM, DPCM, and DM Quantizers / N.S. Jayant // Processing IEEE. –1974 – Vol. 62. – P. 611-632.
99. Jeff Bier. Choosing a Processor: Benchmarks and BeyondBerkeley [Electronic resources] / Bier Jeff. – California: Berkeley Design Technology Inc, 2006. – Режим доступу: http://www.bdti.com/articles/
100. Kang H.G. Improving transcoding capability of speech coders in clean and frame erasured channel environments / Kang H.G., Hong-Kook K., Cox R.V. // Processing of IEEE Workshop on Speech Coding. – 2000. – P. 78–80.
101. Kent S., Atkinson R. IP Encapsulating Security Payload [Electronic resources] / Kent S., Atkinson R. – 1998. – Режим доступу: http://www.ietf.org/rfc/rfc2406.txt
102. Kent S., Atkinson R. IP Authentication Header [Electronic resources] / Kent

S., Atkinson R. – 1998. – Режим доступу: http://www.ietf.org/rfc/ rfc2402.txt

103. Kent S. Security Architecture for the Internet Protocol [Electronic resources] / Kent S., Atkinson R. – 1998. – Режим доступу: http://www.ietf.org/rfc/rfc4301.txt
104. Kroon P. On the use of pitch predictors with high temporal resolution / Kroon P., Atal B.S. // Processing of IEEE Trans. Signal Process. – 1991. – Vol. 39, № 3. – P. 733-735.
105. Korkishko T. Investigation of the characteristics of recursive architecture for multipoint parallel multimedia conferences / T. Korkishko, R. Shevchuk // Proc. of the Intern. Conf. “Modern Problems of Radio Engineering, Telecommunications, and Computer Science” (TCSET’2004). – Lviv-Slavsko, 2004. – P. 388–390.
106. Lan Juan. An 8-kb/s Conjugate-Structure Algebraic CELP (CS-ACELP) Speech Coding / Lan Juan, Lin Biqin, Fu Qiuliang // Processing of ICSP. – 1998.
107. Lee Sunil. Novel tandemless transcoding algorithm for AMR and EVRC speech coders: MS thesis / Sunil Lee. – Korea Advanced Institute of Science and Technology. – 2002. – 65 p.
108. Li W. Comparison of Speech Coding Algorithms: ADPCM, CELP and VSELP / Li W., Sridhar A., Teng T // Project for EE 6390. – Fall, 1999. – 12 p.
109. Macres J. Theory and Implementation of the Digital Cellular Standard Voice Coder: VSELP on the TMS320C5x. Application Report, Texas Instruments SPRA136 / J. Macres. – 1994.
110. Mark D. Grosen. Implementation of a CELP Speech Coder for the TMS320C30 using SPOX [Electronic resources] / D. Mark. – Режим доступу: http://www.ietf.org/rfc/rfc4301.txt
111. Melnik A. Method of multistage mixing speech signals for the real-time multimedia systems / A. Melnik, T. Korkishko, R. Shevchuk // Proc. of the

Intern. Workshop “Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications”. – Sofia, Bulgaria, 2005. – P. 653–656.

112. Melnik A. Transcoding of Formats of Compressed Speech Signals / A. Melnik, R. Shevchuk // Proc. of the 8-th Intern. Conf. Proc. of the Intern. Conf. “Experience of Designing and Application of CAD Systems in Microelectronics” (CADSM’2005). – Lviv-Polyana, 2005. – P. 151–153.
113. Melnik A. Multichannel mixing of speech signals accordant with the method of multistage mixing / A. Melnik, R. Shevchuk, H. Sapozhnyk // Proc. of the Intern. Conf. “Modern Problems of Radio Engineering, Telecommunications and Computer Science” (TCSET’2006). – Lviv-Slavsko, 2006. – P. 169–172.
114. Mouly M. The GSM System for Mobile Communications / Mouly M., Pautet M. – Palaiseau, France: Cell & Sys Publishers, 1992. – 702 p.
115. MP3 Encoder [Electronic resources]. – Режим доступу: http://www.rmbsoft.com.
116. MP3 RM Converter [Electronic resources]. – Режим доступу: www.audiotoolsfactory.com.
117. Olausson M. Hardware for speech and audio coding: PhD thesis / M. Olausson. – University of Linkopings. – 2004. – 115 p.
118. Olausson M. Instruction and Hardware Accelerations in G.723.1 (6.3/5.3) and G.729 / Olausson M., Liu D. // Proceedings of the 1st IEEE International Symposium on Signal Processing and Information Technology (ISSPIT). – 2001. – P. 34-39
119. Overview and Applications of Voice Transcoding [Electronic resources]. – Режим доступу http://www.dilithiumnetworks.com.
120. Padjen R. Cisco AVVID IP Telephony and Design & Implementation / Padjen R., Thurston S., Flannagan M. – Rockland: Syngress Publishing Inc, 2001. – 608 p.
121. Pranata A. Development of network service infrastructure for transcoding

multimedia stream: MS Thesis / A. Pranata. – University of Stuttgart. – 2002. – 103 p.

122. Product overview ComStruct GSM-ARM transcoder. – Motorola Inc, 2001. – 2 p.
123. Product overview ComStruct PMC transcoder. – Motorola Inc, 2001. – 2 p.
124. Radenkovic M. Scaleable Audio for Collaborative Environments: PhD thesis / M. Radenkovic. – University of Nottingham. – 2002. – 196 p.
125. Radenkovic M. Multi-party Distributed Audio Service with TCP Fairness / Radenkovic M., Greenhalgh C // Proceedings of the Sixth conference of the UK_VRSIG. – 1999. – P. 11–20.
126. Rangan P. V. Optimal Communication Architectures for Multimedia Conferencing in Distributed Systems / Rangan P. V., Harrick M. // ICDCS. – 1992. – P. 46–53.
127. Rangan P. V. Communication Architectures and Algorithms for Media Mixing in Multimedia Conferences / Rangan P. V., Harrick M., Ramanathan V. S. // IEEE/ACM Transactions on Networking. – 1993. – Vol. 1. – № 1. – P. 20-30.
128. Rangan P. V. Hierarchical Conferencing Architectures for Inter-Group Multimedia Collaboration / Rangan P. V., Harrick M., Ramanathan V. S. // Proceedings of the Conference on Organizational Computing Systems (COCS'91). – 1991. – Vol. 12. – № 2–3. – P. 43–55.
129. Recommendation G.711. Pulse code modulation (PCM) of voice frequencies [Electronic resources]. – 1988. Режим доступу: http://www.itu.int/rec/T-REC-G.711/e
130. Rocchesso D. Introduction to Sound Processing / D. Rocchesso. – Firenze: Mondo Estremo Publishing, 2003. – 246 p.
131. Salami R. ITU-T Recommendation G.729 Annex A: reduced complexity 8 kbit/s CS-ACELP codec for digital simultaneous voice and data / R. Salami // IEEE Communications Magazine. – 1997 – Vol. 35 – № 9. – P. 56–63.
132. Salomon D. Data compression, 3rd Edition / D. Salomon. – New York :

Springer-Verlag New York Inc, 2004. – 920 p.

133. Schroeder M.R. Code-Excited Linear Prediction (CELP): High Quality Speech at Very Low Bit Rates / Schroeder M.R., Atal B. // Processing ICASSP-85. – Tampa, 1985. – P. 937–940.
134. Schulzrinne H. RTP: A Transport Protocol for Real-Time Applications. IETF RFC 1889 / Schulzrinne H., Casner S., Frederick R. – 1996.
135. Shevchuk R.P. Method of converting speech codec formats between GSM 06.20 and G.729 / R. Shevchuk, L. Honchar, P. Bykovyy // Proc. of the 4-th IEEE Workshop "Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications". – Dortmund, Germany, 2007. – P. 686–689.
136. Shevchuk R. Method of converting speech codec formats between G.723.1 and G.729A / R. Shevchuk // Proc. of the Intern. Conf. "Experience of Designing and Application of CAD Systems in Microelectronics" (CADSM'2007). – Lviv-Polyana, 2007. – P. 483–486.
137. Sze Ming Cheng. Coding with side information: Doctor Thesis / Sze Ming Cheng. – University of Texas. – 2004. – 137 p.
138. Singhal S. High quality audio coding using multipulse LPC / S. Singhal // Proc. ICASSP. – 1990. – P.1101–1104.
139. Smith W. S. The scientist and engineers guide to Digital Signal Processing, Second Edition / W. S. Smith. – San Diego: California Technical Publishing, 1999. – 643 p.
140. Soong F. K. Line Spectrum Pair (LSP) and Speech Data Compression / Soong, F. K., Juang B // IEEE ICASSP. – 1984. – P. 1101–1104.
141. Spanias A. S. Speech Coding: a tutorial review / A. S. Spanias // Proceeding of the IEEE. – 1994. – Vol 82, № 10. – P. 3071-3075.
142. Study Group 15. Security and Encryption for H-Series (H.323 and other H.245-Based) Multimedia Terminals [Electronic resources]. – 2000. – Режим доступу: http://www.networkdictionary.com/node/951
143. Supplee L.M. MELP: The new federal standard at 2400 bps / Supplee L.M.,

McCree A.V. // Processing of the IEEE International Conference on Acoustics, Speech, and Signal. – 1997. – P. 1591–1594.

144. Switch Audio File Conversion Software [Electronic resources]. – Режим доступу: http://www.nch.com.au/switch
145. The LAME Open Source MP3 Encoder [Electronic resources]. – Режим доступу: http://www.mp3dev.org
146. TMS320C6201 Data Sheet. Texas Instruments. SPRS051 [Electronic resources]. – 1998. – Режим доступу: www.datasheetarchive.com/
147. TMS320C6201/C6701 Peripherals Reference Guide. SPRU190. – 1998. – Режим доступу: www.focus.ti.com/lit/ug/spru269f/spru269f.pdf
148. TMS320C6x Evaluation Module Reference Guide. SPRU269. – 1998. – Режим доступу: www.focus.ti.com/lit/ug/spru188d/spru188d.pdf
149. Trancoso I.M. Efficient search procedure for selection the optimum innovation in stochastic coders / I.M. Trancoso // Processing of IEEE Trans. Acoustic Speech Signal Process. – 1990. – Vol. 38. – №3. – P.385–396
150. Trancoso I.M. Efficient search procedure for selection the optimum innovation in stochastic coders / Trancoso I.M., Atal B.S // Processing of IEEE Trans. Acoustic Speech Signal Process. – 1990. – Vol. 38, № 3. – P. 385–396.
151. Un C. K. The Residual-Excited Linear Prediction Vocoder with Transmission Rate below 9.6 kbits/s / Un C. K., Magill D.T // IEEE Trans. – 1975. – P. 1466–1474.
152. Wave-conferencing manual [Electronic resources]. – Режим доступу: www.wave-conferencing.com/seminarb/
153. Windows XP Profesional Product Documentaton. Using Sound Recorder. – Режим доступу: www.support.microsoft.com
154. Zelinski R. Adaptive Transform Coding of Speech Signals / R. Zelinski // IEEE Transactions on Acoustics, Speech and Signal Processing. – 1997. – Vol. ASSP-25. – № 4. – P. 299–309