

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Сумський державний університет

Кафедра електроніки і комп'ютерної техніки

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи магістра на тему:

«Система передачі інформації на базі кодів Шеннона-Фано»

Завідувач кафедрою

А.С. Опанасюк

Керівник роботи

О.В. Бережна

Консультант з

техніко-економічної частини

О.М. Маценко

Проектував студент

О.М. Трохименко

Суми
2021 р.

					ЦЗДВН 8.171.00.10.327 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		0

Факультет ЦЗДВН Кафедра Електроніки і КТ
Спеціальність Електронні системи та компоненти

ЗАТВЕРДЖУЮ:

Зав. кафедри Опанасюк А.С.

« » _____ 20 г.

ЗАВДАННЯ

ДО КВАЛІФІКАЦІЙНОЇ РОБОТИ МАГІСТРА СТУДЕНТУ

Трохименко Олександр Михайлович

1. Тема роботи «Система передачі інформації на базі кодів Шеннона-Фано»

затверджено наказом від «01» листопада 2021 р. № 0701-VI

2. Термін здачі студентом закінченого проекту 15.12.21

3. Вихідні дані до роботи Розробити пристрій, який буде задовольняти наступним вимогам: вихідний код – ASCII; вихідний алфавіт – нерівномірний; метод стиснення даних – Шеннона-Фано.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які підлягають розробці) Вступ. Огляд літератури та постановка завдання. Науково-дослідна частина. Розробка алгоритму функціонування та структурної схеми системи. Розробка схеми електричної функціональної пристрою. Розробка та розрахунок принципів електричних схем вузлів та блоків пристрою. Техніко-економічна частина. Висновки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

1 Схеми алгоритму .

2 Схеми електрична структурна .

3Схеми електрична функціональна .

4Схеми електрична принципова .

6. Консультанти до проекту (роботи), з зазначенням розділів проекту, що до них відносяться

Розділ	Консультант	Підпис, дата	
		Завдання надав	Завдання прийняв
Економіка	Маценко О.М.		

7. Дата надання завдання _____

Керівник _____
(підпис)

Завдання прийняв до виконання _____
(підпис)

Календарний план

№ п/п	Найменування етапів дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1.	Огляд технічної літератури	05.11.21	
2.	Науково-дослідна частина	15.11.21	
3.	Розробка алгоритму функціонування та структурної схеми	20.11.21	
4.	Розробка схеми електричної функціональної	25.11.21	
5.	Розробка схеми електричної принципової	30.11.21	
6.	Розробка техніко-економічної частини	05.12.21	
7.	Оформлення графічної частини	10.12.21	
8.	Оформлення пояснювальної записки	15.12.21	
9.	Рецензування та підготовка до захисту	20.12.21	

Студент-дипломник _____

Керівник проекту _____

РЕФЕРАТ

Пояснювальна записка містить: 88 аркушів, 18 рисунків, 26 таблиць, 18 джерел літератури.

Графічна частина роботи містить: схему алгоритму роботи пристрою стиснення даних на базі кодів Шеннона-Фано, схему електричну структурну пристрою, схему електричну функціональну та схему електричну принципову.

Пояснювальна записка містить два розділи: основну частину та техніко-економічну частину.

У першому розділі розглянуто систему передачі даних, методи кодування інформації, код Шеннона – Фано. Досліджено характеристики коду Шеннона-Фано та Хаффмана. Розроблено схему електричну структурну пристрою та схему алгоритму, функціональну та принципову електричні схеми, розглянуто блоки, з яких складається пристрій.

У другому техніко-економічному розділі було розглянуто питання про соціально-екологічні проблеми інформаційного суспільства, а також розраховано виробничу собівартість, повну собівартість, оптову ціну, роздрібну ціну пристрою стиснення даних.

					ЦЗДВН 8.171.00.10.327 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		3

ЗМІСТ

	С.
Вступ	5
1. Основна частина	7
1.1 Огляд літератури та постановка задачі проектування	7
1.2 Науково-дослідна частина	22
1.2.1 Характеристики ефективних кодів	22
1.2.2 Коди Шеннона - Фано	25
1.2.3 Коди Хаффмана	29
1.2.4 Оцінка ефективності коду Шеннона - Фано	34
1.3 Розроблення електронного пристрою з використанням результатів дослідження	48
1.3.1 Обґрунтування алгоритму функціонування та структурної схеми проектованого пристрою	48
1.3.2 Розроблення схеми електричної функціональної проектваного пристрою	53
1.3.3 Розроблення та розрахунок принципів електричних схем вузлів та блоків пристрою	56
1.3.4 Розроблення програмного забезпечення проектваного пристрою	66
2. Техніко-економічна частина	71
2.1 Соціально-екологічні проблеми інформаційного суспільства	71
2.2 Розрахунок собівартості та оптової ціни пристрою, що розроблюється	77
2.2.1 Розрахунок повної собівартості проектованого пристрою	77
2.2.2 Розрахунок ціни пристрою	83

ЦЗДВН 8.171.00.10.327 ПЗ				
Изм.	Лист	№ докум.	Подпись	Дата
Разраб.		Трохименко О.М.		
Провер.		Бережна О.В.		
Реценз.				
Н. Контр.		Гапич В.М.		
Утверд.		Опеласюк А.С.		
Изм.	Лист	№ докум.	Подпись	Дата
ЦЗДВН 8.171.00.10.327 ПЗ				
Система передачі інформації на базі кодів Шеннона-Фано			Лит.	Лист
			3	88
Пояснювальна записка				Лист
ЦЗДВН 8.171.00.10.327 ПЗ				4
СумДУ, ЕС.мдн – 0				лп

Висновки	85
Список літератури	87
Додаток А	

						ЦЗДВН 8:171.00:10:327 ПЗ	Лист Лист
Изм.	Лист	№ докум.	Подпись	Дата			4
Изм.	Лист	№ докум.	Подпись	Дата			5

ВСТУП

Діяльність людей пов'язана з переробкою і використанням матеріалів, енергії та інформації. Відповідно розвивалися наукові технічні дисципліни, які відбивають питання технології, енергетики та інформатики.

Наука про інформацію тепер знаходить застосування в найрізноманітніших областях теорії та практики. Центральною областю застосування є теорія інформації та теорія зв'язку, що створені К. Шенноном на основі теорії ймовірностей [1].

З передачею та стисненням інформації пов'язана робота будь-якого автоматичного пристрою, обіг живої істоти, творча діяльність людини, розвиток науки та техніки, економічні та соціальні перетворення в суспільстві й в житті. Для більш ефективного використання інформації необхідно обмінюватися інформацією, що неможливо без її передачі по каналах зв'язку та без її стиснення. Стиснення скорочує обсяг простору, який є необхідним для зберігання файлів в електронної обчислювальної машині (ЕОМ), та кількість часу, необхідний для передачі інформації по каналу зв'язку встановленої ширини пропускання. Це є форма кодування. Другими цілями кодування є пошук та виправлення помилок в отриманих повідомленнях, а також шифрування інформації. Процес пошуку та виправлення помилок протилежний стисненню. Такий процес збільшує надмірність даних. Видаляючи з тексту надмірність, стиснення сприяє шифруванню інформації, що утрудняє пошук шифру доступним для несанкціонованого користувача статистичним методом [2].

У багатьох випадках збережена і передана інформація може являти інтерес для осіб, які бажають використовувати її в корисних цілях.

Одним з методів захисту інформації є кодування повідомлень.

Кодування - це відображення повідомлень кодовими комбінаціями за певним правилом присвоєння символів. Код - це правило, що описує відображення одного набору знаків в інший набір знаків (або слів). Кодом також називають й множину образів при цьому відображенні.

Двійковий код на всі можливі комбінації не є завадозахищеним, тому що його кодові комбінації відрізняються одна від одної лише в одному розряді, що не дозволяє на приймальному боці каналу зв'язку виявляти та виправляти помилки, що виникли у процесі передачі кодових комбінацій по каналу зв'язку. У зв'язку з цим виникає необхідність побудови завадозахищеного коду [1, 2].

					ЦЗДВН 8.171.00.10.327 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		6

Оптимальний код – це найбільш ефективний випадок кодування з нульовою надмірністю. При усуненні надмірності істотно знижується кількість символів, необхідних для повідомлень, що кодуються. Внаслідок цього зменшується час передачі повідомлень, знижується необхідний обсяг пам'яті. Основним завданням процесу передачі інформації є передача з максимально можливою швидкістю при мінімальних інформаційних втратах та мінімальних апаратних витратах [2].

					ЦЗДВН 8.171.00.10.327 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		7

1 ОСНОВНА ЧАСТИНА

1.1 Огляд літератури та постановка завдання проектування

Слово «інформація» виникає від латинського «informatio», що означає роз'яснення, повідомлення, інформованість. Однак, у постійне використання воно увійшло не дуже давно, у середині двадцятого сторіччя, з подачі Клода Шеннона. Він увів це поняття у вузькому технічному сенсі стосовно теорії зв'язку або передачі повідомлень, що отримала назву «Теорія інформації». Філософське, й тим більше широке визначення цього поняття дає один із засновників інформатики, відомий американський вчений Норберт Вінер: «Інформація - це позначення змісту, що береться нами із зовнішнього миру в процесі нашого пристосування до нього і приведення у відповідність до нього нашого мислення». Це далеко не єдине визначення інформації [3].

У цей час таке поняття має більше глибокий та багатогранний зміст. Багато в чому, залишаючись інтуїтивним, це визначення отримує різні сенсові наповнення в різних областях людської діяльності [1, 3]:

- у життєвому аспекті під інформацією розуміють відомості про навколишній світ та процеси, які протікають у ньому, що сприймаються людиною або спеціальними пристроями;
- у техніці під інформацією розуміють повідомлення, які передані у формі знаків або сигналів;
- у теорії інформації (по К. Шеннону) важливі не будь-які відомості, а лише ті, які або знімають повністю, або зменшують існуючу невизначеність;
- у семантичній теорії - це відомості, які володіють новизною.

Така різноманітність підходів не є випадковою, а є наслідком того, що виявилася необхідність усвідомленої організації процесів руху й оброблення того, що має загальну назву - інформація.

Поняття інформації має на увазі передачу інформаційних повідомлень по лініях зв'язку. Розглянемо загальну схему передачі повідомлень. Для визначеності будемо говорити, наприклад, про телеграфію. На одному кінці лінії зв'язку відправник подає деяке повідомлення, яке записане за допомогою кожного з відомих нам алфавітів або цифр, або букв та цифр разом. Для передачі такого роду повідомлень використовуються послідовності сигналів постійного струму, деякі характеристики якого телеграфіст може міняти за своїм розсудом, й які

									Лист
									8
Изм.	Лист	№ докум.	Подпись	Дата					

сприймаються іншим телеграфістом на приймальному кінці лінії зв'язку. Найпростішими помітними сигналами, що широко використовуються на практиці, є послідовність посилки (тобто включення його на деякий цілком визначений час) і відсутність посилки, тобто пауза (вимикання струму на той же час). За допомогою одних тільки цих двох сигналів вже можна передати будь-яке повідомлення, якщо домовитися замінювати кожну символ або цифру певною комбінацією послідовностей струму та пауз [1].

У техніці зв'язку правило, яке ставить у відповідність кожному переданому повідомленню деяку комбінацію сигналів, звичайно називається кодом, а сама операція перетворення повідомлення в послідовність різних сигналів – кодуванням повідомлення. Під «кодом» будемо розуміти таку сукупність n кодових позначень, які ставляться у відповідність до n літер алфавіту, для якої виконується умова: жодне кодове позначення однієї символи не збігається з початком будь-якого іншого більш довгого кодового позначення, тобто коди повинні бути такими, що однозначно декодуються. При цьому коди, які використовують тільки два різних елементарних сигнали (наприклад, послідовність струму та паузу), називаються двійковими кодами. У деяких телеграфних апаратах крім простого включення та вимикання струму можна також змінювати напрямок струму на зворотний [4, 5]. При цьому з'являється можливість замість послідовностей струму та пауз використовувати в якості основних сигналів послідовність струму у двох різних напрямках або використовувати одразу три різних елементарних сигнали однієї й тієї ж тривалості - послідовність струму в одному напрямку, послідовність струму в іншому напрямку й паузу (такий спосіб кодування називається троїчним кодом). Можливі також ще більш складні телеграфні апарати, у яких послідовності струму розрізняються не тільки за напрямком, але й по силі струму. Тим самим ми одержуємо можливість зробити кількість різних елементарних сигналів ще більшою. Збільшення кількості різних елементарних сигналів дозволяє зробити кодові комбінації більш короткими. Однак, разом з тим це ускладнює та здорожчує систему передачі, тому в техніці все ж таки переважно використовуються коди з малою кількістю елементарних сигналів [6, 7].

Повідомлення передаються від об'єкту до адресату за допомогою сукупності технічних засобів, які утворюють систему передачі інформації. Скільки існує методів відображення інформації, стільки можна створити й способів її передачі. До систем передачі інформації відносяться й пошта, й телебачення, й інші методи.

					ЦЗДВН 8.171.00.10.327 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		9

Системи передачі інформації (СПІ) припускають наявність джерела інформації, передавального пристрою, лінії зв'язку, приймального пристрою та одержувача (рис. 1.1) [6, 8].

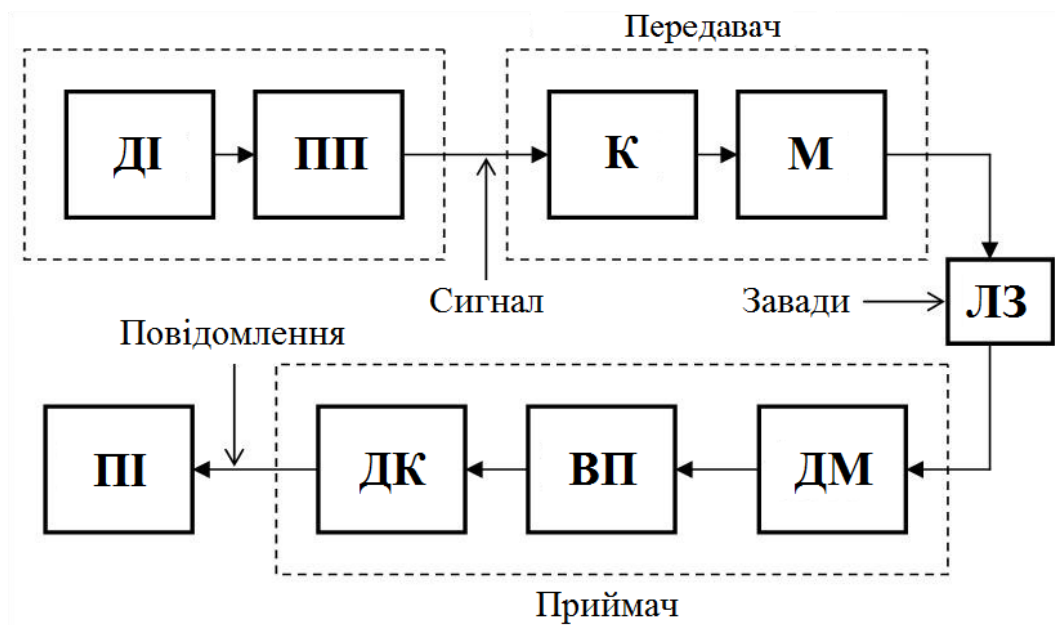


Рисунок 1.1 - Загальна схема системи передачі інформації

Повідомлення надходить від джерела інформації (ДІ) у первинний перетворювач (ПП). За допомогою первинного перетворювача повідомлення, що може мати будь-яку фізичну природу (наприклад, звукове коливання, зображення), перетвориться в первинний сигнал, для технічних пристроїв – як правило в електричний сигнал. Для перетворення первинного сигналу до виду, придатного для використання в лінії зв'язку, призначається передавальний пристрій. У передавальному пристрої здійснюється кодування інформації й вплив на один або кілька параметрів носія за законом, що прийнятий при кодуванні повідомлень. Ці функції покладені на кодер (К) та модулятор (М) відповідно.

Після цього сигнал надходить у лінію зв'язку (ЛЗ), де він підпадає під вплив перешкод. Під перешкодами мають на увазі будь-які зовнішні завади, які спотворюють сигнали, або впливи (атмосферні перешкоди, впливи сторонніх джерел сигналів), а також перекручення сигналів у самій апаратурі (апаратурні перешкоди), що викликають випадкове відхилення прийнятого повідомлення сигналу від переданого.

На приймальній стороні здійснюється відновлення переданого повідомлення по прийнятому сигналу. При цьому спочатку виконується демодуляція сигналу в демодуляторі (ДМ), у результаті якої відновлюється первинний сигнал.

Внаслідок різних перекручень та впливу перешкод прийнятий сигнал може істотно відрізнятись від переданого. Тому виконується оброблення отриманого сигналу з метою найбільш повного відновлення з нього інформації. Ці функції виконуються вирішальним пристроєм (ВП). Процедури оброблення сигналу є досить різноманітними. Вони можуть зводитися до фільтрації сигналів, обмеження, інтегрування (або диференціювання), перемноження сигналів, додавання й т.п.

Після оброблення виконується декодування сигналу, тобто перетворення його в повідомлення, яке надходить до приймача інформації (ПІ). При цьому сигнал повинен бути перетворений до виду, зручному для сприйняття приймачем. Такі функції виконуються декодером (ДК).

В одноканальній системі передачі інформації «об'єкт – це скрипаль у радіостудії, адресат – це слухач») первинним перетворювачем є мікрофон. Сигнал з мікрофона через систему підсилювачів та модулятор надходить до передавача. Потім через приймач демодульований та підсилений сигнал надходить до адресату. Шифратор та дешифратор у даній системі представлені в неявному виді, тому що з деякими допущеннями шифратором можна було б, наприклад, вважати композитора, кодом - ноти, а дешифратором - елементи слухового та емоційного сприйняття слухача. Таке допущення зроблене з метою підкреслити умовність загальної моделі системи передачі інформації [4].

У найпростіших каналах зв'язку приймач, передавач та перетворювач потужності можуть бути з'єднані. Наприклад, у телефоні: сигнали від мікрофона передаються безпосередньо по провідній лінії зв'язку [6].

Канал зв'язку – це сукупність технічних засобів, призначених для передачі інформації від об'єкту до адресату. Лінія зв'язку – це середовище, у якому поширюються сигнали, які несуть інформацію. Для підвищення пропускної здатності ліній зв'язку по них можуть передаватися повідомлення від декількох джерел одночасно. Такий прийом називається ущільненням. У цьому випадку повідомлення від кожного джерела передаються по своєму каналу зв'язку, хоча лінія зв'язку в них є загальною [4 - 6].

Для підвищення швидкості передачі інформації необхідно виконувати її стиснення. Стиснення інформації – це алгоритмічне перетворення даних

					ЦЗДВН 8.171.00.10.327 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		11

(кодування), при якому за рахунок зменшення надмірності повідомлень зменшується їхній обсяг [9].

В основі будь-якого способу стиснення інформації лежить модель джерела інформації, або модель надмірності. Іншими словами, для стиснення інформації використовуються деякі відомості про те, якого роду інформація стискається. Не маючи відомості про інформацію не можна зробити рівно ніяких припущень, що перетворення дозволить зменшити обсяг повідомлення. Ця інформація використовується в процесі стиснення повідомлень та їх зворотного відновлення. Модель надмірності може також будуватися або параметризуватися на етапі стиснення. Методи, які дозволяють на основі вхідних даних змінювати модель надмірності інформації, називаються адаптивними. Неадаптивними є звичайно вузько специфічні алгоритми, які застосовуються для роботи з добре визначеними та незмінними характеристиками. Велика частина досить універсальних алгоритмів є тією або іншою мірою адаптивними [8].

Будь-який метод стиснення інформації містить у собі два перетворення, що є зворотними один до одного:

- стиснення повідомлень;
- зворотне відновлення повідомлень.

Перетворення стиснення забезпечує одержання короткого повідомлення із вихідного. Зворотне відновлення забезпечує одержання вихідного повідомлення (або його наближення) із короткого стисненого повідомлення.

Усі методи стиснення поділяються на два основних класи [2]:

- стиснення без втрат інформації;
- стиснення із втратами інформації.

Кардинальна розбіжність між цими класами стиснення полягає в тому, що стиснення без втрат забезпечує можливість точного відновлення вихідного повідомлення. Стиснення із втратами ж дозволяє одержати тільки деяке наближення вихідного повідомлення, тобто такого, що розрізняється від вихідного, але в межах деяких заздалегідь певних похибок. Ці похибки повинні визначатися іншою моделлю - моделлю приймача, який визначає, які представлені дані й з якою точністю є важливими для одержувача, а які можна взагалі викинути і не враховувати при обробленні отриманої інформації.

Існує дуже багато методів стиснення інформації. Розглянемо кілька з них. Теорія завадостійкого кодування базується на результатах досліджень, проведених Клодом Шенноном. Він сформулював теорему для дискретного каналу із завадами:

					ЦЗДВН 8.171.00.10.327 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		12

«При будь-якій швидкості передачі двійкових символів, меншої, ніж пропускна здатність каналу, існує такий код, при якому ймовірність помилкового декодування буде як завгодно малою» [2, 3, 8].

Для зручності розташуємо всі наявні n символів в один стовпчик у порядку зменшення ймовірностей їх появи у вхідному повідомленні.

Потім всі ці символи потрібно розбити на дві групи - верхню й нижню (першу та другу) – таким чином, щоб сумарна ймовірність в першій групі була найбільш близькою до сумарної ймовірності другої групи. Для символів першої групи в якості першої цифри кодового позначення використовується цифра 1, а для символів другої групи - цифра 0.

Далі кожен із двох груп подібним чином знову потрібно розділити на дві частини й у якості другої цифри кодового позначення використовуємо цифру 1 або 0 залежно від того, належить наша група до першої або до другої з цих підгруп.

Потім кожна з груп, яка містить більше ніж один символ, знову поділяється на дві частини якнайбільше близької сумарної ймовірності й т.п. Процес повторюється доти, поки не отримаємо групи, кожна з яких містить по одному єдиному символу [2].

Такий метод кодування вхідних повідомлень був уперше запропонований у 1948 - 1949 р. незалежно друг від друга Р. Фано і К. Шенноном, тому відповідний код називається кодом Шеннона - Фано.

Близьким до коду Шеннона - Фано є так званий код Хаффмана.

Алгоритм кодування повідомлень Хаффмана дуже схожий з алгоритмом стиснення Шеннона-Фано. Цей алгоритм був винайдений Девидом Хаффманом в 1952 році, і виявився ще більш вдалим, ніж алгоритм Шеннона-Фано. Це обумовлено тим, що алгоритм Хаффмана математично гарантовано створює найменший за розміром код для кожного із символів вихідних даних [2].

Аналогічно застосуванню алгоритму Шеннона-Фано, потрібно побудувати бінарне дерево, яке також буде префіксним деревом, де усі дані зберігаються в листах. Але на відміну від алгоритму Шеннона-Фано, що є спадаючим, цього разу побудова буде виконуватися знизу наверх.

Спочатку ми виконуємо перегляд вхідних даних, підраховуючи кількість появ значень кожного байту, як це робилося й при використанні алгоритму Шеннона-Фано. Як тільки ця таблиця частоти появи символів буде створена, можна перейти до побудови дерева.

					ЦЗДВН 8.171.00.10.327 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		13

Будемо вважати ці пари символ-кількість "пулом" вузлів майбутнього дерева Хаффмана. Видалимо із цього пулу два вузли з найменшими значеннями кількості появ. Приєднаємо їх до нового батьківського вузла й встановимо значення лічильника батьківського вузла рівним сумі лічильників його двох дочірніх вузлів. Помістимо батьківський вузол назад у пул.

Продовжимо цей процес видалення двох вузлів і додавання замість них одного батьківського вузла доти, поки в пулі не залишиться тільки один вузол. На цьому етапі можна видалити з пулу один вузол. Він є кореневим вузлом дерева Хаффмана.

Код Хаффмана завжди є самим економічним із всіх можливих у тому розумінні, що ні для якого іншого методу кодування символів деякого алфавіту середня кількість елементарних символів, які доводяться на один символ, не може бути меншим ніж той, який виходить при кодуванні за цим методом [2, 10].

Коди Хеммінга є кодами, які самоконтролюються, тобто кодами, які дозволяють автоматично виявляти помилки при передачі даних [2]. Для їхньої побудови досить приписати до кожного слова один додатковий (контрольний) двійковий розряд і вибрати цифру цього розряду так, щоб загальна кількість одиниць у зображенні будь-якого числа було, наприклад, парним числом. Одиночна помилка у будь-якому розряді переданого слова (у тому числі, може бути, й у контрольному розряді) змінить парність загальної кількості одиниць. Лічильники по модулю 2, які підраховують кількість одиниць, що містяться серед двійкових цифр числа, можуть давати сигнал про наявність помилок.

При цьому неможливо дізнатися, у якому саме розряді відбулася помилка, й, отже, немає можливості виправити її. Залишаються непоміченими також помилки, які виникають одночасно у двох, у чотирьох або взагалі в парній кількості розрядів. Проте, подвійні, а тим більше чотириразові помилки є, як правило, малоімовірними.

Запропоновані Хеммінгом регулярні методи побудови кодів, які коректують помилки, мають фундаментальне значення. Вони демонструють інженерам практичну можливість досягнення границь, на які вказували закони теорії інформації. Ці коди знайшли практичне застосування при створенні комп'ютерних систем. Робота Хеммінга призвела до рішення проблеми більш щільного упакування для кінцевих полів. Він увів у науковий побут найважливіші поняття теорії кодування - відстань Хеммінга між кодовими комбінаціями у векторному просторі, обумовленому для двійкових кодів як кількість позицій цих комбінацій з

					ЦЗДВН 8.171.00.10.327 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		14

різними символами, та границі Хеммінга для здатності виправляти помилки при застосуванні блокових коригувальних кодів [2, 12].

Робота Хеммінга зіграла ключову роль у наступному розвитку теорії кодування й стимулювала більш досконалі дослідження, які були виконані в наступні роки.

Алгоритм Зіва-Лемпела. Більшість алгоритмів стиснення базується на послідовній схемі стиснення Зіва-Лемпела (1977). Цей алгоритм використовується, зокрема, стандартною процедурою UNIX Compress. Методики зі статистичним моделюванням можуть забезпечити краще стиснення, але вони помітно повільніше. Але існує алгоритм, який сполучає в собі кращі з названих вище рис. Цей алгоритм не передбачає послідовного оброблення вхідних даних, а обробляє текст поблоково. В ньому використовується оборотне перетворення блоку даних до виду, який дозволяє ефективно виконувати стиснення даних за допомогою нескладних алгоритмів. Перетворення має на меті згрупувати символи таким чином, щоб ймовірність появи послідовностей ідентичних символів повідомлення значно зросла. Такий текст може бути легко стиснений за допомогою локально-адаптивних алгоритмів у з'єднанні з кодуванням Хаффмана та арифметичним кодуванням [2].

Послідовність S , яка містить N символів ($\{S(0), S(1), \dots, S(N-1)\}$), піддається N циклічному зсуву (обігу), лексикографічному сортуванню, а останній символ при кожному обігу витягується. З цих символів формується рядок L , де i -ий символ є останнім символом i -го зсуву. Крім рядка L створюється індекс I вихідного рядка S в упорядкованому списку зсувів. Існує ефективний алгоритм відновлення вихідної послідовності символів S на основі рядка L та індексу I . Процедура сортування поєднує результати зсувів з ідентичними початковими символами. Передбачається, що символи в послідовності S відповідають алфавіту, який містить K символів.

Для пояснення роботи алгоритму візьмемо послідовність $S = \text{"abraca"} (N=6)$, алфавіт $X = \{ 'a', 'b', 'c', 'r' \}$ (табл. 1.1).

1. Формуємо матрицю з $N \times N$ елементів, рядки якої являють собою результати циклічного зсуву (обігу) вихідної послідовності S , відсортованих лексикографічно. Принаймні один з рядків M містить вихідну послідовність S . Нехай I є індексом рядка S . У наведеному прикладі індекс $I=1$, а матриця M має вигляд, що наведений у таблиці 1.1.

2. Нехай рядок L являє собою останню колонку матриці M із символами $L[0], \dots, L[N-1]$ (відповідають $M[0, N-1], \dots, M[N-1, N-1]$). Формуємо рядок останніх символів зсуву. Остаточний результат характеризується (L, I) . У даному прикладі $L = \text{'caraab'}$, $I = 1$.

Таблиця 1.1 - Вихідні дані

Номер рядка	Послідовність
0	aabrac
1	abraca
2	acaabr
3	braca
4	caabra
5	racaab

Процедура декомпресії використовує L і I . Метою цієї процедури є одержання вихідної послідовності з N символів (S).

- Спочатку обчислюємо першу колонку матриці M (F). Це робиться шляхом сортування символів рядка L . Кожна колонка вихідної матриці M являє собою перестановки вихідної послідовності S . Таким чином, перша колонка F і L є перестановками S . Тому що рядки в M упорядковані, розміщення символів в F також упорядковані. $F = \text{'aaabcr'}$.
- Розглядаємо ряди матриці M , які починаються із заданого символу «х». Рядки матриці M упорядковані лексикографічно, тому рядки, які починаються з «х» упорядковані аналогічним образом. Визначимо матрицю M' , що виходить із рядків матриці M шляхом циклічного зрушення на один символ вправо.

Для кожного $i=0, \dots, N-1$ і кожного $j=0, \dots, N-1$, $M'[i, j] = m[i, (j-1) \bmod N]$.

В розглянутому прикладі M і M' мають вигляд, наведений у таблиці 1.2.

Подібно M кожний рядок M' є обігом S , і для кожного рядка M існує відповідний рядок M' . M' отримана з M так, що рядки M' упорядковані лексикографічно, починаючи із другого символу. Таким чином, якщо ми розглянемо тільки ті рядки M' , які починаються із заданого символу «х», вони повинні впливати упорядженим чином з обліком другого символу.

Таблиця 1.2 - Побудова матриць М і М'

Рядок	М	М'
0	aabrac	caabra
1	abraca	aabrac
2	acaabr	racaab
3	bracaа	araca
4	caabra	acaabr
5	racaab	bracaа

Отже, для будь-якого заданого символу «х», рядки М, які починаються з «х», з'являються в тій же порядку що й в М', що починаються з «х». У нашому прикладі це видно на прикладі рядків, які починаються з 'а'. Рядка 'aabrac', 'abraca' і 'acaabr' мають номери 0, 1 і 2 в М і 1, 3, 4 в М'.

Використовуючи F і L, перші колонки М і М' ми обчислимо вектор T, що вказує на відповідність між рядками двох матриць, з обліком того, що для кожного $j = 0, \dots, N-1$ рядка j М' відповідають рядкам T[j] М.

Якщо L[j] є k-тою появою «х» в L, тоді T[j]=1, де F[i] є k-тою появою «х» в F. Зазначимо, що T являє собою відповідність один в один між елементами F і елементами L, а F[T[j]] = L[j]. У нашому прикладі T дорівнює: 4 0 5 1 2 3.

3. Тепер для кожного $i = 0, \dots, N-1$ символи L[i] і F[i] є відповідно останніми й першими символами рядка і матриці М. Тому що кожний рядок є зсувом S, символ L[i] є циклічним попередником символу F[i] в S. З T ми маємо F[T[j]] = L[j]. Підставляючи $i = T[j]$, ми одержуємо символ L[T(j)], що циклічно передує символу L[j] в S.

Індекс I вказує на рядок матриці М, де записаний рядок S. Таким чином, останній символ S дорівнює L[I]. Ми використовуємо вектор T для одержання попередників кожного символу: для кожного $i = 0, \dots, N-1$ $S[N-1-i] = L[T^i[I]]$, де $T^0[x]=x$, а $T^{i+1}[x] = T[T^i[x]]$. Ця процедура дозволяє відновити первісну послідовність символів S ('abraca').

Послідовність $T^i[I]$ для $i = 0, \dots, N-1$ не обов'язково є перестановкою чисел $0, \dots, N-1$. Якщо вихідна послідовність S є формою Z^p для деякої підстановки Z і для деякого $p > 1$, тоді послідовність $T^i[I]$ для $i = 0, \dots, N-1$ буде також формою Z^p для деякої субпослідовності Z'. Таким чином, якщо S = 'cancan', Z = 'can' і p=2, послідовність $T^i[I]$ для $i = 0, \dots, N-1$ буде [2,4,0,2,4,0].

Описаний вище алгоритм впорядковує зсув вихідної послідовності символів S і формує рядок L , що складається з останніх символів зсувів. Для того, щоб зрозуміти, чому таке впорядкування приводить до більш ефективного стиснення, розглянемо вплив на окрему літеру у звичайному слові англійського тексту [2, 10].

Візьмемо як приклад символ "t" у слові 'the' й припустимо, що вихідна послідовність містить багато таких слів. Коли список зсувів є упорядкованим, усі зсуви, які починаються з 'he', будуть взаємно упорядковані. Один відрізок рядка L буде містити непропорційно велике число 't', перемішаних з іншими символами, які можуть передувати 'he', такими як пробіл, 's', 'T' і 'S'.

Аналогічні аргументи можуть бути використані для всіх символів всіх слів, таким чином, будь-яка область рядка L буде містити велику кількість деяких символів. У результаті ймовірність того, що символ 'ch' зустрінеться в даній крапці L , є досить більшою, якщо ch зустрічається біля цієї крапки L , і малою в протилежному випадку. Ця властивість сприяє ефективній роботі локально адаптивних алгоритмів стиснення, де кодується відносно положення ідентичних символів. У випадку застосування до рядка L , такий кодувальник буде видавати малі числа, які можуть сприяти ефективній роботі наступного кодування, наприклад, за допомогою алгоритму Хаффмана [10].

Локально адаптивний алгоритм стиснення [8].

Цей алгоритм використовується для кодування (L, I) , де L рядок довжиною N , а I - індекс. Це кодування містить у собі кілька етапів.

1. Спочатку кодується кожен символ L з використанням локально адаптивного алгоритму для кожного із символів індивідуально. Визначається вектор цілих чисел $R[0], \dots, R[N-1]$, що являє собою коди для символів $L[0], \dots, L[N-1]$. Ініціалізується список символів Y , що містить у собі кожен символ з алфавіту X тільки один раз. Для кожного $i = 0, \dots, N-1$ встановлюється $R[i]$ рівним числу символів, які передують символу $L[i]$ зі списку Y . Взявши $Y = ['a', 'b', 'c', 'r']$ у якості вихідного й $L = 'caraab'$, обчислюємо вектор R : (2 1 3 1 0 3).
2. Застосовуємо алгоритм Хаффмана, або інший аналогічний алгоритм стиснення, до елементів R , розглядаючи кожний елемент як об'єкт для стиснення. У результаті виходить код OUT та індекс I .

Розглянемо процедуру декодування отриманого короткого тексту (OUT, I) .

					ЦЗДВН 8.171.00.10.327 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		18

На основі (OUT,I) необхідно обчислити (L,I). Передбачається, що список Y відомий.

1. Спочатку обчислюється вектор R, що містить N чисел: (2 1 3 1 0 3).
2. Далі обчислюється рядок L, який містить N символів, що дає значення R[0],..R[N-1]. Якщо необхідно, ініціалізується список Y, що містить символи алфавіту X (як й при процедурі кодування). Для кожного $i = 0, \dots, N-1$ послідовно встановлюється значення L[i], що дорівнює символу в положенні R[i] зі списку Y (нумерується, починаючи з 0), потім символ зрушується до початку Y. Результуючий рядок L являє собою останню колонку матриці M. Результатом роботи алгоритму буде (L,I). Взяти Y = ['a','b','c','r'], обчислюємо рядок L = 'cagaab'.

Найбільш важливим фактором, що визначає швидкість стиснення, є час, необхідний для сортування зсувів у вхідному блоці. Найбільш швидкий спосіб рішення проблеми полягає в сортуванні зв'язаних рядків по суфіксах.

Для того щоб стиснути рядок S, спочатку сформуємо рядок S', який є об'єднанням S із EOF, новим символом, що не зустрічається в S. Після цього використовується стандартний алгоритм до рядка S'. Оскільки EOF відрізняється від інших символів в S, суфікси S' сортуються в тім же порядку, що й зсуви S'. Це може бути зроблено шляхом побудови дерева суфіксів, що може бути потім застосовано в лексикографічному порядку для сортування суфіксів. Для цієї мети може бути використаний алгоритм формування дерева суфіксів Мак-Крейга. Його швидкодія становить 40% від найбільш швидкої методики у випадку роботи з текстами. Алгоритм роботи з деревом суфіксів вимагає більше чотирьох слів на кожний вихідний символ.

Манбер і Майєрс запропонували нескладний алгоритм сортування суфіксів рядку. Цей алгоритм вимагає тільки двох слів на кожний вхідний символ. Алгоритм працює спочатку з першими і символами суфікса, а потім, використовуючи положення суфіксів у масиві, що сортирується, здійснює сортування для перших 2i символів. Нажаль, цей алгоритм працює помітно повільніше [2, 10].

Існує кращий алгоритм сортування суфіксів [8]. У цьому алгоритмі сортуються суфікси рядку S, який містить N символів S[0,., N-1].

1. Нехай k - кількість символів, які відповідають машинному слову. Утворимо рядок S' із S шляхом додавання k символів EOF до рядку S. Передбачається, що EOF не зустрічається в рядку S.

										Лист
										19
Изм.	Лист	№ докум.	Подпись	Дата						

2. Ініціалізуємо масив W з N слів $W[0, \dots, N-1]$ таким чином, що $W[i]$ містять символи $S'[i, \dots, i+k-1]$, впорядковані таким чином, що цілочисельне порівняння слів узгоджується з лексикографічним порівнянням для k -символьних рядків. Упакування символів у слова має дві переваги: це дозволяє для двох префіксів зрівняти відразу k байт та відкинути багато випадків, що описані нижче.
3. Ініціалізується масив V з N цілих чисел. Якщо елемент V містить j , він являє собою суфікс S' , чий перший символ дорівнює $S'[j]$. Коли виконання алгоритму буде завершено, суфікс $V[i]$ буде i -тим суфіксом у лексикографічному порядку.
4. Ініціалізуємо цілочисельний масив V таким чином, що для кожного $i = 0, \dots, N-1$: $V[i]=i$.
5. Сортуємо елементи V , використовуючи перші два символи кожного суфікса в якості ключа сортування. Далі для кожного символу « x » з алфавіту виконуємо кроки 6 і 7. Коли ці ітерації будуть завершені, V являє собою відсортовані суфікси S і робота алгоритму завершується.
6. Для кожного символу « x » в алфавіті виконуємо сортування елементів V , що починаються з « x », за яким йде « x ». У процесі виконання сортування порівнюємо елементи V шляхом зіставлення суфіксів, які вони представляють при індексуванні масиву W . На кожному кроці рекурсії варто відслідковувати кількість символів, які виявилися однаковими в групі, щоб не порівнювати їх знову. Усі суфікси, які починаються з « x », відсортовані в рамках V .
7. Для кожного елемента $V[i]$, що відповідає суфіксу, що починається з « x » (тобто, для якого $S[V[i]] = \langle x \rangle$), потрібно встановити $W[V[i]]$ значення з « x » у старших бітах і i у молодших бітах. Нове значення $W[V[i]]$ сортується в ті ж позиції, що й попередні значення.

Даний алгоритм може бути поліпшений різними способами. Одним з очевидних методів є вибір символу « x » на етапі 5, починаючи з найменшого загального символу в S і попередній до найбільш загального.

Стиснення даних з використанням перетворення Барроуза-Уїлера [2, 8].

Майкл Барроуз і Давид Уїлер в 1994 році запропонували свій алгоритм перетворення (BWT). Цей алгоритм працює із блоками даних та забезпечує ефективний стиснення без втрати інформації. У результаті перетворення блок даних має таку ж довжину, але має інший порядок розташування символів.

					ЦЗДВН 8.171.00.10.327 ПЗ	Лист
						20
Изм.	Лист	№ докум.	Подпись	Дата		

Алгоритм є тим ефективнішим, чим більший блок даних перетворюється (наприклад, 256-512 КБайт).

Пояснення роботи алгоритму буде виконано на обмеженому обсязі вихідних даних (рядок S довжиною N, наприклад, SEMENOV). Рядок S розглядається як послідовність, що складається з N рядків.

Спочатку здійснюється циклічний зсув рядка S і формується N-1 новий рядок. На практиці рядки не розмножуються, а створюється масив покажчиків на циклічний буфер, де розташований вихідний рядок S (табл. 1.3).

Таблиця 1.3 - Циклічне зрушення рядка

Комбінації	Рядок
SEMENOV	S0
EMENOV S	S1
MENOVSE	S2
ENOVSEM	S3
NOVSEME	S4
OVSEMEN	S5
VSEMENO	S6

Далі виконується лексикографічне впорядкування (сортування) цих рядків (покажчиків) (табл. 1.4). Для впорядкування можна використовувати функції типу strcmp або memcmp (з огляду на особливості структури буфера, це повинні бути їхні функціональні аналоги).

Таблиця 1.4 - Отриманий результат

F						L	Рядок
E	M	E	N	O	V	S	S1
E	N	O	V	S	E	M	S3
M	E	N	O	V	S	E	S2
N	O	V	S	E	M	E	S4
O	V	S	E	M	E	N	S5
S	E	M	E	N	O	V	S0
V	S	E	M	E	N	O	S6

Перша колонка позначена буквою F, а остання - L. Колонка F (EEMNOSV) містить усі символи вихідного рядку S, які записані в упорядкованій послідовності. Рядок L являє собою послідовність префіксних символів для рядка S.

Результатом роботи алгоритму BWT є рядок L і первинний індекс, що являє собою номер елемента рядка L, де зберігається перший символ вихідного рядка S. У наведеному прикладі індекс рівняється 1 [8].

Можна зробити наступні висновки, що інформатика та обчислювальна техніка - це галузь науки і техніки, що включає сукупність засобів, способів і методів людської діяльності, спрямованих на створення та застосування пристроїв зв'язку, систем збору, зберігання та оброблення інформації.

У багатьох випадках збережена та передана інформація може становити інтерес для осіб, які бажають використовувати неї в корисливих цілях.

Одним з методів захисту цієї інформації є її кодування.

Кодування - це відображення повідомлень за допомогою кодових комбінацій за певним правилом присвоєння символів [2, 8, 10].

Код - це правило, що описує відображення одного набору знаків в інший набір знаків (або слів). Кодом також називають й множину образів при цьому відображенні.

Оптимальний код - це найбільш ефективний випадок кодування з нульовою надмірністю. При усуненні надмірності істотно знижується кількість символів, необхідних для повідомлень, що кодуються. Внаслідок цього зменшується час передачі інформації, знижується необхідний обсяг пам'яті.

Керування правильністю передачі інформації виконується за допомогою завадостійкого кодування. Є коди, які виявляють помилки, і коди, які коректують і виправляють помилки. Завадозахищеність досягається за допомогою введення надмірності, додаткових бітів. У симплексних каналах зв'язку відстороняють помилки за допомогою коригувальних кодів. У дуплексних - достатньо застосування кодів, які виявляють помилки. Це основні методи, що використовуються в інформаційних мережах.

Найпростішими способами виявлення помилок є контрольне підсумовування, перевірка на непарність.

Таким чином, знання методів оброблення інформації є базовим для інженерів, робота яких пов'язана з обчислювальними системами та мережами.

					ЦЗДВН 8.171.00.10.327 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		22

Надмірність – це додаткові засоби, які вводяться в систему для підвищення її надійності та захищеності.

Отже, поставимо завдання розробити систему передачі даних на основі коду Шеннона - Фано. Вхідне повідомлення закодовано за допомогою коду ASCII. Пристрій побудувати на основі мікропроцесорної системи і розробити програмне забезпечення. Пристрій повинен виконувати стиснення інформації та забезпечувати передачу даних з максимальною швидкістю при мінімальних апаратних витратах.

Для розроблення системи передачі даних на основі коду Шеннона - Фано необхідно виконати наступне:

1. Дослідити характеристики ефективних кодів, розглянути алгоритм формування кодових комбінацій згідно методу Шеннона-Фано, виконати оцінку ефективності коду Шеннона – Фано та здійснити порівняльний аналіз характеристик ефективних кодів, сформованих за допомогою різних методів стиснення.
2. Розробити структурну схему та схему алгоритму роботи пристрою стиснення даних на базі метода Шеннона-Фано.
3. Розробити функціональну та принципову схеми пристрою.
4. Розрахувати показники собівартості виготовлення пристрою.

1.2 Науково - дослідна частина

1.2.1 Характеристики ефективних кодів

У комп'ютерних інформаційних системах при передачі або зберіганні інформація представляється в певних форматах. При цьому багато традиційно використовуваних форматів є надлишковими, тобто обсяг представленої ними корисної інформації є меншим, ніж відповідний фізичний обсяг, необхідний для передачі або зберігання. При цьому надмірність існує як у первинному алфавіті, так й у природі коду, який використовується для кодування повідомлень, тобто у двійковому алфавіті [4, 5].

При кодуванні текстової інформації надмірність може бути обумовлена й нерівномірністю появи символів у тексті, а також за рахунок надмірності самого двійкового коду (може мати місце ситуація, коли ймовірності появи нулів

					ЦЗДВН 8.171.00.10.327 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		23

та одиниць у двійковому коді є різними, тобто код не є оптимальним. Це відповідає випадку, коли кількість символів алфавіту повідомлення не дорівнює цілому ступеню двійки або за якимись причинами розрядність двійкового коду вибрана більшою, ніж така, що є мінімально необхідною).

З метою підвищення ефективності подання інформації при передачі й зберіганні в комп'ютерних системах застосовується оптимальне кодування. При оптимальному кодуванні мінімізується кількість двійкових символів, що необхідні для подання одного символу повідомлення. Таким чином, зменшується надмірність подання інформації, й, відповідно, зменшується необхідний фізичний обсяг повідомлення, що передається.

Завданням оптимального кодування є побудова оптимального коду, надмірність якого прагне до нуля. Принципова можливість побудови такого коду сформульована в теоремі Шеннона.

Повідомлення, що складене з літер деякого алфавіту, можна закодувати таким чином, щоб середня кількість двійкових символів на один символ алфавіту повідомлення буде як завгодно близькою до ентропії джерела повідомлення, але не бути менше її.

Ентропією джерела повідомлення є величина, яка враховує його ймовірнісні характеристики та обчислюється за формулою [2]:

$$H = -\sum_{i=1}^n p_i \log_2 p_i, \quad (1.1)$$

де p_i - ймовірність i -го символу алфавіту повідомлення;

n - кількість символів в алфавіті повідомлення (потужність первинного алфавіту).

У цій формулі не враховується можлива кореляція між символами повідомлення.

Середня довжина кодової комбінації є характеристикою ефективності оптимального коду й обчислюється за формулою [2]:

$$l_{cp} = \sum_{i=1}^n p_i l_i, \quad (1.2)$$

					ЦЗДВН 8.171.00.10.327 ПЗ	Лист
						24
Изм.	Лист	№ докум.	Подпись	Дата		

де P_i - ймовірність і-го символу алфавіту повідомлення;

l_i - довжина кодової комбінації для і-го символу алфавіту повідомлення ;

n - кількість символів в алфавіті повідомлення.

Наявність у повідомленнях надмірності дозволяє ставити питання про стиснення даних, тобто про передачу тієї ж кількості інформації за допомогою послідовностей символів меншої довжини. Для цього використовуються спеціальні алгоритми стиснення повідомлень, які зменшують їх надмірність. Ефект стиснення оцінюють коефіцієнтом стиснення [2]:

$$K = n/q, \quad (1.3)$$

де n - кількість мінімально необхідних символів для передачі повідомлення (практично це число символів на виході еталонного алгоритму стиснення);

q - кількість символів у повідомленні, яке було стиснене за допомогою даного алгоритму.

Так, при двійковому кодуванні n дорівнює ентропії джерела інформації. Часто ступінь стиснення оцінюють відношенням довжин кодів на вході та виході алгоритму стиснення.

Поруч з методами стиснення, які не зменшують кількість інформації в повідомленні, застосовуються методи стиснення, що засновані на втраті інформації, яка не має великого значення.

При формуванні кожного символу кодової комбінації усі зусилля необхідно направити на те, щоб за допомогою цього символу була передана максимальна кількість інформації. В ідеальному випадку при двійковому кодуванні символи 0 і 1 повинні зустрічатися з рівною ймовірністю, причому символи в різних розрядах повинні бути незалежними один від одного.

Методи Шеннона - Фано та Хаффмана реалізують розглянуті ідеї оптимального кодування та відрізняються алгоритмами побудови оптимального коду. Побудовані за допомогою цих методів кодові комбінації мають різну довжину і мають меншу надмірність у порівнянні з рівномірними кодами внаслідок зменшення значення середньої довжини кодових слів - це так звані оптимальні нерівномірні коди [8].

1.2.2 Коди Шеннона - Фано

Алгоритм Шеннона - Фано – це один з перших алгоритмів стиснення, який вперше сформулювали американські вчені К. Шеннон і Р. Фано. Даний метод стиснення має велику подібність з алгоритмом Хаффмана, що з'явився на кілька років пізніше. Алгоритм використовує коди змінної довжини: символ, що зустрічається часто, кодується кодовою комбінацією меншої довжини, а символ, що зустрічається рідко - кодом більшої довжини. Коди Шеннона - Фано є префіксними, тобто жодне кодове слово не є префіксом будь-якого іншого. Інакше – жодна кодова комбінація не є початком іншої. Ця властивість дозволяє однозначно декодувати будь-яку послідовність кодових слів [8].

Алгоритм кодування Шеннона - Фано є наступним.

1. Повідомлення записуються в таблицю в порядку зменшення ймовірності їх появи.
2. Виконується розподіл на групи так щоб суми ймовірностей в кожній з груп були б по можливості однаковими.
3. Першій групі присвоюється значення «0», а другій – значення «1».
4. Повернення до пункту 2. Розподіл виконується до тих пір поки в кожній з груп не буде по одному символу [2].
5. Записується кодове зображення.

Розглянемо приклад формування кодових комбінацій за методом Шеннона-Фано.

Нехай задані такі характеристики первинного алфавіту:

$$N_p = 6; P_1 = 0,4; P_2 = 0,2; P_3 = 0,2; P_4 = 0,1; P_5 = 0,05; P_6 = 0,05.$$

Потрібно сформувати код Шеннона-Фано, перевірити кодові комбінації на префіксність, визначити їхню середню довжину та безумовну ентропію, переконатись у тому, що сформований код відповідає теоремі Шеннона для каналів зв'язку без завад.

Процес формування коду Шеннона-Фано наведений в таблиці 1.5.

Сформоване кодове зображення:

$$F(A) = \{(a1;0);(a2;10);(a3;110);(a4;1110);(a5;11110);(a6;11111)\}.$$

										Лист
										26
Изм.	Лист	№ докум.	Подпись	Дата						

Таблиця 1.5 - Формування коду Шеннона-Фано

a_i	$P(a_i)$	Розподіл на групи	Кодове слово				
a_1	0,4	} I	0				
a_2	0,2	} II } I	1	0			
a_3	0,2	} II } II } I	1	1	0		
a_4	0,1	} II } II } II } I	1	1	1	0	
a_5	0,05	} II } II } II } II } I	1	1	1	1	0
a_6	0,05	} II } II } II } II } II } I	1	1	1	1	1

Сформоване кодове зображення:

$$F(A) = \{(a_1;0);(a_2;10);(a_3;110);(a_4;1110);(a_5;11110);(a_6;11111)\}.$$

Основний принцип, покладений в основу кодування по методу Шеннона - Фано, полягає в тому, що при виборі кожної цифри кодового позначення кількість інформації, що міститься в ній, було найбільшою, тобто щоб незалежно від значень усіх попередніх цифр ця цифра приймала будь-яке можливе для неї значення (0 або 1) по можливості з однаковою ймовірністю. Кількість цифр у різних позначеннях при цьому виявляється різною, тобто код Шеннона – Фано є нерівномірним. Жодне кодове позначення тут не може бути початком іншого, більш довгого позначення. Тому закодоване повідомлення завжди може бути однозначно декодовано.

У результаті, середнє значення довжини такого позначення виявляється не набагато більшим мінімального значення безумовної ентропії H цього повідомлення. Так, для розглянутого вище прикладу 6-символьного алфавіту найкращий рівномірний код складається з тризначних кодових позначень (оскільки $2^2 < 6 < 2^3$). Тому в ньому на кожен символ вихідного повідомлення доводиться рівно 3 елементарні сигнали. А при використанні коду Шеннона – Фано середня кількість елементарних символів, які доводяться на один символ повідомлення, дорівнює

$$1 \cdot 0,4 + 2 \cdot 0,2 + 3 \cdot 0,2 + 4 \cdot 0,1 + 5 \cdot (0,05 + 0,05) = 2,3.$$

Це значення помітно менше, ніж 3, і не дуже далеко від ентропії

$$H = -0,4 \log 0,4 - 2 \cdot 0,2 \log 0,2 - 0,1 \log 0,1 - 2 \cdot 0,05 \log 0,05 \approx 2,22.$$

Таким чином $L_{сер} \geq H(A)$.

Аналогічно при розгляді 18-символьного алфавіту найкращий рівномірний код складається з п'ятизначних кодових позначень (тому що $2^4 < 18 < 2^5$). У випадку формування коду Шеннона – Фано є символи, що кодуються навіть за допомогою семи двійкових розрядів, але зате середня кількість елементарних сигналів, які доводяться на один символ, дорівнює

$$2 \cdot 0,5 + 3 \cdot 0,1 + 4 \cdot 0,15 + 5 \cdot 0,15 + 6 \cdot 0,06 + 7 \cdot 0,04 = 3,29.$$

Останнє значення помітно менше, ніж 5, й вже не набагато відрізняється від величини

$$H = -0,3 \log 0,3 - 0,2 \log 0,2 - \dots - 6 \cdot 0,01 \log 0,01 \approx 3,25.$$

Особливо зручним може бути кодувати за методом Шеннона – Фано не окремі символи, а одразу цілі блоки з декількох символів. Правда, при цьому однаково неможливо перевершити граничне значення H двійкових знаків на один символ повідомлення [8, 10].

Розглянемо, наприклад, випадок, коли є лише два різних символи A і B , які характеризуються ймовірностями $p(A) = 0,7$ і $p(B) = 0,3$.

В процесі кодування за методом Шеннона – Фано окремих символів отримуємо код, що наведений у таблиці 1.6.

Таблиця 1.6 - Вихідні дані та результат кодування

Символ	Ймовірність	Кодова комбінація
A	0,7	1
B	0,3	0

Тоді

$$H = -0,7 \log 0,7 - 0,3 \log 0,3 = 0,881.$$

Застосування методу Шеннона - Фано до вихідного двохсимвольного алфавіту виявляється безцілним: воно призводить лише до найпростішого рівномірного коду, що потребує для передачі кожного символу одного двійкового знаку. А це на 12% більше, ніж мінімальне значення, яке можна отримати - 0,881 дв.зн./символ.

Застосовуючи метод Шеннона - Фано для кодування двохсимвольних комбінацій (ймовірність яких визначається правилом множення ймовірностей для незалежних подій), отримаємо код, неведений у таблиці 1.7.

Таблиця 1.7 – Результати кодування

Комбінація символів	Ймовірність	Кодова комбінація
<i>АА</i>	0,49	1
<i>АВ</i>	0,21	01
<i>БА</i>	0,21	001
<i>ББ</i>	0,09	000

Середнє значення довжини кодового позначення (кової комбінації) в цьому випадку дорівнює

$$1 \cdot 0,49 + 2 \cdot 0,21 + 3 \cdot 0,3 = 1,81.$$

Тобто на один символ алфавіту доводиться в середньому $\frac{1,81}{2} = 0,905$ двійкових знаків – лише на 3% більше значення 0,881 дв.зн./символ.

Ще кращі результати одержимо, застосувавши метод Шеннона - Фано до кодування трьохбуквеної комбінації (табл. 1.8).

Середнє значення довжини кодового позначення в цьому дорівнює 2,686, тобто на один символ тексту доводиться в середньому 0,895 двійкових знаків, що всього на 1,5% більше значення $H \approx 0,881$ дв.зн./символ.

У випадку ще більшої різниці в ймовірностях символів *А* і *Б* наближення до мінімально можливого значення H дв.зн./символ може бути трохи менш швидким, але воно проявляється не менш наочно.

Таблиця 1.8 - Результати виконання алгоритму

Комбінація символів	Ймовірність	Кодова комбінація
<i>AAA</i>	0,343	11
<i>AAB</i>	0,147	10
<i>ABA</i>	0,147	011
<i>BAA</i>	0,147	010
<i>ABB</i>	0,063	0010
<i>BAB</i>	0,063	0011
<i>BBA</i>	0,063	0001
<i>BBB</i>	0,027	0000

Таким чином, при $p(A) = 0,89$ і $p(B) = 0,11$ це значення дорівнює

$$-0,89 \log_2 0,89 - 0,11 \log_2 0,11 \approx 0,5 \text{ дв.зн./символ.}$$

Рівномірний код $A \rightarrow 1, B \rightarrow 0$ (рівносильний застосуванню коду Шеннона – Фано до сукупності двох наявних символів) вимагає витрати одного двійкового знаку на кожен символ – це у два рази більше. Неважко перевірити, що застосування коду Шеннона - Фано до будь-яких двохсимвольних комбінацій призводить до коду, у якому на кожен символ доводиться в середньому 0,66 двійкових знаків, застосування до трьох символьних комбінацій - 0,55, а до чотирьохсимвольних блоків - у середньому 0,52 двійкових знаків - усього на 4% більше, ніж мінімальне значення 0,50 дв.зн./символ [2].

1.2.3 Коди Хаффмана

Близьким до коду Шеннона – Фано, але ще вигіднішим, ніж останній, є код Хаффмана. Побудова цього коду спирається на просте перетворення алфавіту, що використовується. Таке перетворення носить назву стиснення алфавіту [2, 8].

Алгоритм Хаффмана формування кодів комбінацій складається з наступних етапів.

1. Повідомлення записуються в таблицю в порядку зменшення ймовірності їх появи.

2. Два останніх повідомлення об'єднуються в одне допоміжне, якому присвоюється сумарна ймовірність.

3. Повідомлення знову розташовують в порядку зменшення ймовірностей в додатковому стовпці, а два останніх об'єднуються.

4. Процес продовжується до тих пір поки не отримаємо єдине повідомлення з ймовірністю, що дорівнює «1».

5. Щоб скласти кодову комбінацію, відповідно з цим повідомленням, необхідно простежити шлях переходу повідомлення по рядках та стовпцях таблиці. Для наочності будується кодове дерево.

Для цього з точки, яка відповідає ймовірності «1», направляються дві гілки. Гілки з більшою ймовірністю присвоюється «1», з меншою ймовірністю – «0». Розгалуження продовжується доки не буде досягнута ймовірність кожного повідомлення.

6. Рухаючись по кодовому дереву зверху вниз для кожного повідомлення записується відповідна кодова комбінація.

7. Записується кодове зображення.

Розглянемо приклад формування кодових комбінацій за методом Хаффмана. Нехай задані такі характеристики первинного алфавіту:

$$N_p = 6; P_1 = 0,4; P_2 = 0,2; P_3 = 0,2; P_4 = 0,1; P_5 = 0,05; P_6 = 0,05.$$

Потрібно сформувані код Хаффмана, перевірити кодові комбінації на префіксність, визначити їхню середню довжину та безумовну ентропію, переконатись у тому, що сформований код відповідає теоремі Шеннона для каналів зв'язку без завад.

Процес формування коду Хаффмана наведений в таблиці 1.9.

Таблиця 1.9 - Формування коду Хаффмана

a_i	$P(a_i)$	Допоміжні стовпці					
a_1	0,4	0,4	0,4	0,4	0,4	0,6	1
a_2	0,2	0,2	0,2	0,2	0,4	0,4	
a_3	0,2	0,2	0,2	0,2	0,2		
a_4	0,1	0,1	0,1	0,2			
a_5	0,05	0,05	0,1				
a_6	0,05	0,05					

Побудуємо кодове дерево (рис. 1.2).

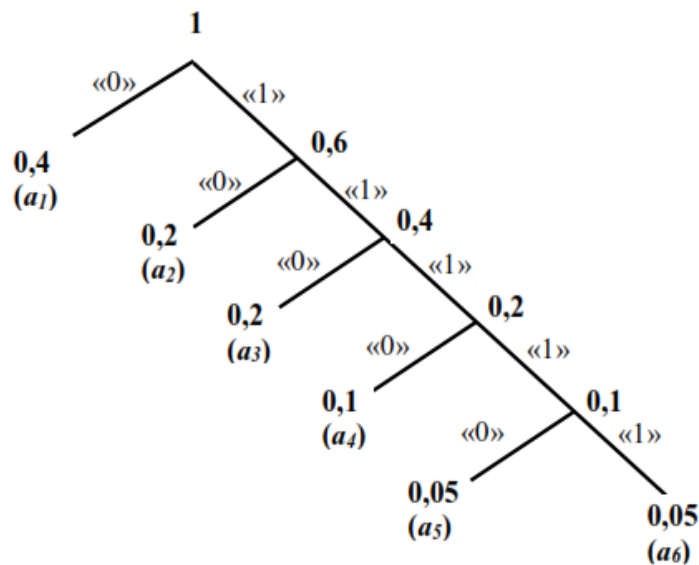


Рисунок 1.2 - Кодове дерево

Рухаючись по кодовому дереву зверху вниз для кожного повідомлення сформуємо відповідні кодові комбінації.

Кодове зображення:

$$F(A) = \{(a_1;0);(a_2;10);(a_3;110);(a_4;1110);(a_5;11110);(a_6;11111)\}.$$

Отриманий код є найекономічнішим.

Середня кількість символів на одне повідомлення $L_{сер}$ для розглянутого набору повідомлень

$$1 \cdot 0,4 + 2 \cdot 0,2 + 3 \cdot 0,2 + 4 \cdot 0,1 + 5 \cdot (0,05 + 0,05) = 2,3.$$

Ентропія вхідного повідомлення:

$$\begin{aligned} H(A) &= - \sum_i P(a_i) \log_2 P(a_i) = \\ &= -0,4 \cdot \log_2 0,4 - 2 \cdot 0,2 \cdot \log_2 0,2 - 0,1 \cdot \log_2 0,1 - 2 \cdot 0,05 \cdot \log_2 0,05 \approx \\ &\approx 2,22 \text{ біт/символ} \end{aligned}$$

Таким чином $L_{сер} \geq H(A)$.

Нехай є алфавіт A , який містить символи a_1, a_2, \dots, a_n , ймовірності появи яких у повідомленні відповідно дорівнюють p_1, p_2, \dots, p_n . При цьому вважаємо символи такими, що розташовані в порядку спадання імовірностей їх появи (або частот), тобто

$$p_1 \geq p_2 \geq p_3 \geq \dots \geq p_{n-1} \geq p_n.$$

Умовимося тепер не розрізняти між собою два найменш можливих символи вхідного алфавіту, тобто будемо вважати, що a_{n-1} і a_n – це той самий символ b нового алфавіту A_1 , що містить тепер вже символи a_1, a_2, \dots, a_{n-2} та b (тобто a_{n-1} або a_n), ймовірності появи яких у повідомленні відповідно дорівнюють p_1, p_2, \dots, p_{n-2} та $p_{n-1} + p_n$. Алфавіт A_1 й називається алфавітом, що отриманий з алфавіту A за допомогою стиснення (або однократного стиснення).

Розташуємо символи нового алфавіту A_1 у порядку спадання їх імовірностей і виконаємо стиснення алфавіту A_1 . При цьому прийдемо до алфавіту A_2 , що отриманий з первинного алфавіту A за допомогою дворазового стиснення (й з алфавіту A_1 – за допомогою простого або однократного стиснення). При цьому алфавіт A_2 буде містити вже усього $(n - 2)$ символи. Продовжуючи цю процедуру, будемо приходити до все більш коротких алфавітів. Після $(n-2)$ -кратного стиснення прийдемо до алфавіту A_{n-2} , який містить вже усього два символи [6].

Приклад перетворення за допомогою послідовних стиснень алфавіту, який містить 6 символів, ймовірності яких дорівнюють, відповідно, 0,4, 0,2, 0,2, 0,1, 0,05 і 0,05, наведений в таблиці 1.9 та на рисунку 1.2.

Умовимося тепер приписувати до двох символів останнього алфавіту A_{n-2} кодові позначення 1 і 0. Далі, якщо кодові позначення вже приписані до усіх символів алфавіту A_j , то символам «попереднього» алфавіту A_{j-1} (де $A_{1-1} = A_0$ – це вихідний алфавіт A), що зберігся й в алфавіті A_j , пишемо ті ж кодові позначення, які вони мали в алфавіті A_{j-1} . Двом символам a' і a'' алфавіту A_j , які збіглись в один символ « b » алфавіту A_{j-1} , припишемо позначення, які отримані з кодового позначення символу b додаванням цифр 1 і 0 наприкінці (табл. 1.10).

					ЦЗДВН 8.171.00.10.327 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		33

Таблиця 1.10 - Побудова коду Хаффмана

№ символу	Ймовірності появи символів алфавіту у вихідному повідомленні				
	Вихідний алфавіт A	Алфавіти у результаті стиснення			
		A_1	A_2	A_3	A_4
1	0,4 0	0,4 0	0,4 0	0,4 0	0,6 1
2	0,2 10	0,2 10	0,2 10	0,4 11	0,4 0
3	0,2 111	0,2 111	0,2 111	0,2 10	
4	0,1 1101	0,1 1101	0,2 110		
5	0,05 11001	0,1 1100			
6	0,05 11000				

Із побудови одержуваного в такий спосіб коду Хаффмана випливає, що він задовольняє загальній умові - ніяке кодове позначення не є початком іншого, більш довгого кодового позначення. Тобто отриманий код є префіксним. Помітимо ще, що кодування деякого алфавіту за методом Хаффмана (так як й за методом Шеннона - Фано) не є однозначною процедурою [6].

Так, наприклад, на будь-якому етапі побудови коду можна замінити цифру 1 на цифру 0 й навпаки. При цьому одержимо два різних коди, що відрізняються досить несуттєво один від одного. Але, крім того, в деяких випадках можна побудувати й кілька кодів Хаффмана, що істотно розрізняються. Так, наприклад, у наведеному прикладі можна будувати код й відповідно до таблиці 1.11.

Таблиця 1.11 – Побудова коду Хаффмана

№ символу	Ймовірності появи символів алфавіту				
	Вихідний алфавіт A	Алфавіти у результаті стиснення			
		A_1	A_2	A_3	A_4
1	0,4 11	0,4 11	0,4 11	0,4 0	0,6 1
2	0,2 01	0,2 01	0,2 10	0,4 11	0,4 0
3	0,2 00	0,2 00	0,2 01	0,2 10	
4	0,1 100	0,1 101	0,2 00		
5	0,05 1011	0,1 100			
6	0,05 1010				

Сформований при цьому новий код також є кодом Хаффмана, але довжини кодових комбінацій тепер є зовсім іншими.

Відзначимо, що середня кількість елементарних сигналів, які доводяться на один символ, для обох побудованих кодів Хаффмана виявляється однаковою. У першому варіанті середня кількість розрядів дорівнює

$$1 \cdot 0,4 + 2 \cdot 0,2 + 3 \cdot 0,2 + 4 \cdot 0,1 + 5 \cdot (0,05 + 0,05) = 2,3.$$

В другому варіанті середня кількість розрядів дорівнює

$$2 \cdot (0,4 + 0,2 + 0,2) + 3 \cdot 0,1 + 4 \cdot (0,05 + 0,05) = 2,3.$$

Код Хаффмана є найбільш економічним. Для будь-якого іншого методу кодування символів деякого алфавіту середня кількість елементарних сигналів, які доводяться на один символ, не може бути меншою ніж при кодуванні за методом Хаффмана. Звідси, впливає й те, що для будь-яких двох кодів Хаффмана середня довжина кодового позначення повинна бути однаковою [6].

1.2.4 Оцінка ефективності коду Шеннона - Фано

Оцінимо ефективність стиснення даних з використанням коду Шеннона-Фано на прикладі текстових файлів різної довжини, а також виконуваних файлів.

Через те, що алгоритми кодування / декодування за методами Шеннона - Фано та Хаффмана мають схожу структуру, виявилось можливим реалізувати їх аналіз за допомогою однієї програми. При цьому очевидним є виграш у розмірі машинного коду, оскільки обидва методи, як при кодуванні, так й при декодуванні, використовують підпрограми із загального модуля. Результати розрахунків наведені у таблицях 1.12 та 1.13.

За результатами розрахунків побудуємо діаграми порівняння коефіцієнтів стиснення та часу кодування повідомлення за допомогою коду Шеннона - Фано та коду Хаффмана для різних типів файлів (рис. 1.3, 1.4).

Таблиця 1.12 - Результати кодування файлів за методом Шеннона-Фано

Тип файлу, ім'я	Вихідна довжина [Байт]	Ентропія [біт/симв]	Довжина після стиснен- ня [Байт]	Середня довжина кодових комбінацій [біт/симв]	Коефіці- єнт стис- нення [%]	Час коду- вання [с]
arch.exe	43470	6.9596	39155	7.1065	90.07	1.26
turbo.exe	402474	6.9309	356963	7.0745	88.69	11.75
opt1.doc	196096	3.9661	100025	4.0378	51.01	3.74
assem.txt	101753	5.1462	68438	5.2980	67.26	2.41
user_gui.txt	603074	4.7448	370046	4.8949	61.36	13.24
prog_gui.txt	1037826	4.5504	613750	4.7229	59.14	22.25

Таблиця 1.13 - Результати кодування файлів за методом Хаффмана

Тип файлу, ім'я	Вихідна довжина [Байт]	Ентропія [біт/симв]	Довжина після стиснен- ня [Байт]	Середня довжина кодових комбінацій [біт/симв]	Коефіці- єнт стис- нення [%]	Час коду- вання [с]
arch.exe	43470	6.9596	38511	6.9880	88.59	1.26
turbo.exe	402474	6.9309	351399	6.9639	87.31	11.59
opt1.doc	196096	3.9661	98813	3.9883	50.39	3.74
assem.txt	101753	5.1462	67009	5.1857	65.85	2.36
user_gui.txt	603074	4.7448	361749	4.7848	59.98	13.01
prog_gui.txt	1037826	4.5504	597328	4.5964	57.56	21.87

Із порівняння результатів стиснення даних за методами Шеннона - Фано та Хаффмана можна зробити висновок, що метод Хаффмана в усіх випадках дає коефіцієнт стиснення більш кращий, ніж метод Шеннона - Фано.

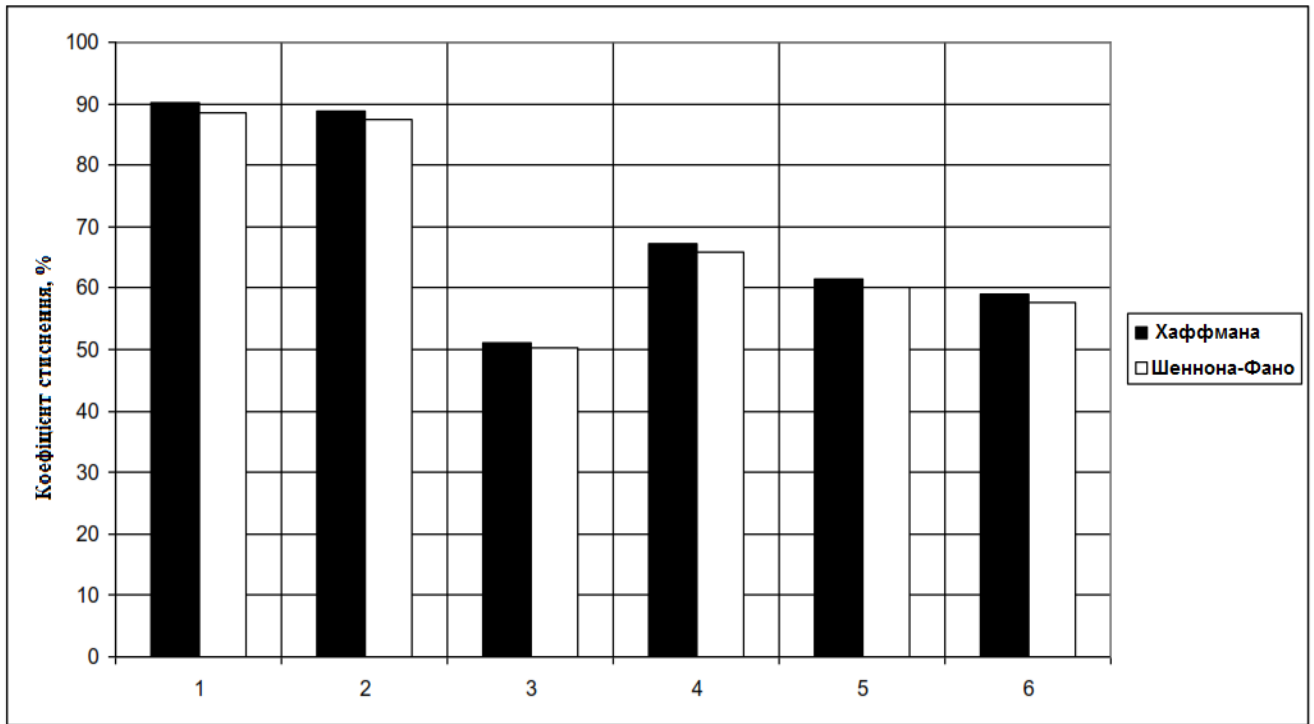


Рисунок 1.3 - Порівняння коефіцієнтів стиснення повідомлень при кодуванні за методами Шеннона - Фано та Хаффмана

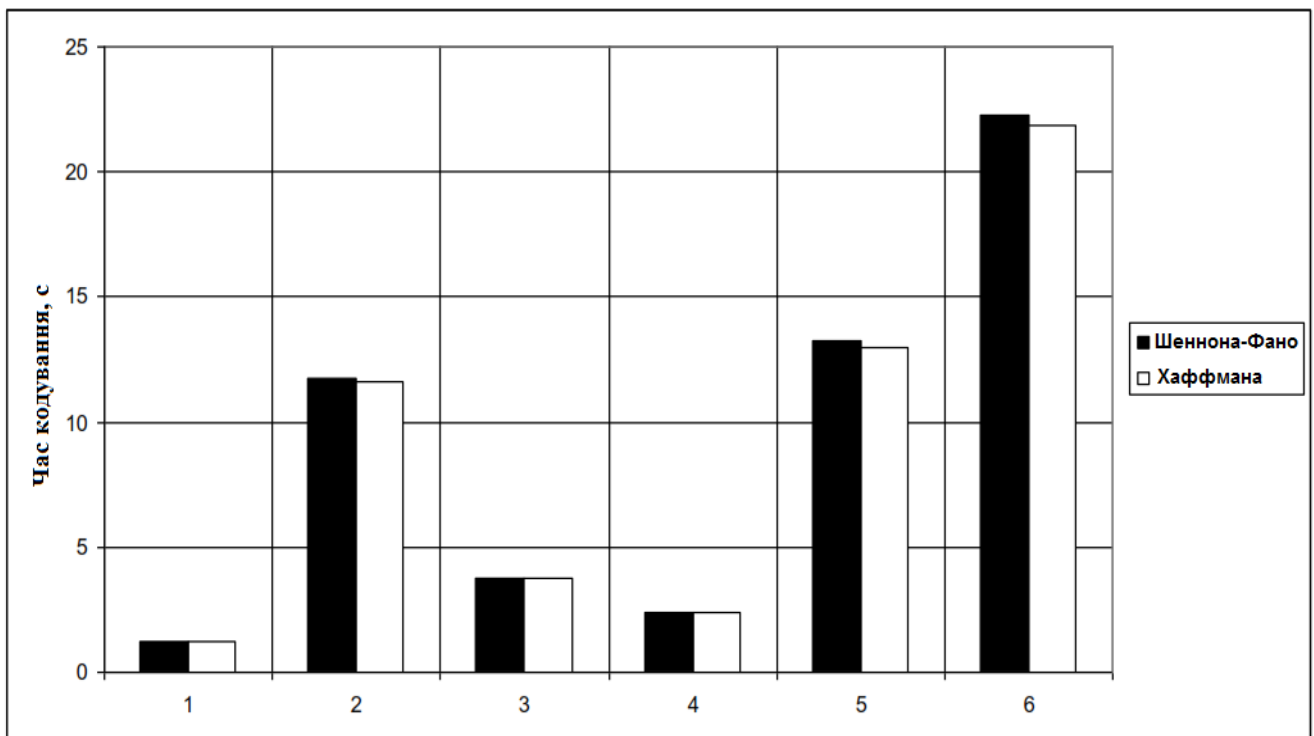


Рисунок 1.4 - Порівняння часу кодування при кодуванні за методами Шеннона - Фано та Хаффмана

Изм.	Лист	№ докум.	Подпись	Дата

Позначення на рисунках:

- 1 – arch.exe з вихідною довжиною 43470 байт;
- 2 – turbo.exe з вихідною довжиною 402474 байт;
- 3 – opt1.doc з вихідною довжиною 196096 байт;
- 4 – assem.txt з вихідною довжиною 101753 байт;
- 5 – user_gui.txt з вихідною довжиною 603074 байт;
- 6 – prog_gui.txt з вихідною довжиною 1037826 байт.

Проаналізувавши отримані при виконанні роботи результати, необхідно відзначити наступне, що метод Шеннона - Фано не гарантує однозначності побудови коду, оскільки при розбиванні таблиці ймовірностей на підгрупи можливі різні варіанти розбивання.

Методика Хаффмана вільна від цього недоліку й гарантує однозначність побудови коду з найменшим числом символів на символ алфавіту повідомлення при заданому розподілі ймовірностей, чим і обумовлене його застосування в багатьох відомих архіваторах.

З рисунку 1.4 видно, що якщо порівнювати код Шеннона - Фано та код Хаффмана за часом кодування, то результати не суттєво відрізняються для цих двох кодів. Порівнюючи файли з невеликою довжиною можна сказати, що, час кодування збігається у двох кодах.

Код Шеннона - Фано поступається по деяких характеристиках для коду Хаффмана, але не суттєво. Але перевагою коду Шеннона - Фано є те, що він має простий алгоритм побудови коду.

Побудуємо діаграми порівняння коефіцієнту стиснення й часу кодування txt файлів за методами Шеннона - Фано та Хаффмана (рис. 1.5, 1.6).

На рисунку 1.5 для txt файлів видно, що час кодування коду Шеннона - Фано не поступається суттєво коду Хаффмана.

Зворотним процесом до процедури кодування - процедура декодування.

Розрахуємо час декодування для кодів Шеннона - Фано та Хаффмана і зведемо результати у таблицях 1.13, 1.14.

					ЦЗДВН 8.171.00.10.327 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		38

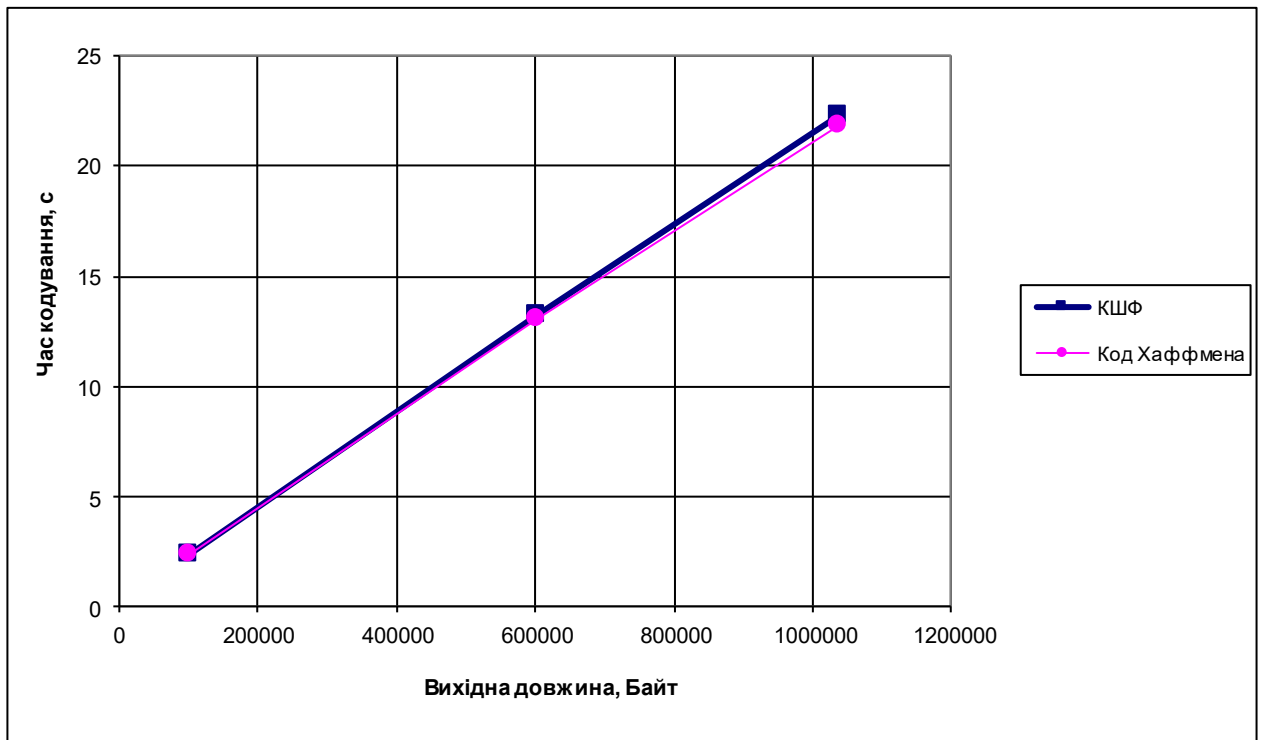


Рисунок 1.5 - Порівняння часу кодування txt файлів при кодуванні за методами Шеннона - Фано та Хаффмана

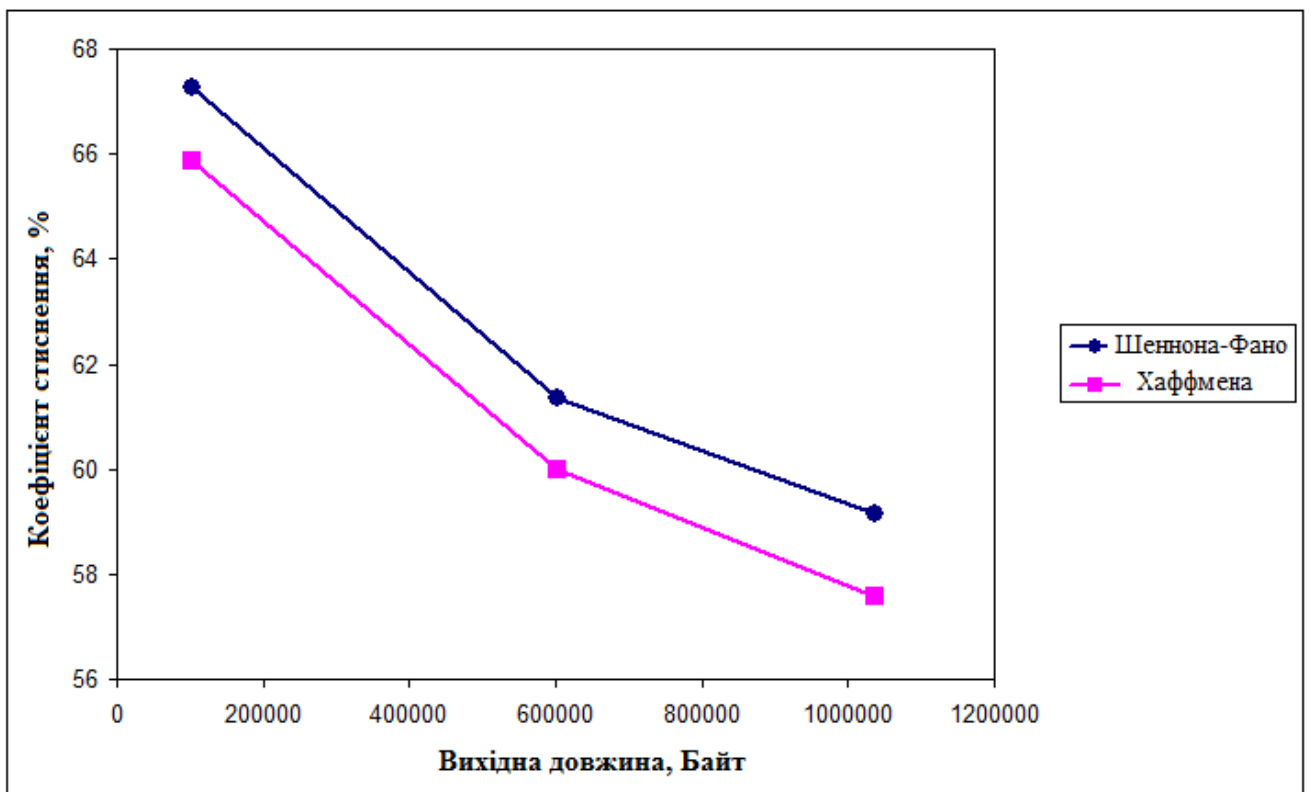


Рисунок 1.6 - Порівняння коефіцієнту стиснення txt файлів при кодуванні за методами Шеннона - Фано та Хаффмана

Таблиця 1.13 - Результати декодування файлів, стиснених за методом Шеннона - Фано

Тип файлу, ім'я	Вихідна довжина [Байт]	Довжина після декодування [Байт]	Час декодування [с]
arch.exe	39155	43470	1.49
turbo.exe	356963	402474	13.56
opt1.doc	100025	196096	4.07
assem.txt	68438	101753	2.69
user_gui.txt	370046	603074	14.61
prog_gui.txt	613750	1037826	24.39

Таблиця 1.14 - Результати декодування файлів, стиснених за методом Хаффмана

Тип файлу, ім'я	Вихідна довжина [Байт]	Довжина після декодування [Байт]	Час декодування [с]
arch.exe	38511	43470	1.48
turbo.exe	351399	402474	13.35
opt1.doc	98813	196096	4.01
assem.txt	67009	101753	2.58
user_gui.txt	361749	603074	14.50
prog_gui.txt	1037826	1037826	23.83

За результатами розрахунків побудуємо діаграму порівняння часу декодування повідомлення за допомогою кодів Шеннона - Фано та Хаффмана для різних типів файлів (рисунок 1.7).

На рисунку видно, що час декодування не є суттєво більшим для коду Шеннона - Фано, в порівнянні з кодом Хаффмана, тому що методика Шеннона-Фано не гарантує однозначності побудови коду, оскільки при розбиванні таблиці ймовірностей на підгрупи можливі різні варіанти розбивання.

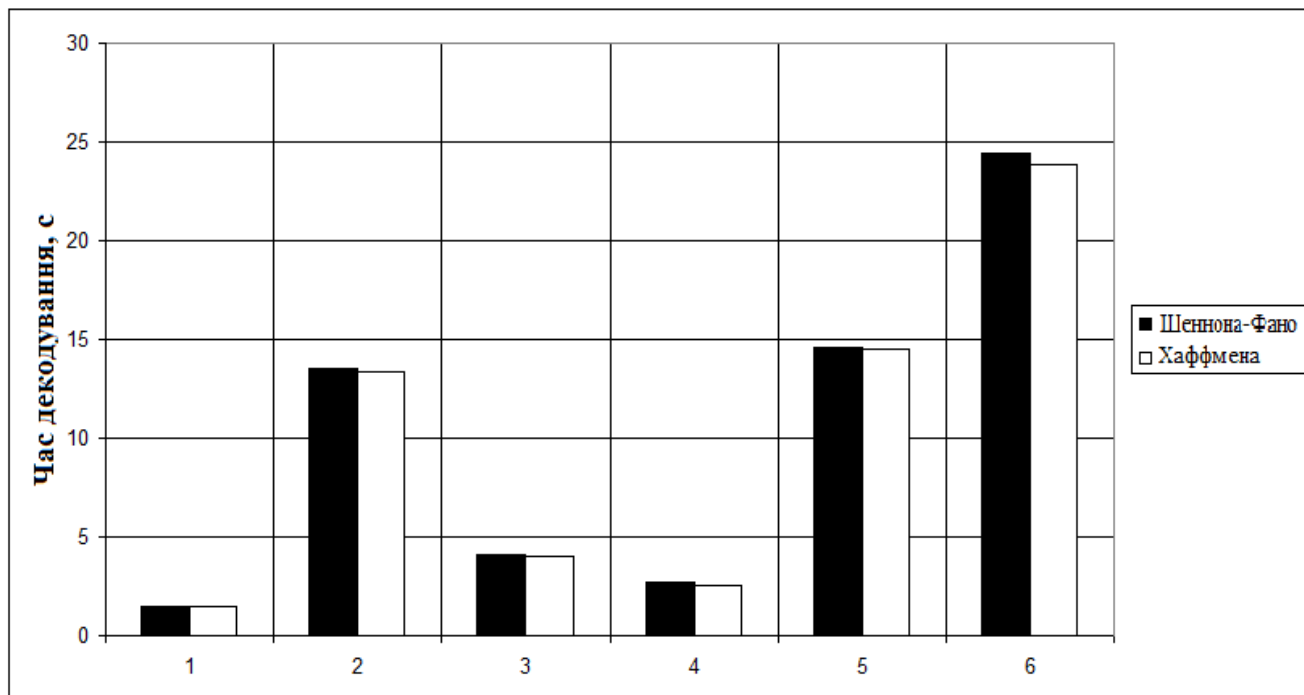


Рисунок 1.7 - Порівняння часу декодування повідомлень за методами Шеннона-Фано та Хаффмана

Позначення на рисунку:

- 1 – arch.exe з вихідною довжиною 43470 байт;
- 2 – turbo.exe з вихідною довжиною 402474 байт;
- 3 – opt1.doc з вихідною довжиною 196096 байт;
- 4 – assem.txt з вихідною довжиною 101753 байт;
- 5 – user_gui.txt з вихідною довжиною 603074 байт;
- 6 – prog_gui.txt з вихідною довжиною 1037826 байт.

Проаналізуємо ефективність стиснення файлів різного типу за допомогою алгоритмів Шеннона-Фано, Хаффмана, архіваторів ZIP і RAR. Результати досліджень наведені у таблиці 1.15.

Порівнюючи ефективність стиснення даних відомими архіваторами RAR і ZIP з ефективністю стиснення за допомогою методів Шеннона-Фано та Хаффмана необхідно зазначити, що архіватори RAR і ZIP забезпечують суттєво кращий коефіцієнт стиснення. Причиною цього є те, що при кодуванні за методами Шеннона - Фано та Хаффмана не враховувалася можлива кореляція між символами повідомлення. В архіваторах RAR і ZIP при стисненні даних ця кореляція враховується, що й забезпечує більш високий ступінь стиснення.

Таблиця 1.15 - Порівняння ефективності стиснення файлів за допомогою алгоритмів Шеннона-Фано, Хаффмена, архіваторів ZIP і RAR

Тип, ім'я файлу	Шеннона-Фано		RAR		ZIP		Хаффмана	
	Коеф. стиснення [%]	Час [с]	Коеф. стиснення [%]	Час [с]	Коеф. стиснення [%]	Час [с]	Коеф. стиснення [%]	Час [с]
arch.exe	90.07	1.26	58.20	1	60.46	1	88.59	1.26
turbo.exe	88.69	11.75	49.98	1	52.10	1	87.31	11.59
opt1.doc	51.01	3.74	19.54	1	20.55	1	50.39	3.74
assem.txt	67.26	2.41	22.16	1	22.72	1	65.85	2.36
user_gui.txt	61.36	13.24	25.18	2	26.12	2	59.98	13.01
prog_gui.txt	59.14	22.25	22.17	3	23.06	3	57.56	21.87

Визначимо та проаналізуємо основні характеристики повідомлень при кодуванні за допомогою алгоритмів Шеннона-Фано та Хаффмена RAR архівів. Результати досліджень наведені в таблицях 1.16, 1.17.

Таблиця 1.16 - Результати кодування RAR архівів за допомогою коду Шеннона-Фано

Тип файлу, ім'я	Вихідна довжина [байт]	Ентропія перед кодуванням [біт/симв]	Довжина після стиснення [байт]	Середня довжина кодових комбінацій [біт/симв]	Коефіцієнт стиснення [%]	Ентропія після кодування [біт/симв]
arch.rar	25301	7.9917	26246	8.1283	103.74	7.9409
turbo.rar	201146	7.9984	206260	8.1616	102.54	7.9432
opt1.rar	38326	7.9830	39470	8.1262	102.98	7.9459
assem.rar	22550	7.9906	23428	8.1202	103.89	7.9403
user_gui.rar	151835	7.9983	155964	8.1622	102.72	7.9439
prog_gui.rar	230098	7.9989	235515	8.1518	102.35	7.9472

Таблиця 1.17 - Результати кодування RAR архівів за допомогою методу Хаффмана

Тип файлу, ім'я	Вихідна довжина [байт]	Ентропія перед кодуванням [біт/симв]	Довжина після стиснення [байт]	Середня довжина кодових комбінацій	Коефіцієнт стиснення [%]	Ентропія після кодування [біт/симв]
arch.rar	25301	7.9917	25840	8.0000	102.13	7.9782
turbo.rar	201146	7.9984	202197	8.0000	100.52	7.9967
opt1.rar	38326	7.9830	38840	7.9947	101.34	7.9828
assem.rar	22550	7.9906	23089	8.0000	102.39	7.9750
user_gui.rar	151835	7.9983	152886	8.0000	100.69	7.9957
prog_gui.rar	230098	7.9989	231149	8.0000	100.46	7.9977

Загальним та очевидним недоліком застосування нерівномірних оптимальних кодів є зниження надійності передачі та зберігання інформації, тому що у випадку перекодування або втрати хоча б одного біту кодової комбінації при декодуванні буде загублений не тільки поточний символ (який декодується), але й усі наступні символи вихідного повідомлення.

Розглянемо приклад побудови коду Шеннона - Фано. Для цього розрахуємо первісні ймовірності для нерівноймовірних символів алфавіту. Виконаємо нормування зазначених імовірностей. Розрахуємо ентропію алфавіту з рівноймовірних символів. Виконаємо розрахунок ентропії алфавіту з нерівноймовірних символів. З врахуванням заданих довжин символів обчислимо швидкість передачі та надмірність коду. Побудуємо оптимальний код за методом Шеннона-Фано.

Ймовірності появи символів первинного алфавіту наведені у таблиці 1.18.

Визначення кількості інформації на символ повідомлення, складеного з даного алфавіту. Кількість інформації на символ повідомлення для символів даного алфавіту, які зустрічаються з рівними ймовірностями:

$$H_{\max} = \log_2 24 = 4,5850 \text{ біт/символ.}$$

Таблиця 1.18 – Ймовірності появи символів первинного алфавіту

$p_i = p_i / \left(\sum_{i=1}^k p_i \right)$
$p_1=0,0417$
$p_2=0,0018$
$p_3=0,0020$
$p_4=0,0022$
$p_5=0,0024$
$p_6=0,0026$
$p_7=0,0029$
$p_8=0,0033$
$p_9=0,0037$
$p_{10}=0,0042$
$p_{11}=0,0048$
$p_{12}=0,0055$
$p_{13}=0,0064$
$p_{14}=0,0076$
$p_{15}=0,0091$
$p_{16}=0,0111$
$p_{17}=0,0139$
$p_{18}=0,0179$
$p_{19}=0,0238$
$p_{20}=0,0333$
$p_{21}=0,0500$
$p_{22}=0,0833$
$p_{23}=0,1667$
$p_{24}=0,5000$
$\sum_{i=1}^k p_i = 1$

Кількість інформації на символ повідомлення для символів даного алфавіту, які зустрічаються в повідомленні з різними ймовірностями:

$$H = - (0,0417 \cdot \log_2 0,0417 + 0,0018 \cdot \log_2 0,0018 + 0,0020 \cdot \log_2 0,0020 + 0,0022 \cdot \log_2 0,0022 + 0,0024 \cdot \log_2 0,0024 + 0,0026 \cdot \log_2 0,0026 + 0,0029 \cdot \log_2 0,0029 + 0,0033 \cdot \log_2 0,0033 + 0,0037 \cdot \log_2 0,0037 + 0,0042 \cdot \log_2 0,0042 + 0,0048 \cdot \log_2 0,0048 + 0,0055 \cdot \log_2 0,0055 + 0,0064 \cdot \log_2 0,0064 + 0,0076 \cdot \log_2 0,0076 + 0,0091 \cdot \log_2 0,0091 + 0,0111 \cdot \log_2 0,0111 + 0,0139 \cdot \log_2 0,0139 + 0,0179 \cdot \log_2 0,0179 + 0,0238 \cdot \log_2 0,0238 + \dots)$$

$0,0333 \cdot \log_2 0,0333 + 0,0500 \cdot \log_2 0,0500 + 0,0833 \cdot \log_2 0,0833 + 0,1667 \cdot \log_2 0,1667 + 0,5000 \cdot \log_2 0,5000 = 2,6409$ біт/символ.

Надмірність повідомлень, складених з даного алфавіту:

$$D = 1 - (H/H_{\max}) = 1 - (2,6409 / 4,5850) = 0,4240.$$

Таблиця 1.19 - Побудова оптимального коду

Символ	Ймовірність появи символів	Кодові слова	Кількість розрядів у кодовому слові	$P_i \cdot l_i$
A[1] (p24)	0,5000	0	1	0,5
A[2] (p23)	0,1667	111	3	0,50001
A[3] (p22)	0,0833	110	3	0,2500
A[4] (p21)	0,0500	1000	4	0,2000
A[5] (p1)	0,0417	10011	5	0,2083
A[6] (p20)	0,0333	10010	5	0,1667
A[7] (p19)	0,0238	101111	6	0,1429
A[8] (p18)	0,0179	1011100	7	0,1250
A[9] (p17)	0,0139	101101	6	0,0833
A[10] (p16)	0,0111	101110	6	0,0667
A[11] (p15)	0,0091	1010011	7	0,0636
A[12] (p14)	0,0076	10100100	8	0,0606
A[13] (p13)	0,0064	1010001	7	0,0449
A[14] (p12)	0,0055	1010011	7	0,0385
A[15] (p11)	0,0048	10101111	8	0,0381
A[16] (p10)	0,0042	101011100	9	0,0375
A[17] (p9)	0,0037	10101101	8	0,0294
A[18] (p8)	0,0033	10101110	8	0,0261
A[19] (p7)	0,0029	10101010	8	0,0234
A[20] (p6)	0,0026	101010111	9	0,0237
A[21] (p5)	0,0024	101010110	9	0,0214
A[22] (p4)	0,0022	10101000	8	0,0173
A[23] (p3)	0,0020	101010011	9	0,0178
A[24] (p2)	0,0018	101010010	9	0,0163

На першому етапі множину символів розташуємо в порядку спадання ймовірностей їх появи. Далі розіб'ємо дану множину на дві групи таким чином, щоб сумарні ймовірності повідомлень обох груп були по можливості однаковими.

Таблиця 1.20 - Побудова оптимального коду за методом Шеннона-Фано

Символ	Ймовірності символів	Кодова комбінація	Додавання ймовірностей																
P1	0,010	11111111	0,520	0,277	0,147	0,086	0,051	0,035	0,022	0,010									
P2	0,012	11111110													0,012				
P3	0,013	11111110												0,013					
P4	0,016	1111110												0,016					
P5	0,017	1111101												0,017					
P6	0,018	1111100												0,018					
P7	0,019	111011												0,019					
P8	0,020	111010												0,020					
P9	0,022	11100												0,022					
P10	0,023	110111												0,023					
P11	0,025	110110												0,025					
P12	0,026	11010												0,026					
P13	0,027	11001												0,027					
P14	0,029	11000												0,029					
P15	0,030	101111	0,243	0,243	0,130	0,095	0,062	0,030											
P16	0,032	101110											0,032						
P17	0,033	10110											0,033						
P18	0,035	1010											0,035						
P19	0,036	10011											0,036						
P20	0,038	10010											0,038						
P21	0,039	1000											0,039						
P22	0,040	011111						0,471	0,262	0,168	0,124	0,081	0,040						
P23	0,041	011110																0,041	
P24	0,043	01110																0,043	
P25	0,044	0110											0,044						
P26	0,046	0101											0,046						
P27	0,048	0100											0,048						
P28	0,050	00111											0,050						
P29	0,051	00110											0,051						
P30	0,053	0010	0,209	0,209	0,154	0,101	0,053												
P31	0,055	000																	

Але оскільки рівність сумарних ймовірностей в групах не досягається, то розподіл на групи виконаємо таким чином, щоб у верхній частині (в першій групі) залишалися символи, сумарна ймовірність яких буде менше ніж сумарна ймовірність символів у нижній частині (в другій групі). Першій групі привласнюємо символ 0, а другій групі - символ 1. Кожну з сформованих підгруп ділимо на дві частини таким чином, щоб сумарні ймовірності утворених на наступному етапі підгруп були по можливості однаковими. Першим групам кожної з підгруп знову привласнюємо 0, а другим 1. Таким чином, ми одержуємо другі розряди кодових комбінацій. Потім кожну з чотирьох груп ще раз розподілимо на частини до тих пір, поки в кожній з підгруп не залишиться по одному символу.

Таблиця 1.21 - Оптимальний код (отриманий результат)

Символ	Ймовірності появи символів	Кодове слово	Кількість розрядів у кодовому слові	$p_i \cdot l_i$
1	2	3	4	5
P1	0,055	000	3	0,165
P2	0,053	0010	4	0,212
P3	0,051	00110	5	0,255
P4	0,050	00111	5	0,250
P5	0,048	0100	4	0,192
P6	0,046	0101	4	0,176
P7	0,044	0110	4	0,114
P8	0,043	01110	5	0,215
P9	0,041	011110	6	0,246
P10	0,040	011111	6	0,240
P11	0,039	1000	4	0,156
P12	0,038	10010	5	0,190
P13	0,036	10011	5	0,180
P14	0,035	1010	4	0,140
P15	0,033	10110	5	0,165
P16	0,032	101110	6	0,192
P17	0,030	101111	6	0,180
P18	0,029	11000	5	0,145
P19	0,027	11001	5	0,135

Изм.	Лист	№ докум.	Подпись	Дата

Продовження таблиці 1.21

1	2	3	4	5
P20	0,026	11010	5	0,130
P21	0,025	110110	6	0,150
P22	0,023	110111	6	0,138
P23	0,022	11100	5	0,110
P24	0,020	111010	6	0,120
P25	0,019	111011	6	0,114
P26	0,018	111100	6	0,108
P27	0,017	111101	6	0,102
P28	0,016	111110	6	0,096
P29	0,013	1111110	7	0,091
P30	0,012	11111110	8	0,096
P31	0,010	11111111	8	0,080

Проаналізувавши отримані результати необхідно відзначити наступне.

1. Методика Шеннона-Фано не гарантує однозначності побудови коду, оскільки при розбиванні таблиці ймовірностей на підгрупи можливі різні варіанти розбивання.

2. Методика Хаффмана вільна від цього недоліку і гарантує однозначність побудови коду з найменшою кількістю розрядів на символ алфавіту повідомлення при заданому розподілі ймовірностей, чим й обумовлене його застосування в багатьох відомих архіваторах.

3. Із аналізу результатів стиснення даних за методами Шеннона-Фано та Хаффмана стає відомим, що метод Хаффмана у всіх випадках дає коефіцієнт стиснення кращий, ніж метод Шеннона-Фано. Це викликано вже згадуваною неоднозначністю розбивання на групи при кодуванні повідомлень з використанням алгоритму Шеннона-Фано.

4. Порівнюючи ефективність стиснення даних відомими архіваторами RAR і ZIP зі стисненням за методами Шеннона-Фано та Хаффмана, необхідно відзначити, що архіватори RAR і ZIP забезпечують суттєво кращий коефіцієнт стиснення. Причиною цьому є та обставина, що при кодуванні за методами Шеннона-Фано та Хаффмана не враховується можлива кореляція між символами повідомлення. В архіваторах RAR і ZIP при стисненні даних кореляція враховується, що й забезпечує більш високий ступінь стиснення повідомлень.

5. Загальним та очевидним недоліком застосування нерівномірних оптимальних кодів є зниження надійності передачі та зберігання інформації, тому що у випадку перекручення або втрати хоча б одного біту кодової комбінації при декодуванні буде загублена не тільки поточний символ (який декодується), але й усі наступні символи вихідного повідомлення.

6. Аналіз характеристик ефективності кодів Шеннона - Фано та Хаффмана показує, що характеристики цих кодів не суттєво відрізняються, але код Шеннона - Фано має простий алгоритм побудови коду в порівнянні з кодом Хаффмана. Це є його основною перевагою.

1.3 Розроблення електронного пристрою з використанням результатів дослідження

1.3.1 Обґрунтування алгоритму функціонування та структурної схеми проєктованого пристрою

Схема алгоритму стиснення інформації на основі коду Шеннона-Фано представлена на рисунку 1.8.

Пристрій стиснення інформації функціонує згідно наступного алгоритму [2].

1. Одержується вихідне повідомлення із заданим первинним алфавітом.
2. Розраховуються ймовірності появи символів первинного алфавіту.
3. Для зручності розташовуються усі наявні n символів в один стовпчик у порядку спадання ймовірностей.

4. Потім усі ці символи потрібно розбити на дві групи (першу та другу) таким чином, щоб сумарна ймовірність символів першої групи була найбільш близькою до сумарної ймовірності символів другої групи.

5. Для символів першої групи в якості першого розряду (цифри кодового позначення використовується 1, а для символів другої групи - 0.

6. Далі кожен з двох груп подібним чином знову потрібно розділити на дві частини й у якості другої цифри кодового позначення потрібно використовувати 1 або 0 залежно від того, належить символ до першої або до другої підгрупи.

7. Потім кожен з груп, яка містить більше, ніж один символ знову потрібно розділити на дві частини з якнайбільше близькими сумарними ймовірностями й т.п. Процес повторюється доти, поки не отримаємо групи, кожна з яких містить по одному символу.

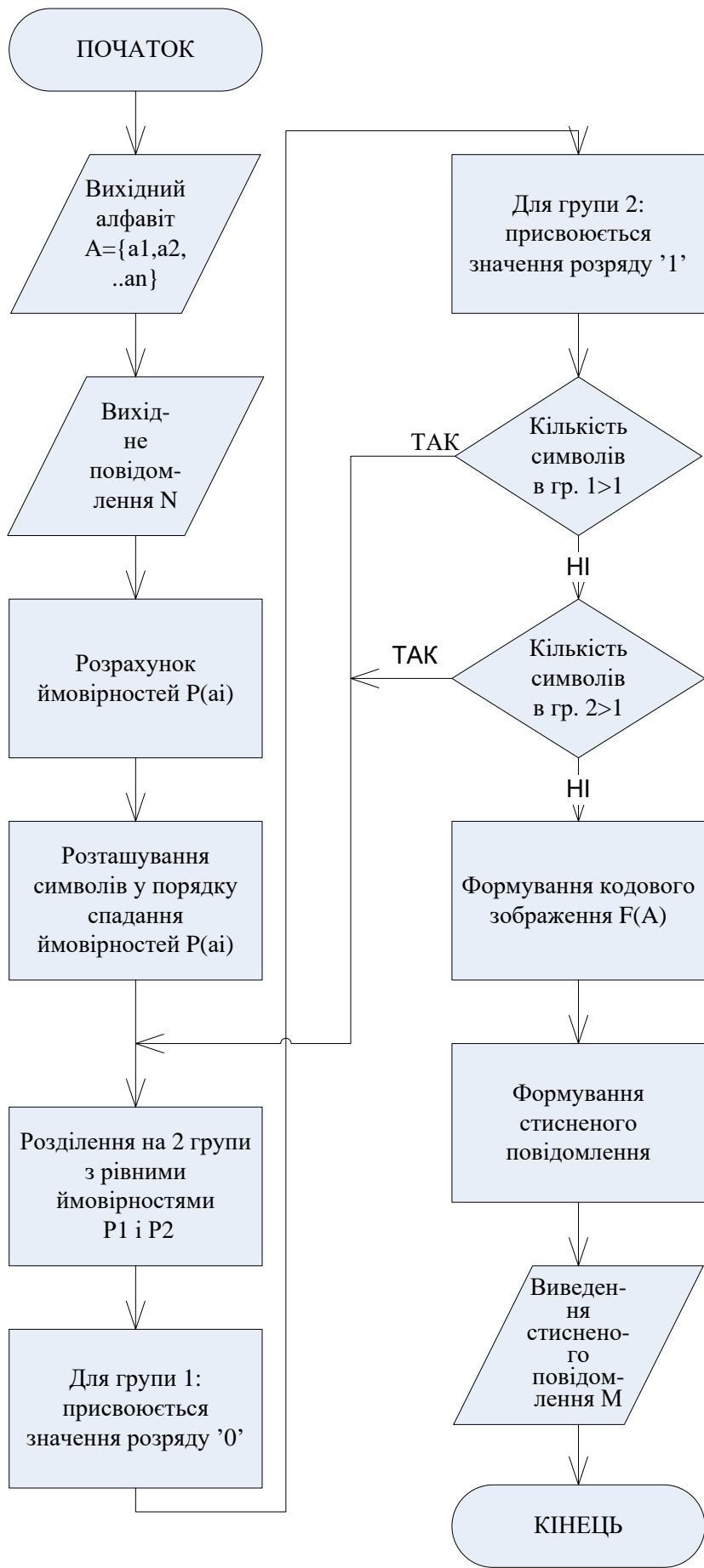


Рисунок 1.8 - Схема алгоритму стиснення інформації

Изм.	Лист	№ докум.	Подпись	Дата

Наприклад, якщо алфавіт містить усього шість символів, ймовірності яких (у порядку спадання) дорівнюють 0,4, 0,2, 0,2, 0,1, 0,05 і 0,05, то на першому етапі розподілу символів на групи ми виділимо лише один перший символ (1 група), залишивши в другій групі всі інші символи.

Далі, другий символ складе першу підгрупу 2-ої групи, друга підгрупа тієї ж групи, що містить чотири символи, які залишилися, буде й далі послідовно розділена на частині таким чином, щоб кожен раз перша частина буде складатися лише з одного символу (табл. 1.22).

Таблиця 1.22 - Розділення символів на підгрупи

№ символу	Ймовірність	Розділення на підгрупи (римські цифри позначають номери груп і підгруп)					Кодове позначення		
1	0,4	} I					1		
2	0,2	} II	} I				01		
3	0,2			} II	} I			001	
4	0,1		} II			} I			0001
5	0,05						} II	} I	
6	0,05								00000

Аналогічно попередньому прикладу розглянемо приклад «алфавіту», що містить 18 символів, які мають наступні ймовірності: 0,3; 0,2; 0,1 (2 символи); 0,05; 0,03 (5 символів); 0,02(2 символи); 0,01 (6 символів) (табл. 1.23).

Основний принцип, покладений в основу кодування за методом Шеннона – Фано полягає в тому, що на кожному етапі виконання алгоритму кожен символ кодового позначення вибирається таким чином, щоб кількість інформації, що міститься в ньому, була найбільшою. Тобто щоб незалежно від значень усіх попередніх цифр ця цифра приймала одне з можливих для неї значення (0 або 1) по можливості з однаковою ймовірністю. Кількість цифр (розрядах) в різних кодових позначеннях (кодових комбінаціях) при цьому виявляється різною (зокрема, у другому прикладі кількість розрядів змінюється від двох до семи, тобто код Шеннона – Фано є нерівномірним. Будь-яке кодове позначення не може бути початком іншого, більш довгого позначення. Тому закодоване повідомлення завжди може бути однозначно декодовано [2, 8].

Таблиця 1.23 - Розділення символів на підгрупи

№ символу	Ймовірність	Розділення на підгрупи				Кодове позначення		
1	0,3	} I	} I			11		
2	0,2			} II		10		
3	0,1	} II	} I	} I		011		
4	0,1			} II		0101		
5	0,05			} II		0100		
6	0,03			} I	} I		00111	
7	0,03		} II		00110			
8	0,03		} I		00101			
9	0,03		} II		00100			
10	0,03		} II		} I	} I		00011
11	0,02					} II		000101
12	0,02			} II		000100		
13	0,01	} II		} I		000011		
14	0,01			} II		0000101		
15	0,01			} II		0000100		
16	0,01			} I		000001		
17	0,01			} II		0000001		
18	0,01		} II		0000000			

У результаті, середнє значення довжини таких кодових комбінацій буде лише трохи більшим, ніж мінімальне значення безумовної ентропії H вихідного повідомлення.

Структурна схема пристрою (рис. 1.9) складається із блоку джерела інформації (ДІ) звідки надходить повідомлення в блок зберігання вихідного повідомлення, цей блок служить для зберігання повідомлення, що надійшло з ДІ. Розрахунок імовірностей відбувається в блоці підрахунку ймовірностей. Для зручності розташовують усі наявні n символів первинного алфавіту в один стовпчик у порядку спадання їх ймовірностей. Це відбувається у формувачі послідовності повідомлення.

Рисунок 1.9 - Схема электрична структурна пристрою стиснення даних

					ЦЗДВН 8.171.00.10.327 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		53

Потім усі символи в блоці розбивання на групи потрібно розбити на дві групи (першу та другу) таким чином, щоб сумарна ймовірність символів першої групи була найбільш близькою до сумарної ймовірності символів другої групи. Після цього потрібно в блоці зберігання першої групи й у блоці зберігання другої групи зберегти сформовані групи символів.

Для символів першої групи в якості першої цифри кодового позначення в блоці формування значення розряду привласнюється цифра 1, а для символів другої групи - цифра 0. Далі, кожен з двох груп подібним чином знову потрібно розділити на дві частини й у якості другої цифри кодового позначення потрібно записати цифру 1 або 0 залежно від того, належить символ до першої або до другої з цих підгруп.

У блоці підрахунку кількості символів визначається дотримання умові, щоб кожна з груп, якщо вона містить більше, ніж один символ, повинна знову бути розділена на дві частини з якнайбільше близькими сумарними ймовірностями й т.п. Процес повторюється доти, поки не утворяться групи, кожна з яких буде містити по одному символу.

Після цього формується кодове зображення та заноситься в блок зберігання.

Структурна схема пристрою стиснення інформації на основі Шеннона-Фано показано на рисунку 1.9.

1.3.2 Розроблення схеми електричної функціональної проектного пристрою

Під розробленням схеми електричної функціональної пристрою розуміється визначення функціонального складу блоків (модулів), які є складовими проектного пристрою, розрахунок та обґрунтування технічних вимог до зазначених блоків і встановлення необхідних електричних зв'язків між ними [12].

Для керування процесами перетворень та обчислень до складу проектного пристрою повинен входити мікропроцесор або мікропроцесорний контролер, який повинен надавати мікро-ЕОМ, яка працює по програмі та володіє можливостями підключення до неї пристроїв введення та виведення даних [13].

Схема електрична функціональна проектного пристрою наведена на рисунку 1.10.

					ЦЗДВН 8.171.00.10.327 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		54

Рисунок 1.10 - Схема електрична функціональна пристрою стиснення даних

					ЦЗДВН 8.171.00.10.327 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		55

Проектований контролер складається з мікропроцесорного блоку, до складу якого входить центральний процесор, буферні регістри, шинний формувач, мультиплексор. Мікропроцесорний блок управляє роботою контролера, формуючи керуючі сигнали, виконує усі необхідні обчислення.

До функцій оперативного запам'ятовуючого пристрою (ОЗП) та постійного запам'ятовуючого пристрою (ПЗП) входить збереження інформації [12, 13].

Для керування зовнішніми пристроями, а також для забезпечення необхідної швидкості введення отриманого коду в пристрій пам'яті, необхідно застосувати інтерфейсний блок.

Центральний процесорний модуль є основним блоком контролера. Він забезпечує керування та синхронізацію роботи всього пристрою, забезпечує прийом, виведення, зберігання та оброблення даних, які надходять по системній шині [13].

Блоком пам'яті в проектованому пристрої є ПЗП та ОЗП. В ПЗП зберігаються коди програми, а в ОЗП – проміжні, вхідні та вихідні дані.

Для керування зовнішніми пристроями, а також для забезпечення необхідної швидкості введення отриманого коду у блок пам'яті, необхідно застосувати паралельний інтерфейс, який дозволяє організовувати введення/виведення паралельної інформації різного формату.

Для тимчасового зберігання вхідного та стисненого повідомлення застосовуються буферні регістри.

Контролер, як й будь-яку мікропроцесорну систему (МПС), слід будувати по модульному (блоковому) принципу з шинною організацією зв'язків між блоками. Фізично шина адреси (ША), шина даних (ЩД) та шина сигналів управління (ШСУ) являють собою паралельні провідники, що призначені для передачі багаторозрядних цифрових кодів або у двох, або в одному напрямку. Достоїнство шинних зв'язків – це суттєве зменшення кількості сполучних проводів. Однак, наявність шинних зв'язків висуває певні вимоги до підключених до них блоків та вузлів. У першу чергу по виходу блоки та вузли обов'язково повинні мати високоімпедансний стан (Z - стан). Можна навантажувати на шини також мікросхеми, які допускають об'єднання по виходу (схеми з відкритим колектором). Крім того, при роботі на шину блоки та вузли повинні мати достатню навантажувальну здатність [13].

1.3.3 Розроблення та розрахунок принципових електричних схем вузлів та блоків пристрою

1.3.3.1 Вибір елементної бази.

Для побудови пристрою стиснення даних необхідно вибрати серію мікросхем, на якій будуть реалізовані блоки пристрою.

Оптимальним для розроблювального пристрою буде мікропроцесор КР1821ВМ85А, перевагами якого є [14]:

- низька вартість;
- орієнтування на роботу в складі мікроконтролерів;
- програмна сумісність з мікропроцесором КР580ВМ80А;
- наявність одного джерела живлення +5В;
- наявність вбудованого генератора тактових імпульсів;
- вбудований системний контролер;
- вбудований контролер переривання.

Мікропроцесор КР1821ВМ85А виконаний по КМОП технології, має з'єднані шину даних та шину адреси.

До складу центрального процесорного модуля (ЦПМ) входять процесор КР1821ВМ85А, два буферних регістри КР580ІР82, двохнаправлений шинний формувач КР580ВА86, мультиплексор К555КП11, зовнішній резонатор (ZQ, C₂), схема формування скидання (R₁, C₁, S).

У якості ПЗП використовується БІС К573РФ5. Такий ПЗП може бути реалізований на базі однієї мікросхеми пам'яті з ультрафіолетовим стиранням, що має організацію 2к*8 й має час вибірки адреси 450 нс. Ця інтегральна мікросхема (ІМС) має 11 адресних входів, 8 виходів даних, входи дозволу програмування WE, вибирання кристалу CE, дозволу виведення даних з виходів OE.

У якості проєктованого ОЗП вибираємо статичний ОЗП КР537РУ8 обсягом 2 Кбайт. ОЗП бувають статичного та динамічного типів. Оскільки обсяг оперативної пам'яті в проєктованому контролері є невеликим, то нема сенсу застосовувати в ньому динамічний ОЗП. А також застосування ОЗП статичного типу дозволяє вирішити завдання збереження даних у пам'яті (на відміну від динамічного ОЗП статичне не вимагає циклів регенерації пам'яті) [13].

Для керування зовнішніми пристроями, а також для забезпечення необхідної швидкості введення отриманого коду в блок пам'яті, необхідно застосувати

					ЦЗДВН 8.171.00.10.327 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		57

контролер KP580BV55 - паралельний інтерфейс, що дозволяє організувати введення/виведення паралельної інформації різного формату.

Ця мікросхема складається з трьох восьмирозрядних портів введення/виведення PA, PB, PC. До того ж порт PC може також працювати в якості двох незалежних 4-х розрядних портів.

Мікросхеми KP580IP82 – це 8-розрядні адресні регістри, які призначені для тимчасового зберігання вхідного та стисненого повідомлення та мають підвищену навантажувальну здатність. Кожна мікросхема складається з восьми однакових функціональних блоків та схеми управління. Блок містить D-регістр-защипку та потужний вихідний вентиль без інверсії або з інверсією. За допомогою схеми управління виконується стробування інформації, що записується, та керування третім станом потужних вихідних вентилів.

1.3.3.2 Мікропроцесорний блок.

Центральний процесорний модуль є основним блоком контролеру. Цей модуль забезпечує управління та синхронізацію роботи всього пристрою, забезпечує прийом, видачу, зберігання та оброблення даних, які надходять по системній шині.

Мікропроцесор KP1821BM85A підключений від одного джерела живлення +5В, виконаний по КМОП-технології. Призначення виводів мікросхеми KP1821BM85A представлено на рисунку 1.11 [14].

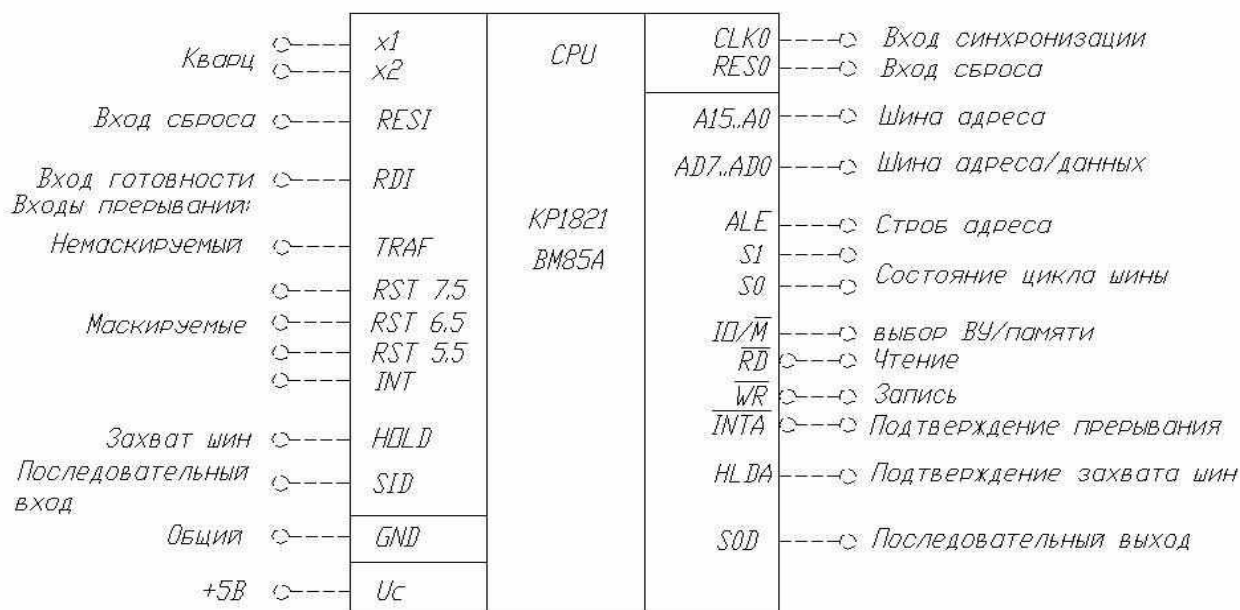


Рисунок 1.11 - Призначення виводів мікросхеми KP1821BM85A

До складу центрального процесорного модуля (ЦПМ) входять процесор КР1821ВМ85А, два буферних регістри КР580ІР82, двохнаправлений шинний формувач КР580ВА86, мультиплексор К555КП11, зовнішній резонатор (ZQ, C₂), схема формування скидання (R₁, C₁, S).

Мікропроцесор КР1821ВМ85А має з'єднані шину даних та шину адреси. Для розподілу сигналів цих шин застосовуються буферні регістри. Поява в першому такті машинного циклу на шині А 15-А8 старшого байту адреси, а на шині АD 7-АD0 - молодшого, стробується сигналом процесора АLE, що використовується для дозволу запису в регістри. При передачі по шині АD 7-АD0 даних цей сигнал відсутній. Таким чином, у регістрах буде записана адреса, а дані будуть передаватися через шинний формувач. До того ж, регістри та шинний формувач виконують функцію збільшення навантажувальної здатності ЦПМ (32 мА/вивід) [13].

Мультиплексор перетворить сигнали процесора в сигнали читання/запису пам'яті та зовнішніх пристроїв - MEMR, MEMW, I/OR, I/OW.

Саме від процесора залежить швидкодія системи в цілому, точність оброблення даних, а також зручність розроблення програмного забезпечення для усього контролера. Достоїнствами мікропроцесора КР1821ВМ85А є:

- низька вартість;
- орієнтування на роботу в складі мікроконтролерів;
- програмна сумісність з мікропроцесором КР580ВМ80А;
- наявність одного джерела живлення +5В;
- наявність вбудованого генератора тактових імпульсів;
- вбудований системний контролер;
- вбудований контролер переривання.

Однак, поруч із переліченими достоїнствами, у КР1821ВМ85А є й деякі недоліки [14]:

- низька швидкодія (тривалість одного машинного такту (0,5 мкс; час виконання однієї команди 7-8 мкс);
- розрядність даних, що оброблюються, 8 біт;
- відсутність команди розподілу у складі команд.

Структура процесора КР1821ВМ85А наведена на рисунку 1.12.

										Лист
										59
Изм.	Лист	№ докум.	Подпись	Дата						

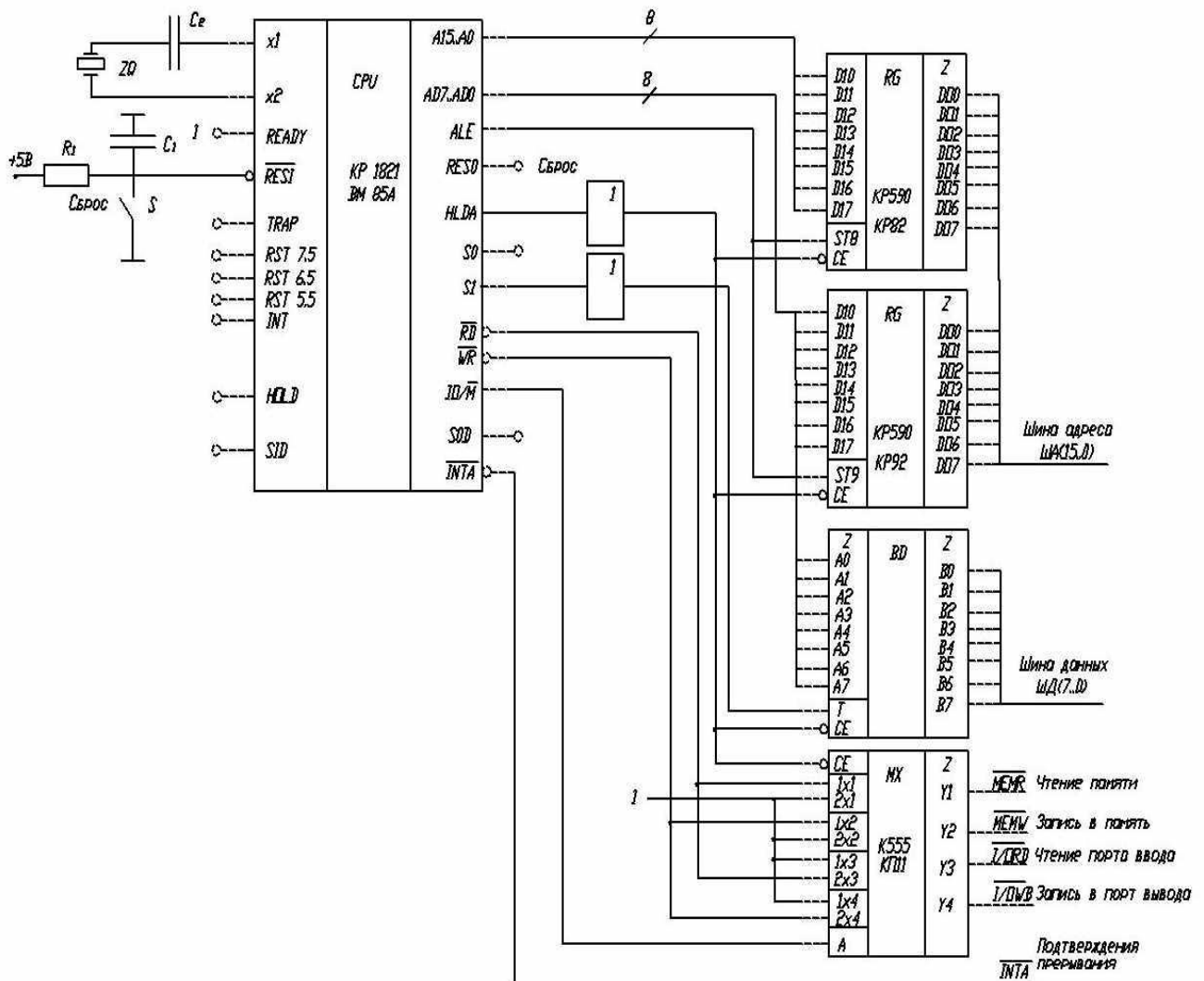


Рисунок 1.12- Структура процесора KP1821BM85A

Для стабілізації частоти системного генератора до виводів X1 і X2 БІС KP1821BM85A підключають кварцовий резонатор з номінальною частотою 18500кГц. При цьому тривалість машинного такту буде дорівнювати 0,486 мкс.

Конденсатор 32 - настроювальний і використовується для регулювання частоти системного генератора в невеликих межах. Ланцюг R1C1 служить для короткочасного формування імпульсу з негативним переднім фронтом тривалістю не менш, ніж 1,5 мкс (3 машинних такта). При прийнятті постійної часу ланцюга R1C1 $\tau = 10$ мкс, $R1 = 10$ кОм, а $C1 = 1000$ пФ.

Принцип роботи схеми скидання полягає в наступному. У нормальному стані C1 заряджений і вхід БІС KP1821BM85A RESI через резистор R1 з'єднаний з

джерелом живлення +5В, яке утримує на ньому «логічну одиницю». При замиканні перемикача S, конденсатор C1 розряджається на корпус. При розмиканні перемикача починається заряджання конденсатора, а вхід RESI БІС КР1821ВМ85А виявляється при цьому замкнутим на корпус, що відповідає стану «логічного нуля». По закінченню заряджання конденсатора, на вході RESI знову встановлюється «логічна одиниця».

1.3.3.3 Блок пам'яті.

При побудові блоку пам'яті вирішують наступні завдання:

- визначення характеристик пам'яті (обчислення її обсягу та розрядності комірок пам'яті);
- розподіл адресного простору між пам'яттю програм та пам'яттю даних;
- розроблення засобів доступу до пам'яті та селекторів адреси.

Блоками пам'яті в проектованому пристрої є ПЗП та ОЗП. У першому пристрої (ПЗП) зберігаються коди програми, а в другому (ОЗП) – дані, що вводяться, виводяться та проміжні дані.

Апаратним шляхом коміркам пам'яті ПЗП та ОЗП можна привласнити будь-які адреси, починаючи від 0 до 65535, але при цьому треба враховувати ту обставину, що при включенні живлення та після скидання мікропроцесор завжди починає зчитувати код команди, яка розташована в комірці з адресою 0000H.

У зв'язку із цією обставиною доцільно для ПЗП відвести комірки пам'яті з адресами 0000h до 07FFh. Нижня адреса ОЗП в цьому випадку 2048 (0800h), верхня границя в 2 Кбайт відповідає адресі 4095 (0FFFh).

У якості ПЗП використовується БІС К573РФ5. Це ПЗП може бути реалізоване на базі однієї мікросхеми пам'яті з ультрафіолетовим стиранням, що має організацію 2к*8 й має час вибірки адреси 450 нс. ІМС має 11 адресних входів, 8 виходів даних, входи дозволу програмування WE, вибирання кристалу CE, дозволу виведення даних з виходів OE [14].

Призначення виводів мікросхеми пам'яті показане на рисунку 1.13. Функціональна схема мікросхеми пам'яті наведена на рисунку 1.14.

У якості проектованого ОЗП вибираємо статичний ОЗП КР537РУ8 обсягом 2 Кбайт. ОЗП бувають статичного й динамічного типів. Оскільки обсяг оперативної пам'яті в проектованому контролері є невеликим, то немає потреби застосовувати в ньому динамічний ОЗП.

					ЦЗДВН 8.171.00.10.327 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		61

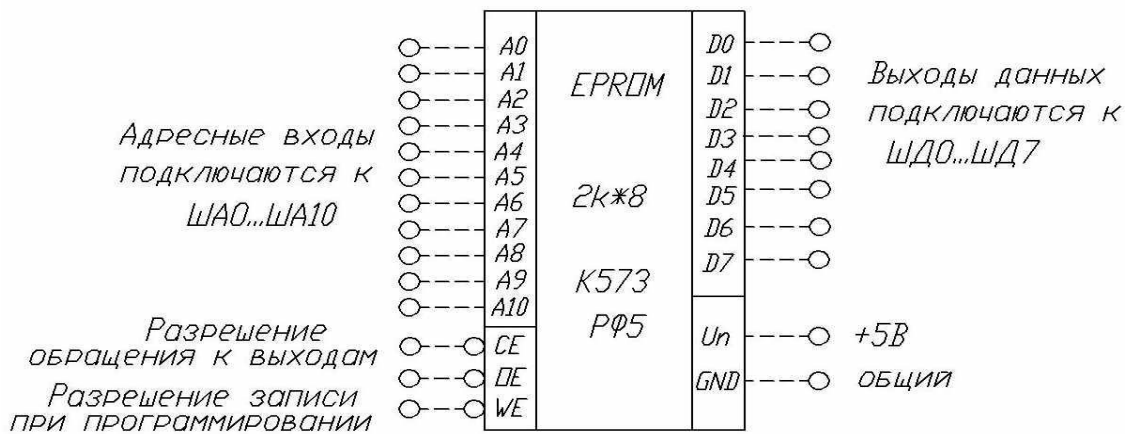


Рисунок 1.13 - Призначення виводів мікросхеми пам'яті з ультрафіолетовим стиранням

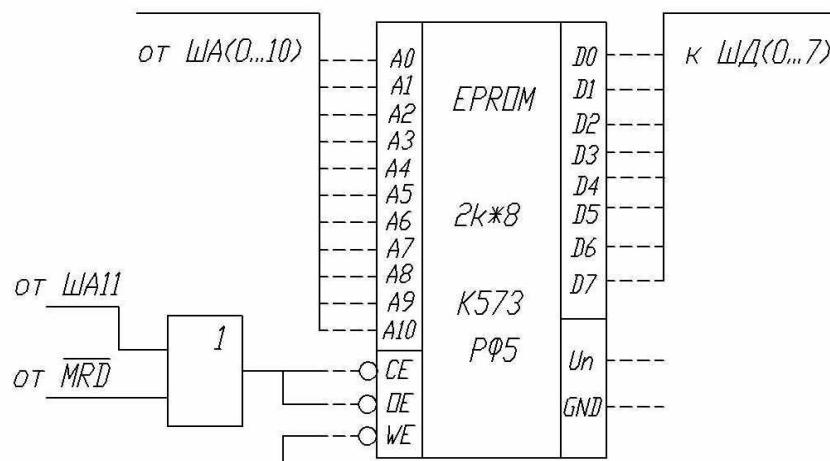


Рисунок 1.14 - Функціональна схема блоку ПЗП

Застосування ОЗП статичного типу дозволяє також вирішити завдання збереження даних у пам'яті (на відміну від динамічного ОЗП статичний пристрій не потребує циклів регенерації пам'яті). Це дозволяє істотно спростити апаратну частину контролера. Для запису даних з входів D 0-D7 у мікросхему необхідно на входах A 0-A9 встановити необхідну адресу комірки пам'яті, подати на входи CS та WR напругу низького рівня. Для читання даних із пам'яті необхідно встановити адресу комірки, на вхід CS подати напругу низького рівня, а на вхід WR - високого рівня [14].

Функціональне позначення та призначення виводів даної мікросхеми наведена на рисунку 1.15.

Функціональна схема блоку ОЗП представлена на рисунку 1.16.

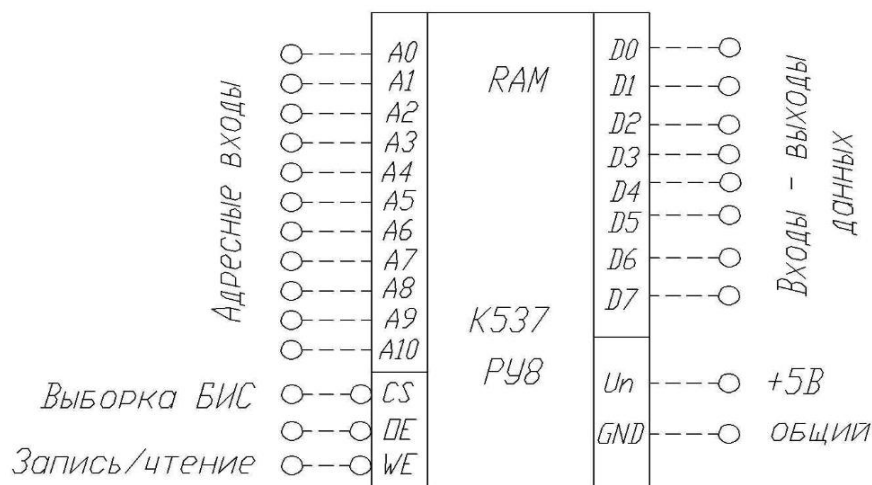


Рисунок 1.15 - Функціональне позначення та призначення виводів БІС КР537РУ8

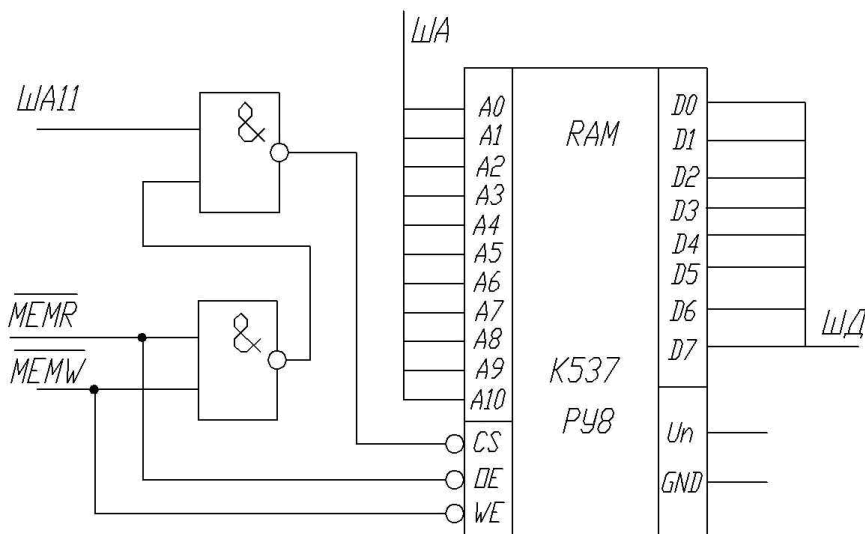


Рисунок 1.16 - Функціональна схема блоку ОЗП

1.3.3.4 Інтерфейсний блок.

Для управління зовнішніми пристроями, а також для забезпечення необхідної швидкості введення отриманого коду у пам'ять необхідно застосувати контролер КР580ВВ55 - паралельний інтерфейс, який дозволяє організувати введення/виведення паралельної інформації різного формату (рис. 1.17) [14].

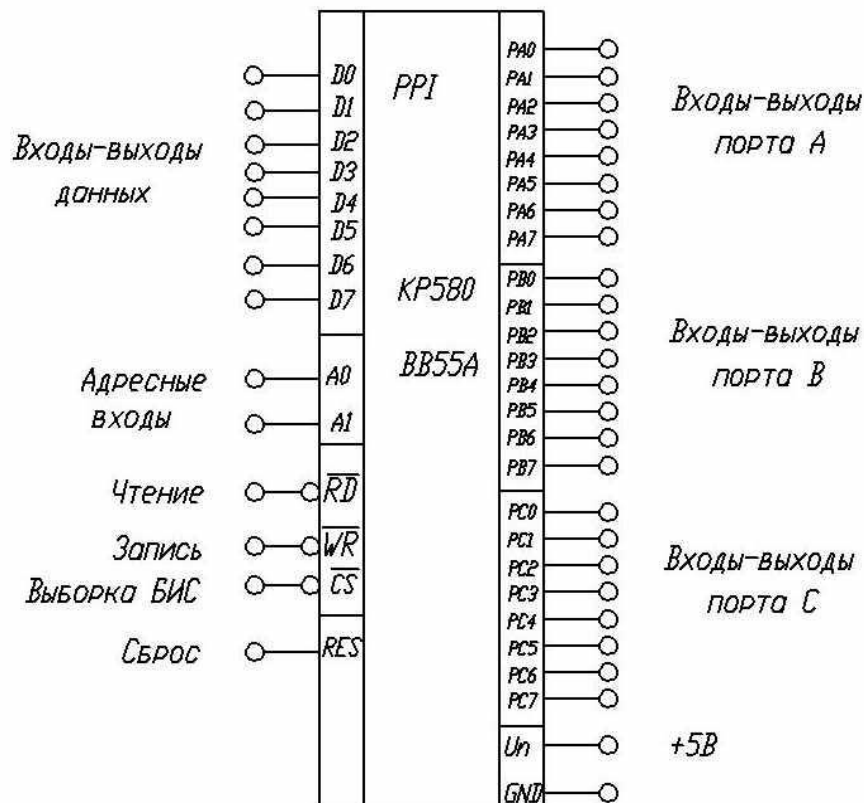


Рисунок 1.17 - Функціональне позначення й призначення виводів БІС КР580ВВ55

Ця мікросхема складається із трьох восьмирозрядних портів введення/виведення РА, РВ, РС. До того ж порт РС може працювати як два незалежних 4-х розрядні порти.

Призначення виводів мікросхеми КР580ВВ55 [14]:

- А1, А0 - адреси. Вони задають адресу поточного порта: 00 - РА, 01 - РВ, 10 - РС, 11 - реєстр керуючого слова. Підключаються до 2-х молодших бітів адресної шини;
- D7..D0 - шина даних;
- РА7..РА0, РВ7..РВ0, РС7..РС0 - відповідні виводи портів РА, РВ, РС;
- RD - читання. Нульовий рівень означає, що процесор читає дані з шини даних, яка в цей момент підключена до порта (РА, РВ, РС) залежно від адреси, обумовленої виводами А1, А0. Підключається до виводу системного контролера IOR;

- WR - запис. Нульовий рівень означає, що процесор видав дані на шину даних, підключену в цей момент до порта (РА, РВ, РС) залежно від адреси, обумовленої виводами А1, А0. Підключається до виводу системного контролера ІОW;
- CS - вибирання мікросхеми. Одиничний рівень переводить входи мікросхеми в Z-стан. При поданні сигналу з дешифратора задається адреса порта.

Для завдання роботи порта використовується керуюче слово, що містить 8 біт.

D 6-5: номер режиму порта РА у двійковій системі;

D4:1 - введення РА, 0 - виведення РА;

D3:1 - введення РС7...РС4, 0 - виведення РС7...РС4;

D2: номер режиму порта РВ у двійковій системі;

D1:1 - введення РВ, 0 - виведення РВ;

D0:1 - введення РС3...РС0, 0 - виведення РС3...РС0;

D7 = 1 - у режимі встановлення.

Порт А програмується на виведення даних (для керування входами аналогового комутатора). Порт В програмується на введення даних з аналого-цифрового перетворювача (АЦП). РС0...РС3 - для керування роботою пристрою відображення інформації. РС4...РС7 - на цих виходах формуються послідовно логічні одиниці, по фронту яких виконується запис семисегментного коду у відповідний регістр.

1.3.3.5 Буферний регістр у режимі зберігання.

Мікросхеми КР580ІР82 (рис. 1.18) - 8-розрядні адресні регістри, які призначені для зв'язку мікропроцесора з системною шиною, мають підвищену навантажувальну здатність. Мікросхема КР580ІР82 - 8-розрядний D-регістр заціпка без інверсії й з трьома станами на виході [14].

Кожна мікросхема складається з восьми однакових функціональних блоків та схеми керування. Блок містить D-регістр заціпку та потужний вихідний вентиль без інверсії або з інверсією. За допомогою схеми керування виконується стробування інформації, що записується, та керування третім станом потужних вихідних вентилів.

Призначення виводів мікросхеми представлено в таблиці 1.24.

					ЦЗДВН 8.171.00.10.327 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		65

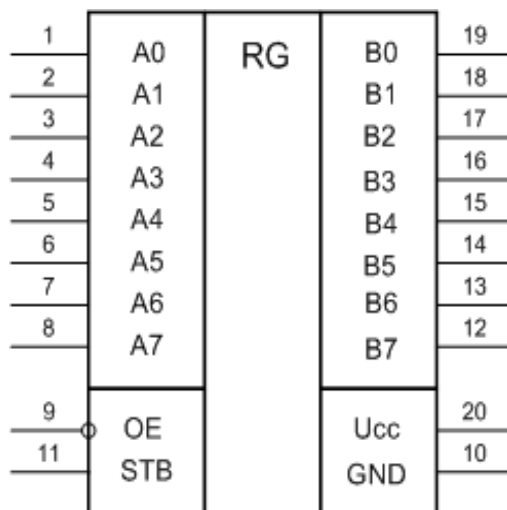


Рисунок 1.18 - Мікросхема КР580ІР82

Залежно від стану сигналу стробування мікросхеми можуть працювати у двох режимах: у режимі шинного формувача та у режимі зберігання.

DD2 тимчасово зберігає вхідні повідомлення в коді ASCII. Код ASCII –це міжнародний стандартний код для обміну інформацією. ASCII являє собою 8-бітне кодування для подання десяткових цифр, латинського та національного алфавітів, знаків пунктуації та керуючих символів. Нижню половину кодової таблиці (від 0 по 127) займають символи US-ASCII, а верхню (від 128 по 255) — різноманітні інші символи.

DD11 тимчасово зберігає повідомлення, що було стиснене за допомогою коду Шеннона - Фано.

Таблиця 1.24 - Призначення виводів мікросхеми КР580ІР82

Вивід	Позначення	Тип виводу	Функціональне призначення виводів
1-8	D 0-D7	Вхід	Інформаційна шина
9	OE	Вхід	Дозвіл передачі
10	GND	-	Загальний
11	STB	Вхід	Сигнал стробування
12-19	Q7-Q0	Вихід	Інформаційна шина
20	Ucc	-	Напруга живлення +5В

1.3.4 Розроблення програмного забезпечення проектованого пристрою

1.3.4.1 Опис реалізації програмного забезпечення для програмованого контролеру послідовного введення-виведення.

Фрагмент програми на асемблері для мікроконтролера КР1821ВМ85, що наведений в цьому розділі, являє собою один із варіантів організації асинхронної передачі символів масиву інформації за допомогою послідовного каналу зв'язку, який використовує програмований контролер послідовного введення-виведення КР1821ВМ85А.

Для введення до контролеру інструкцій та виведення даних використовуються відповідно адреси INSTR і DATA. Ідентифікатор N означає кількість елементів масиву даних. ADDR означає адресу буфера даних в пам'яті. При виникненні помилки парності відбувається звернення до підпрограми оброблення помилок, яка розташована за адресою ERR.

Фрагмент програми:

Мітка – Мнемокод - Коментар

BEGIN DI ;Заборона переривання

MVI A,40H ;Запис інструкції програмного скидання

OUT INSTR

MVI A,7DH ; Запис інструкції режиму

OUT INSTR

MVI A,31H ; Запис інструкції команди передачі

OUT INSTR

MVI B,N ; Встановлення лічильнику масиву даних

LXI H,ADDR ; Завантаження початкової адреси масиву

ENTR: MOV A,M ; Передача елемента масиву до акумулятору

OUT DATA ; Запис елемента масиву

WAIT: IN INSTR ; Слово стану

MOV C,A ; Зберігання слова стану

ANI 08 ; Виділення біту помилки парності

					ЦЗДВН 8.171.00.10.327 ПЗ	Лист
						67
Изм.	Лист	№ докум.	Подпись	Дата		

CNZ ERR ; Якщо помилка, то до програми оброблення
 MOV A,C ; Відновлення слова стану
 RAR ; Контроль готовності передавача
 JNC WAIT ; Якщо передавач не готовий, то повторення
 DCR B ; Зміна лічильнику елементів масиву
 JZ EXIT ; Якщо все, то вихід із програми
 INX H ; Наступний елемент масиву
 JMP ENTR ; Повторення циклу передачі
 EXIT: MVI A,38H ; Запис інструкції команди кінця передачі
 OUT INSTR
 EI ; Дозвіл переривання

1.3.4.2 Опис реалізації алгоритму кодування.

Через те, що алгоритми кодування / декодування за методами Шеннона-Фано та Хаффмана мають схожу структуру, виявилось можливим реалізувати їх роботу в одній програмі. При цьому очевидний вииграш у розмірі машинного коду, тому що обидва методи, як при кодуванні, так й при декодуванні використовуються підпрограми із загального модуля.

Кодування даних програмою відбувається в кілька наступних етапів.

1. Сканування вихідного файлу з метою складання таблиці кількості появ кожного символу у вхідному повідомленні, у якій зберігається інформація про те, скільки разів у файлі зустрічається кожен символ вхідного повідомлення.

2. Відповідно до методу кодування даних (задається ключем командного рядка) формується кодове дерево, листами якого є символи алфавіту повідомлення, а вузлами - допоміжні елементи. Кожен вузол дерева має дві гілки, що виходять з нього, які можуть закінчуватися або вузлом, або листом. При цьому одній з областей (у лівій частині) відповідає символ 0, а іншій області (у правій частині) - символ 1. Таким чином, ліва область відповідає верхній підгрупі для коду Шеннона-Фано або символу (основному або допоміжному) з більшою ймовірністю для коду Хаффмана. Алгоритми побудови кодового дерева для кожного з методів будуть описані далі.

3. Побудова таблиці кодових комбінацій шляхом рекурсивного обходу кодового дерева.

4. У вихідному файлі зберігається заголовок і таблиця ймовірностей появи символів первинного алфавіту.

5. Для кожного символу вихідного файлу згідно таблиці кодових комбінацій визначається кодова комбінація, яка потім побітно заноситься у вихідний файл.

1.3.4.3 Опис реалізації алгоритму декодування.

Декодування даних здійснюється в кілька наступних етапів.

1. Зчитування заголовка з вхідного файлу, визначення типу коду, кількості закодованих символів, розміру таблиці кількості появ кожного символу у вхідному повідомленні. Створення вихідного файлу з відповідним ім'ям.

2. Зчитування із вхідного файлу таблиці кількості появ кожного символу у вхідному повідомленні.

3. Побудова кодового дерева відповідно типу коду.

4. Зчитування біту коду із вхідного файлу. Зчитаний біт приймається за перший біт кодової комбінації. Якщо значення зчитаного біту дорівнює 0, то потрібно переміститись від кореня дерева по лівій області на один вузол. Якщо значення зчитаного біту дорівнює 1, то потрібно переміститись по правій області на один вузол.

5. Якщо при переміщенні вже досягнутий лист дерева, то він містить закодований символ, у цьому випадку записуємо його у вихідний файл і переходимо до п. 7.

6. Зчитуємо наступний біт і переміщаємося по відповідній області дерева (аналогічно п. 4) . Переходимо до п. 5.

7. Якщо не дійшли до кінця файлу, то переходимо до п.4.

8. Відновлення атрибутів, часу та дати останньої зміни вихідного файлу.

1.3.4.4 Побудова кодового дерева.

Спочатку здійснюється побудова списку символів, упорядкованого за спаданням кількості появ кожного символу первинного алфавіту у вхідному повідомленні за допомогою об'єкту List на основі інформації, яка міститься у таблиці кількості появ кожного символу первинного алфавіту у вхідному повідомленні. Створюється вузол кодового дерева, який є коренем дерева, що будується.

					ЦЗДВН 8.171.00.10.327 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		69

Потім виконується безпосередньо побудова кодового дерева шляхом рекурсивних викликів процедури MakeSFTree. Вхідними параметрами цієї процедури є:

- покажчик на початок списку, що оброблюється;
- сумарна кількість появ символів первинного алфавіту списку, що оброблюється;
- покажчик на поточний вузол кодового дерева.

Для побудови кодового дерева в якості параметрів в процедуру MakeSFTree передають покажчик на початок упорядкованого списку, кількість символів у вхідному файлі, покажчик на створений корінь кодового дерева.

Алгоритм роботи процедури MakeSFTree складається з наступних етапів:

1. Якщо у списку, що оброблюється, залишився єдиний елемент, то перетворити поточний вузол кодового дерева в лист і вийти із процедури у блок, що викликає.

2. Якщо кількість появи символу (першого елемента) списку, що оброблюється, більше половини сумарної кількості появ символів списку, то потрібно розділити список, що розглядається, на два списки (групи): перший елемент - у перший список, інші елементи - у другий список. Покажчик на початок першого списку потрібно поставити у відповідність лівому нащадкові поточного вузла дерева, покажчик на початок другого списку – правому. Далі потрібно створити два нових вузли кодового дерева і викликати процедуру MakeSFTree для кожного з двох списків. Потім вийти з процедури в блок, що викликає.

3. Прохід по оброблюваному списку від початку до кінця з одночасним розрахунком сумарної кількості появ пройдених елементів списку доти, поки сумарна кількість появ пройдених елементів у сумі із кількістю появи наступного елемента не перевищить половини сумарної кількості появ елементів списку, що оброблюється. На цьому прохід списку припиняється і список розділяється на два списки (пройдені елементи - у перший список, інші елементи, починаючи з наступного - у другий список). Покажчик на початок першого списку ставиться у відповідність лівому нащадкові поточного вузла дерева, покажчик на початок другого списку – правому нащадкові поточного вузла дерева. Створюються два нових вузли кодового дерева і викликається процедура MakeSFTree для кожного з двох списків. Здійснюється вихід із процедури в блок, що викликає.

1.3.4.5 Побудова таблиці кодових комбінацій.

Таблиця кодових комбінацій будується шляхом рекурсивного обходу кодового дерева. Програмно це реалізовано за допомогою рекурсивного виклику процедури MakeCode . Вхідним параметром функції є покажчик на вузол дерева.

Спочатку встановлюється нульова довжина поточної кодової комбінації, що відповідає порожньому ланцюжку бітів. Потім викликається процедура MakeCode з покажчиком на корінь кодового дерева в якості параметра. Після виконання цієї процедури таблиця кодових комбінацій містить побудовані кодові комбінації для усіх символів алфавіту повідомлення, а кодове дерево містить єдиний вузол, що є коренем (інші вузли вилучені).

Алгоритм роботи процедури MakeCode складається з наступних етапів:

1. Якщо поточний вузол є аркушем, то поточна кодова комбінація заноситься в таблицю кодових комбінацій по індексу, що дорівнює ASCII-коду відповідного символу і здійснюється вихід із процедури (у блок, що викликає).

2. Якщо лівий та правий нащадки не існують, то здійснюється вихід із процедури (у блок, що викликає).

3. У кінець поточної кодової комбінації дописується біт із значенням 0.

4. Якщо лівий нащадок існує, то рекурсивно викликається процедура MakeCode з покажчиком на нього в якості аргументу.

5. Лівий нащадок видаляється.

6. Із поточної кодової комбінації видаляється останній біт, внаслідок чого довжина цієї кодової комбінації зменшується на одиницю.

7. У кінець поточної кодової комбінації дописується біт із значенням 1.

8. Якщо правий нащадок існує, то рекурсивно викликається процедура MakeCode з покажчиком на нього в якості аргументу.

9. Правий нащадок видаляється.

10. Із поточної кодової комбінації видаляється останній біт, внаслідок чого довжина цієї кодової комбінації зменшується на одиницю.

11. Вихід із процедури (у блок, що викликає).

					ЦЗДВН 8.171.00.10.327 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		71

2 ТЕХНІКО - ЕКОНОМІЧНА ЧАСТИНА

2.1 Соціально-екологічні проблеми інформаційного суспільства

На сьогодні людство знаходиться на стадії широкомасштабної інформаційно-технологічної революції, пов'язаної з інтернет-технологіями, які є потужним «набором інструментів, який може бути використаний мудро або нерозумно майже у всіх областях, й як частина майже будь-якої стратегії». Спостерігається перехід на новий етап світового розвитку, заснований на економіці знань. Інформація та знання стають одним зі стратегічних ресурсів. Інвестиції у знання в промислово розвинених країнах ростуть швидше, ніж інвестиції в основні фонди. Ці постулати є різновидом теорії постіндустріального суспільства, основу якої заклали Белл і Тоффлер. Ідею інформаційного суспільства в середині 40-х років ХХ століття пропонував японський соціолог і футуролог Йонедзі Масуда [15].

Поняття «інформаційне суспільство» увів у 1969 році професор Токійського технологічного інституту Ю. Хаяші. Основу теорії інформаційного суспільства заклали Д. Белл, З. Бжезинский, Е. Тоффлер. До теми інформаційного суспільства неодноразово зверталися й інші вчені – А.І. Ракитов, І.Н. Курносов та інші [15].

Передбачається, що головним фактором розвитку інформаційного суспільства є економіка, яка не тільки використовує знання в різноманітній формі, але й створює їх у вигляді високотехнологічної продукції, висококваліфікованих послуг, наукової продукції та устаткування. Але, незважаючи на загальне визнання та практичне застосування даної концепції, виникає безліч проблем, які ставлять під сумнів правильність та успішність подальшого розвитку нашої земної цивілізації [15].

Сучасні проблеми інформаційного суспільства взаємозалежні між собою й є наслідком нероздільного ланцюга процесів, ініційованих самими людьми в різні періоди розвитку людства. Необхідність забезпечення власного виживання та подальшого існування сприяла розвитку світового господарства, економіки та одночасно виникненню екологічних проблем. Ці два напрямки - економіка та екологія - є основними для аналізу проблем та наслідків життєдіяльності людства. Задоволення постійно зростаючого попиту населення на різні блага в умовах ринкової економіки породило конкуренцію за право виробляти та пропонувати ці

					ЦЗДВН 8.171.00.10.327 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		72

блага, отримуючи постійний прибуток, що призводить до особистого збагачення деякої частини населення.

Мало хто відмовляється від одержання прибутків та надприбутків будь-якими способами.

Однак, при цьому відбуваються не тільки позитивні процеси, які стосуються задоволення попиту, але й негативні. Сучасний стан природного середовища на нашій планеті можна охарактеризувати як екологічну кризу. Протягом найближчих 30-40 років вчені припускають подальший розвиток необоротних процесів, які ведуть до знищення життя на Землі взагалі. Аналіз еволюції даного явища показує, що є причинами цього. Людина, як біологічний і соціальний вид або суб'єкт господарювання, сам підриває підстави свого існування тими способами господарювання, які вона розробила та впровадила, прийняла в якості законів та закономірностей економічного розвитку країн та народів. Мають місце зовнішні ефекти. У цей час це призвело до таких явищ у масштабах планети, як парниковий ефект, порушення озонового шару, забруднення світового океану, кислотні дощі, проблема відходів, загальна криза здоров'я населення. Дані факти свідчать про низький рівень сформованості екологічного мислення та моральності, а також про те, що тільки економічними методами неможливо вирішити проблеми збереження навколишнього середовища. Необхідно звернутися до морально - смислової складової свідомості людини, її ідеалам та цінностям [15].

Реформи мають потребу в кадрах, які мають сучасні знання та нове економічне мислення. Придбання знань відбувається в процесі утворення особистості. Тому у побудові конкурентоспроможної економіки важливе місце займають освітні установи, які займаються підготовкою кадрів. Таким чином, спостерігається взаємозв'язок та взаємовплив економічного та освітнього аспектів розвитку інформаційного суспільства.

Одним з найважливіших завдань, які стоять у цей час перед людством, є підвищення якості підготовки фахівців у системі неперервної освіти. Однак, ця проблема складається у відсутності повноцінного наукового визначення поняття якості освіти та тих критеріїв, за якими її можна оцінювати. У той же час, зміни, які відбуваються у світовому співтоваристві, вимагають модернізації освітніх систем [16].

Якість освіти містить у собі якість освітнього середовища, якість педагогічних кадрів, якість керівництва, якість матеріально – технічної бази, якість учнів. У зв'язку цим, виникає ряд питань. Чи реально державному вузу переходити

					ЦЗДВН 8.171.00.10.327 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		73

на самооплатність за рахунок виконання госпдоговірних робіт? Які форми об'єднання навчальних закладів із виробництвом є більш перспективними: кластерного типу або самостійні? Чи є необхідною рейтингова система оцінки викладачів та навчального закладу в цілому? Чи має право викладач знати свій рейтинг? Чи правомірно оцінювати викладача «очами студентів» та приховувати від нього цю інформацію, але використовувати при призначенні доплат та надбавок до зарплати? Чи повинен інженер, що викладає, або інший фахівець одержувати педагогічну освіту? Чи може викладати на курсах підвищення кваліфікації у вищому навчальному закладі фахівець, який не має наукового ступеню, якщо аудиторія слухачів - кандидати та доктори наук? Зростає роль нових технологій навчання, наприклад, дистанційне, мультимедійне. Але не враховується відсутній при цьому індивідуальний вплив педагога. Зникають також механізми запам'ятовування згідно традиційної педагогіки, що пов'язані з моторною, механічною, фізичною, слуховою пам'яттю. Складність оцінки якості освіти в порівнянні з якісною оцінкою будь-якого матеріального об'єкта полягає в неможливості застосувати до людини, як до споживача освітньої інформації, об'єктивні методики моніторингу його духовного розвитку. Це духовне вдосконалення на основі отриманих знань не можна виміряти в метричній або іншій системі мер і співвіднести з певним стандартом. Не піддається оцінці також й ступінь розвитку людського мислення, інтелекту як результату отриманої освіти. Якість освіти у вищій школі є залежним від попередніх ступенів освітньої системи. Крім того, освіта кожної людини є індивідуальною і залежить від його генетичних даних та їх проявлень [16].

У процесі навчання формуються також й моральні якості особистості, які впливають на відношення до роботи й, відповідно, на спрямованість використання отриманих знань.

Однією з проблем сучасної освіти, також взаємозалежної з економікою, є відповідність професійного вибору учнів їхньої майбутньої спеціальності. Цю проблему можна віднести до категорії моральних, тому що саме моральні мотиви можуть впливати на рішення майбутнього студента про вибір професії. Важливість проблеми моральних основ професійного вибору визначається кількома факторами. У період навчання у вищому навчальному закладі відповідність до своєї майбутньої спеціальності сприяє розвитку пізнавальних інтересів студентів, підвищує якість підготовки фахівців і допомагає знайти не тільки професію, але й покликання. Моральні принципи випускників вищого навчального закладу

					ЦЗДВН 8.171.00.10.327 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		74

(фахівців – практиків) впливають на ті сфери діяльності, в яких вони працюють. Викладача, шкільного вчителя, класного керівника - на рівень знань та вихованості учнів, економіста - на законність та ефективність фінансових проектів, лікаря - на відношення до хворих, а у випадку заняття даним фахівцем керівної адміністративної посади – на роботу усього колективу, підбір та розміщення кадрів й, в остаточному підсумку, на якість роботи установи в цілому. Отже, важливо, щоб вибір професії визначався й моральними факторами. Виховання моральних якостей, одночасно з придбанням професійних знань та практичним досвідом, формують рівень готовності особистості до заняття тієї або іншої посади. Необхідно враховувати, що теоретичні професійні знання будь-якого керівника в області маркетингу на практиці переломлюються через його особисту моральну позицію, що, у свою чергу, опосередковано впливає на стратегію керування й, в остаточному підсумку, на показники діяльності установи. Таким чином, проблеми кадрової політики, моральності, якості освіти та економіки є взаємозалежними [16].

Інформатизація впливає як на окремих індивідуумів, так й на суспільство в цілому. Обсяг інформації зростає. Цей процес вимагає розроблення нових, більш складних електронних пристроїв та систем, розвиваються нові сучасні технології. Створюються системи штучного інтелекту, у яких електронні обчислювальні машини добувають та використовують інформацію без допомоги людини. Розвивається автоматизація не тільки ручної роботи, як це спостерігалось в індустріальному суспільстві, а й розумової праці. Дані процеси сприяють вивільненню робочої сили та виникненню безробіття, тобто створюють економічні проблеми на ринку роботи. Відсутність роботи вимагає звернення безробітної людини до ринку освітніх послуг. Виникає проблема підготовки та перепідготовки професіоналів своєї справи в системі освіти.

Спостерігається також проблема, що породжується нерівністю в доступності використання сучасних інформаційних ресурсів. Зростаючі вимоги до освіти можуть привести до величезного розриву між групами населення з вищою освітою та без вищої освіти. Така нерівність існує як між країнами, так й серед різних груп населення усередині однієї країни. У результаті відбувається віддалення цих груп населення, губиться зв'язок між ними. Така ситуація, що викликана економічною нерівністю, сприяє збільшенню економічних проблем і одночасно проблем освітніх, оскільки ускладнює доступ до знань.

					ЦЗДВН 8.171.00.10.327 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		75

Одна з проблем становлення фахівця – це формування його ділової культури, що згодом буде сприяти професійній адаптації. Роль установ освіти в підвищенні ділової культури можна розглядати з двох точок зору. Перша - це підготовка кадрів, які володіють достатнім рівнем ділової культури. Друга - це проблеми підвищення рівня ділової культури саме в установах освіти. Загальна культура особистості є тією основою, на якій формується свідомість фахівця [16].

Освіта особистості є основою для розвитку та удосконалення суспільства в цілому як у матеріальній, так й в духовній сферах. У процесі освіти формуються основи культури, моральності та професіоналізму. Але негативний або позитивний початок багато в чому визначається суб'єктивізмом окремих особистостей, які завдяки своїй активності та волі визначають напрямок розвитку, що надалі позначається на результаті та породжує проблеми, які вимагають рішення. Освіта – одна із складових культури особистості, а культура формується у процесі освіти, відповідно, й їхні проблеми є взаємозалежними. Проблеми сучасної культури можна класифікувати, виходячи з самого визначення цієї сфери людського життя. Культура – це все, що створено людством протягом його багатовікової історії. Таким чином, аналіз виникаючих проблем можна виконати у наступних напрямках. Перший – це людина та формування її особистості як творця та носія культури. Другий - це відношення людей до навколишнього світу, до того, що повинне представляти загальнолюдські цінності. Третій - це проблеми духовного та матеріального світу, які породжені людством, завдяки його неправомірному відношенню до того, що власно ним й було створено [16].

Існує й проблема вибору. Що вибрати споживачеві з величезної кількості товарів, послуг у зв'язку зі змінами в сфері освіти та культури? Це, багато в чому, проблема моральності. В інформаційному суспільстві може з'являтися та поширюватися те, що є непристойним або забороненим й сприяє моральному розкладанню, падінню характерів. Поширюється масова культура. Уніфікується масова свідомість.

Таким чином, відбувається вплив інформаційного суспільства на формування особистості. Особистість може втратити здатність орієнтуватися у зовнішньому світі, який постійно змінюється. Інтернет не тільки поєднує, але й розділяє людей. Можливість віртуального спілкування часом заміняє реальне спілкування, внаслідок чого люди можуть почувати себе усе більш самотньо. Можливий розпад інституту родини, тому що спілкування родичів відбувається з використанням електронних програм. Перенасичення інформацією може

					ЦЗДВН 8.171.00.10.327 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		76

сформувані потреби у використанні різних засобів відходу від дійсності, у вживанні заборонених речовин, а також привести до надмірного захоплення віртуальними мирами, що знижує здатність до взаємодії із навколишнім середовищем [16].

Педагогічне рішення даної проблеми не є однозначним. Один з напрямків виховної роботи – це правильна організація читацької діяльності. Нажаль, доводиться констатувати, що воно рідко застосовується в повсякденній практиці життя. У періодичній пресі та у засобах масової інформації рекомендуються всілякі заняття, за винятком читання книг, як серед дітей, так й серед дорослих. Читацька діяльність сприяє активізації процесу соціалізації особистості, що розвивається. Соціалізація через літературу означає, що, читаючи або слухаючи читання, дитина як би займає позицію усередині зображуваних у літературному творі дійсних обставин, які сприяють збагаченню його непрямого життєвого досвіду. Діти та підлітки ще не приймають участі в багатьох видах діяльності, які формують особистість та її ідеали. Тому читацька діяльність грає в їхньому житті роль набагато більш важливу, ніж у житті дорослих, будучи джерелом непрямого життєвого досвіду. Читання використовувалося в дитячих клубах педагогами С.Т. Шацким і А.У. Зеленко (1905-1907 рр.). А.С. Макаренко одним з видів клубної діяльності називав створення кружків по роботі з книгою. Велике значення роботи з юними книголюбамі надавав В.А. Сухомлинский. В інформаційному суспільстві культура інтерпретації та розуміння письмових текстів знижується. Так виникає проблема навчання.

У той же час відбувається становлення електронного середовища для економічної діяльності, глобальної мережевої економіки. Але виникає одна з серйозних проблем сучасного інформаційного суспільства – проблема захисту інформації.

Аналізуючи здоров'я, можна відзначити, що використання інформаційних пристроїв негативно позначаються на фізичному стані й дітей, й дорослих, призводить до малорухомого способу життя, погіршує зір, створює психологічну залежність, людина випробує вплив випромінювання, вдихає запах пластмаси й т.п. Остаточні наслідки впливу цих факторів на організм у цей час ще не можуть бути докладно вивчені в силу короткочасності їхнього існування в порівнянні з людським життям. Але дані процеси сприяють зниженню працездатності, зниженню продуктивності роботи і знову повертають людство до економічних проблем.

					ЦЗДВН 8.171.00.10.327 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		77

Будучи джерелом дисгармонії в природі, людина, як розумна істота, в стані усунути ті порушення, які створив власноруч. Подальша стратегія розвитку людства складається в коеволюції людини та біосфери, ноосферному розвитку. Людина і природа з'являються в якості відносно самостійних, але взаємодоповнюючих суб'єктів взаємодії. До людини повинне прийти усвідомлення того, що людське суспільство залежить від стану живої природи. Біосфера не має потреби в існуванні людини, у той же час людина без біосфери існувати не зможе. Коеволюційна стратегія виживання вимагає обмеження перетворення планети в процесі господарської діяльності, духовного вдосконалення самої людини, переоцінку цінностей, моральних орієнтирів та ідеалів у напрямку їхньої екологічної гуманізації. Необхідно формувати глобальне мислення во ім'я збереження нашого світу.

2.2 Розрахунок собівартості та оптової ціни пристрою, що розроблюється

2.2.1 Розрахунок повної собівартості проектного пристрою

Розрахуємо повну собівартість проектного пристрою – стиснення інформації на основі коду Шеннона - Фано.

Собівартість виробу, що випускається - це виражені в грошовій формі поточні витрати підприємства на його виробництво та збут. Витрати на виробництво утворюють виробничу собівартість, а витрати на виробництво та збут - повну собівартість. Розрахунок собівартості продукту, що випускається, по статтях витрат називається калькуляцією. Калькуляція собівартості пристрою відбувається згідно «Типовому положенню із планування, обліку й калькулювання собівартості продукції (робіт, послуг) у промисловості» [17].

В процесі виробництва будь-якого виробу споживаються різні матеріали, комплектуючі вироби, використовуються різні види обладнання та інструменти, проводиться велика кількість технологічних операцій [18]. У зв'язку з цим для обліку фактичних витрат на виробництво та для обґрунтування собівартості необхідна певна класифікація цих витрат. Для розрахунку собівартості одиниці певного виду продукції, що випускається, застосовується класифікація за калькуляційними статтями витрат.

					ЦЗДВН 8.171.00.10.327 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		78

У плануванні та в обліку собівартості продукції застосовується наступне типове групування за статтями калькуляції [18]:

- основна заробітна плата;
- додаткова заробітна плата;
- відрахування від заробітної плати;
- матеріали та комплектуючі;
- витрати на утримання та експлуатацію обладнання;
- виробничі витрати;
- адміністративні витрати;
- позавиробничі витрати (комерційні витрати) [18].

Групування витрат по калькуляційних статтях витрат дозволяє визначити рівень собівартості виробу, а відповідно й рівень його ціни. Вона характеризує місце виникнення витрат та їх цільове призначення.

Вихідними даними для складання калькуляції собівартості на проєктований пристрій є стаття калькуляції на покупні та комплектуючі вироби. Необхідно врахувати вартість напівфабрикатів, що йдуть на виготовлення друкованої плати.

Дані по цій статті витрат приведені в таблиці 2.1.

Таблиця 2.1 - Дані на покупні та комплектуючі вироби

Найменування комплектуючих	Кількість	Ціна однієї одиниці, (грн.)	Сума на один виріб, (грн.)
1	2	3	4
Мікросхеми			
KP1821BM85	1	110	110
KP580IP82	3	25	75
K1533ЛH2	1	12	12
KP580IP82	1	25	25
KP580BA86	1	10	10
K555КП11	1	15	15
K1533ЛA3	1	15	15
K1533ЛI1	1	10	10
KP580BB55A	1	25	25

Продовження таблиці 2.1

1	2	3	4
К573РФ5	1	95	95
К537РУ8	1	20	20
Конденсатори			
К10-7-50В-22 пФ	16	5	80
К73-9-100В-0.01 мкФ	9	10	90
Резистори			
МЛТ-0.125-8.2 Ом	2	1	2
МЛТ 0.125-10 кОм	4	1	4
Резонатор			
КР 169-12 МГц	1	10	10
Витратні матеріали			
Гетінакс(двосторонній)	100 см ²	40	40
Припой, флюс і др.		15	15
Загальна вартість, (грн.)			653

Витрати на основну заробітну плату (Z_0):

$$Z_0 = T * \Gamma * K * A, \quad (2.1)$$

де T – сумарна трудомісткість розробки продукту (годин), яка визначається експертним шляхом виходячи з фактично витраченого часу на виробництво та налаштування продукту, $T = 8$ (годин);

Γ – середня годинна тарифна ставка одного робітника задіяного у виробництві продукту, грн. / год, $\Gamma = 60$ грн. / год;

K – коефіцієнт трудової участі (розрядності), $K = 1,3$;

A – кількість працівників задіяних у виробництві, $A = 2$.

Тоді

$$Z_0 = T * \Gamma * K * A = 8 * 60 * 1,3 * 2 = 1248 \text{ (грн.)}$$

Додаткова заробітна плата (10 – 30% від Z_o):

$$Z_d = Z_o * K_d / 100, \quad (2.2)$$

де K_d – відсоток додаткової заробітної плати, $K_d = 10\%$.

$$Z_d = Z_o * K_d / 100 = 1248 * (10\% / 100) = 124,80 \text{ (грн.)}$$

Нарахування на заробітну плату – єдиний соціальний внесок у розмірі 22%.

$$H_B = (Z_o + Z_d) * 22 / 100. \quad (2.3)$$

$$H_B = (1248 + 124,8) * 22 / 100 = 302,02 \text{ (грн.)}$$

Витрати на утримання та експлуатацію обладнання.

Оренда машинного часу (O_M):

$$O_M = M_B * \chi_M, \quad (2.4)$$

де M_B – величина машинного часу, необхідного для розробки та налагодження продукту, годин, $M_B = 1 \text{ дн} * 8 \text{ ч} = 8 \text{ годин}$;

χ_M – вартість оренди машинного часу, грн. / год:

$$\chi_M = C_{\text{сoм}} / N_p * 259 * 8, \quad (2.5)$$

де $C_{\text{сoм}}$ – ціна обладнання, задіяного при виробництві виробу,

$$C_{\text{сoм}} = 35000 \text{ грн};$$

N_p – термін ефективної роботи, $N_p = 5$;

259 – кількість робочих днів;

8 – тривалість зміни.

$$\chi_M = 35000 / 5 * 259 * 8 = 3,40 \text{ (грн./ч)}$$

					ЦЗДВН 8.171.00.10.327 ПЗ	Лист
						81
Изм.	Лист	№ докум.	Подпись	Дата		

Тоді

$$O_M = M_B * Ч_M = 8 * 3,4 = 27,20 \text{ (грн.)}. \quad (2.6)$$

Загальновиробничі витрати. Являють собою витрати, що пов'язані з управлінням підрозділом, витрати на службові відрядження співробітників підрозділу, амортизаційні відрахування від вартості основних фондів загальноцехового призначення й т.п.

Загальновиробничі витрати ($B_{ЗВ}$) визначаються в розмірі 130-250% від основної заробітної плати.

$$B_{ЗВ} = Z_o * \% B_{ЗВ} = 1248 * 1,3 = 1622,40 \text{ (грн.)}. \quad (2.7)$$

Визначимо виробничу собівартість:

$$\begin{aligned} B_c &= Z_o + Z_d + H_B + M + O_M + B_{ЗВ} = \\ &= 1248 + 124,80 + 302,02 + 653 + 27,20 + 1622,40 = 3977,42 \text{ (грн.)}. \end{aligned} \quad (2.8)$$

Адміністративні витрати можуть містити:

- витрати, пов'язані з управлінням підприємством;
- витрати на службові відрядження адміністрації підприємства;
- витрати на пожежну і сторожову охорону;
- витрати, пов'язані з підготовкою (навчанням) і перепідготовкою кадрів;
- витрати на перевезення працівників до місця роботи і назад;
- витрати на оплату відсотків за фінансові кредити, а також відсотків за товарні і комерційні кредити;
- витрати, пов'язані з оплатою відсотків за користування матеріальними цінностями, взятими в оренду (лізинг);
- витрати, пов'язані з оплатою послуг комерційних банків та інших кредитно-фінансових підприємств;
- податки, відрахування.

Адміністративні витрати (B_a) визначаються в розмірі 140 - 200% від основної заробітної плати.

					ЦЗДВН 8.171.00.10.327 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		82

$$B_a = 3_0 * \% B_a = 1248 * 1,4 = 1747,20 \text{ (грн.)}. \quad (2.9)$$

Витрати на збут (B_3). Включають витрати на рекламу і предреалізаційну підготовку пристрою. Орієнтовно ці витрати визначаються у розмірі 5 - 10% від виробничої собівартості.

Тоді

$$B_3 = B_c * (5 - 10)\% = 3977,42 * 0,05 = 198,87 \text{ (грн.)}. \quad (2.10)$$

Повна собівартість пристрою (C):

$$C = B_c + B_a + B_3. \quad (2.11)$$

$$C = 3977,42 + 1747,20 + 198,87 = 5923,49 \text{ (грн.)}.$$

Калькуляція собівартості виробу зведена у таблицю 2.2.

Таблиця 2.2 - Калькуляція собівартості виробу, що розробляється

Найменування статей калькуляції	Величина, грн.
1. Основна заробітна плата	1248,00
2. Додаткова заробітна плата	124,80
3. Нарахування на заробітну плату	302,02
4. Матеріали та комплектуючі	653,00
5. Оренда машинного часу	27,20
6. Загальновиробничі витрати	1622,40
7. Адміністративні витрати	1747,20
8. Витрати на збут	198,87
Разом повна собівартість	5923,49

2.2.2 Розрахунок ціни пристрою

Розрахунок оптової ціни виробу проведемо за схемою «собівартість плюс прибуток»:

$$\text{Ц}_{\text{опт}} = \text{C} + \text{П}, \quad (2.12)$$

де C – собівартість пристрою;

П – величина прибутку.

Прибуток визначається виходячи з нормативу рентабельності виробництва продукції:

$$\text{R} = (\text{П} / \text{C}) * 100\%, \quad (2.13)$$

де R - рентабельність продукції (продукту), приймається в розмірі до 35%.

$$\text{R} = 10\%.$$

Тоді оптова ціна:

$$\text{Ц}_{\text{опт}} = \text{C} + (\text{R} * \text{C} / 100) = 5923,49 + 0,1 * 5923,49 = 6515,84 \text{ (грн.)}. \quad (2.14)$$

Визначимо роздрібну ціну розробленого виробу:

$$\text{Ц}_{\text{розн}} = \text{Ц}_{\text{опт}} * 1,2 = 6515,84 * 1,2 = 7819 \text{ (грн.)}, \quad (2.15)$$

де 20% ПДВ.

Позитивні сторони даної методики полягають в її простоті, комплексної очевидності такої функції ціни як відшкодування витрат на виробництво та забезпечення прибутковості від створення та реалізації пристрою. Недолік даної методики полягає в тому, що вона недостатньо враховує ринкові чинники ціноутворення й, перш за все, попит. Однак, у реальній перехідній економіці існують ситуації, коли підприємствам доцільно її застосовувати: в умовах відсутності конкуренції (монополії), при обмеженні рентабельності продукції з

					ЦЗДВН 8.171.00.10.327 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		84

боку держави, виконанні одноразових замовлень, при виготовленні оригінальної продукції.

Необхідно відзначити, що для встановлення реальної ціни, яка б відповідала умовам існуючого ринку, необхідні відповідні маркетингові дослідження.

					ЦЗДВН 8.171.00.10.327 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		85

ВИСНОВКИ

У роботі була розроблена система передачі даних на основі коду Шеннона - Фано. Пристрій побудований на основі мікропроцесорної системи. Розроблені структурна, функціональна, принципова електричні схеми та схема алгоритму роботи пристрою.

У техніко-економічній частині було розглянуте питання про соціально-екологічні проблеми інформаційного суспільства, а також розрахована виробнича собівартість, повна собівартість, оптова ціна, роздрібна ціна розробленого пристрою.

З передачею та стисненням інформації зв'язані дії будь-якого автоматичного пристрою, життя людини, творча діяльність людини, розвиток науки та техніки, економічні та соціальні перетворення в суспільстві й у житті. Для більш ефективного використання інформації необхідно обмінюватися інформацією, що неможливо без її передачі по каналах зв'язку та її стиснення. Стиснення інформації скорочує обсяг простору, що є необхідним для зберігання файлів в електронних обчислювальних машинах, та кількість часу, необхідного для передачі інформації по каналу зв'язку. Іншими цілями кодування інформації є пошук та виправлення помилок, а також шифрування даних. Процес пошуку та виправлення помилок є протилежним стисненню - він збільшує надмірність даних. Видаляючи з тексту надмірність, стиснення сприяє шифруванню даних, що ускладнює пошук шифру доступним для зломщика статистичним методом.

До появи робіт Шеннона та Фано кодування символів алфавіту при передачі повідомлення по каналах зв'язку здійснювалося однаковою кількістю біт, визначеною по формулі Хартлі. З появою цих робіт почали з'являтися способи, які кодують символи різною кількістю бітів у залежності від ймовірностей появи символів у вхідному повідомленні, тобто більш ймовірні символи кодуються короткими кодовими комбінаціями, а символи, що рідко зустрічаються, - більш довгими кодовими комбінаціями.

Методика Шеннона-Фано не гарантує однозначності побудови коду, оскільки при розділенні таблиці ймовірностей на підгрупи можливі різні варіанти розділення.

Із порівняння результатів стиснення даних за методами Шеннона-Фано та Хаффмана можна зробити висновок, що метод Хаффмана в усіх випадках дає коефіцієнт стиснення інформації не істотно кращий, ніж метод Шеннона-Фано.

					ЦЗДВН 8.171.00.10.327 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		86

Із результатів розрахунків видно, що характеристики ефективності коду Шеннона - Фано та коду Хаффмана не істотно відрізняються одні від одних, але код Шеннона - Фано має більш простий алгоритм побудови коду в порівнянні з кодом Хаффмана.

Перевагами даних методів є їхня очевидна простота реалізації й, як наслідок цього, висока швидкість кодування та декодування вхідних повідомлень.

					ЦЗДВН 8.171.00.10.327 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		87

СПИСОК ЛІТЕРАТУРИ

1. Yeung, Raymond W. (2008). "The Science of Information". Information Theory and Network Coding. Springer.
2. Варфоломеева О.Г., Отрох С.І., Твердохліб М.Г., Чумак О.І. Оптимальне приймання сигналів. Основи теорії завадостійкості. Навчальний посібник. – Київ: ННІТІ ДУТ, 2018. – 120 с.
3. Information theory and coding. – 2019. – <https://uclouvain.be/en-cours-2019-LINGI2348>.
4. A.G. Loshkovskii. Teletraffic Theory. Methodical instructions for laboratory works on course «Teletraffic Theory in Telecommunications» / A.G. Loshkovskii, T.R. Shmeleva. – Odessa: ONAT after A.S. Popov, 2013. – 36 p.
5. Signals Transmission In Network Communication. – June 2018. – <http://ecestudyaid.blogspot.com/2018/07/signals-transmission-in-network.html>.
6. Беркман Л.Н., Отрох С.І., Тарбаєв С.І., Чумак Н.С. Загальні поняття про сигнали та канали зв'язку. Навчальний посібник. Частина 1. – Київ: ДУТ ННІТІ, 2017. – 132 с.
7. A Mathematical Theory of Communication. – May 2017. – <http://www.math.harvard.edu/~ctm/home/text/others/shannon/entropy/entropy.pdf>.
8. Беркман Л.Н., Варфоломеева О.Г., Коршун Н.В., Макаренко А.О. Сигнали в системах телекомунікацій та методи їх обробки. Навчальний посібник. Частина 2. – Київ: ДУТ ННІТІ, 2017. – 92с.
9. Information Theory And Coding. – March 2016. – <https://educheer.com/essays/information-theory-and-coding>.
10. Беркман Л.Н., Бондарчук А.П., Гайдур Г.І., Чумак Н.С. Кодування джерел інформації та каналів зв'язку. Навчальний посібник. – Київ: ННІТІ ДУТ, 2018. – 91с.
11. Coding theory. – July 2018. - https://wiki2.org/en/Coding_theory.
12. Куліш С.М. Компоненти елементної бази радіоелектроніки: навчальний посібник до лабораторного практикуму / С.М. Куліш, Ю.А. Волошин. – Х.: Нац. аерокосм. ун-т ім. М.Є. Жуковського «Харк. авіац. ін-т», 2019. – 46 с.

					ЦЗДВН 8.171.00.10.327 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		88

13. Жуйков В.Я. Мікропроцесори і мікроконтролери: електронний підручник з грифом МОН / Жуйков В.Я., Терещенко Т.О., Петергеря Ю.С., Хохлов Ю.В. – <http://www.kaf-pe.ntu-kpi.kiev.ua>, 2010.
14. Нефедов А.В. Интегральные микросхемы и их зарубежные аналоги: Справочник. Т. 10. – М.: ИП Радио Софт, 2000. – 544 с.
15. Ключкова Е.Н. Экономика предприятия: учебник / Е.Н. Ключкова, В.И. Кузнецов, Т.Е. Платонова; под редакцией Е.Н. Ключковой. – М.: Издательство Юрайт, 2019. – 447 с.
16. Арбузов В.В., Скордина Е.М., Бережная О.В. Успешный выпускник как интегральный критерий качества высшего образования // Цели и достижения науки: материалы VI (LVIII) Международной научно-практической конференции по философским, филологическим, юридическим, педагогическим, экономическим, психологическим, социологическим и политическим наукам (Украина, г. Киев, 26 июня 2015 г.). – Горловка: ФЛП Пантюх Ю.Ф., 2015. – С. 29-32.
17. Тарасюк М. Бюджетне планування в Україні // Вісник КНТЕУ. – 2018.– № 2. – С. 19-31.
18. Типовое положение по планированию, учету и калькулированию себестоимости продукции. Утверждено КМ Украины от 26 апреля 1996 №473//Бізнес.-№32-35.

					ЦЗДВН 8.171.00.10.327 ПЗ	Лист
						89
Изм.	Лист	№ докум.	Подпись	Дата		