

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Сумський державний університет
Кафедра електроніки і комп'ютерної техніки

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проекту на тему:

«Кодуючий пристрій на базі адаптивних рівноважних кодів»

Завідувач кафедри

Керівник проекту

Проектував студент

А.С. Опанасюк

О.В. Бережна

В.П. Індучний

Суми
2020 р.

Сумський Державний Університет

Факультет Заочний Кафедра Електроніки і КТ
Спеціальність Електронні системи

ЗАТВЕРДЖУЮ:

Зав. кафедри Опанасюк А.С.

« » 2020 р.

Завдання

на дипломний проект студенту

Індучному Вадиму Павловичу

(прізвище, ім'я, по батькові)

1. **Тема проекту** «Кодуючий пристрій на базі адаптивних рівноважних кодів»

затверджена наказом по університету від «13» квітня 2020 р. № 0532-III

2. **Термін здачі студентом закінченого проекту** 01.06.20

3. **Вихідні дані до проекту** Вхідний код – двійковий, кількість розрядів
вхідного коду $n=7$ (КОІ -7); вихідний код –рівноважний; параметри
рівноважного коду n та k змінюються в межах $k < n \leq 15; 1 \leq k < n$.

4. **Зміст розрахунково-пояснювальної записки (перелік питань, які підлягають розробці)** Вступ. 1. Огляд літератури та постановка завдання. 2. Розробка алгоритму функціонування та структурної схеми проєктованого пристрою. 3. Розробка схеми електричної функціональної. 4. Розробка та розрахунок принципових електричних схем вузлів та блоків пристрою. Висновки.

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

1 Схеми алгоритму. _____

2 Схеми електрична структурна. _____

3 Схеми електрична функціональна. _____

4 Схеми електрична принципова. _____

Календарний план

№ п/п	Найменування етапів дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1.	Огляд технічної літератури	20.04.20	
2.	Розробка алгоритму функціонування та структурної схеми	25.04.20	
3.	Розробка схеми електричної функціональної	30.04.20	
4.	Розрахунок вузлів та блоків пристрою та розробка схеми електричної принципової	10.05.20	
5.	Оформлення графічної частини	20.05.20	
6.	Оформлення пояснювальної записки	25.05.20	
7.	Рецензування та підготовка до захисту	30.05.20	

Студент-дипломник _____

Керівник проекту _____

РЕФЕРАТ

Пояснювальна записка містить: 44 сторінки тексту, 14 рисунків, 3 таблиці, 11 джерел літератури.

Графічна частина містить 4 креслення: блок-схему алгоритму роботи пристрою, структурну, функціональну та принципову схеми.

Пояснювальна записка містить п'ять розділів, в яких розглядається наступне.

Розділ 1 – Огляд літератури. Розглядаються принципи завадостійкого кодування та способи побудови адаптивних систем передачі інформації.

Розділ 2 – Постановка завдання.

Розділ 3 – Розробка алгоритму функціонування та структурної схеми проєктованого пристрою. Для одержання завадостійких рівноважних кодових комбінацій виконується проміжне перетворення вхідних повідомлень у біноміальні числа.

Розділ 4 – Синтез функціональної схеми пристрою кодування.

Розділ 5 – Розробка принципової схеми кодуючого пристрою та програмного забезпечення.

ЗМІСТ

	С.
Вступ	4
1 Огляд літератури	7
1.1 Основи завадостійкого кодування. Принципи побудови завадостійких кодів	7
1.2 Класифікація завадостійких кодів	10
1.3 Основні параметри коригувальних кодів	11
1.4 Граничні співвідношення між параметрами коригувальних кодів	14
1.5 Декодування завадостійких кодів	15
1.6 Способи побудови адаптивних систем кодування	17
2 Постановка завдання	20
3 Розробка алгоритму функціонування та структурної схеми проектного пристрою	21
3.1 Синтез алгоритму функціонування кодуємого пристрою	21
3.2 Синтез структурної схеми проектного пристрою	22
4 Синтез функціональної схеми пристрою кодування	26
5 Розробка принципової схеми кодуємого пристрою	31
5.1 Вибір елементної бази	31
5.2 Вибір мікропроцесорного модуля	33
5.3 Блок пам'яті	35
5.4 Дешифратор адреси ПЗП та ОЗП	35
5.5 Дешифратор зовнішніх пристроїв	37
5.6 Інтерфейсний блок	39
5.7 Розробка програмного забезпечення проектного пристрою	40
Висновки	43
Список літератури	44

					ЦЗДВН 6.05080202.661 ПЗ			
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>				
<i>Разраб.</i>	Індучний В.П.				Кодуючий пристрій на базі адаптивних рівноважних кодів Пояснювальна записка	<i>Лит.</i>	<i>Лист</i>	<i>Листов</i>
<i>Провер.</i>	Бережна О.В.						3	44
<i>Реценз.</i>						СумДУ, гр. ЕСдн – 51п		
<i>Н. Контр.</i>	Бережна О.В.							
<i>Утверд.</i>	Опанасюк А.С							

ВСТУП

З розвитком електронної техніки з'явилася потреба в досягненні великих швидкостей передачі інформації одночасно з досягненням високого рівня вірогідності. Однак, це не завжди вдається. Проблема підвищення вірогідності обумовлена невідповідністю вимог, які пред'являються до характеристик каналів зв'язку, що використовуються при передачі даних, та якістю реальних каналів зв'язку. Наприклад, у мережах передачі даних автоматизованих систем керування потрібно забезпечити ймовірність помилкового приймання вихідних повідомлень не вище ніж 10^{-6} - 10^{-9} . При використанні в реальних каналах зв'язку (ймовірність перекручення біту в таких каналах знаходиться у межах 10^{-2} - 10^{-3}) простого (ненадлишкового) коду ймовірність помилкового приймання вихідних повідомлень приймає значення близько $6 \cdot 10^{-3}$ [1].

Вирішення завдання підвищення вірогідності переданих повідомлень здійснюється у двох напрямках: вдосконалення каналоутворюючої апаратури та використання спеціальних процедур, заснованих на використанні завадостійких (коригувальних) кодів [1].

Завадостійке кодування, по суті, являє собою метод оброблення сигналів, призначений для збільшення надійності передачі по цифрових каналах зв'язку. Хоча різні схеми кодування дуже несхожі один на одного та засновані на різних математичних теоріях, усім ним властиві дві загальні властивості. Одна з них – використання надмірності. Закодовані цифрові повідомлення завжди містять додаткові (або надлишкові) символи. Ці символи використовуються для того, щоб підкреслити індивідуальність кожного повідомлення. Їх завжди обирають таким чином, щоб зробити малою ймовірною втрату повідомленням його індивідуальності через перекручення при впливі перешкод досить великої кількості символів. Друга властивість полягає в усередненні шуму. Ефект усереднення досягається за рахунок того, що надлишкові символи залежать від декількох інформаційних символів [2].

Термінологія теорії кодування виникає з теорії зв'язку, так як розвиток кодів, що коректують помилки, спочатку стимулювався завданнями систем зв'язку. Коди мають багато призначень. Коригувальні коди використовуються для захисту даних у пам'яті обчислювальних пристроїв, на цифрових стрічках та дисках, а також для захисту від неправильного функціонування або шумів у

цифрових логічних ланцюгах. Коди використовуються також для стиснення даних, а теорія кодування тісно пов'язана з теорією планування статистичних експериментів [2, 3].

Двійкові повідомлення звичайно передаються між обчислювальними терміналами, між літальними апаратами, між супутниками. Завадостійкі коди можуть бути використані для одержання надійного зв'язку навіть тоді, коли потужність прийнятого сигналу близька до потужності теплових шумів. Оскільки електромагнітний спектр усе більше й більше заповнюється створюваними людиною сигналами, коди, що коректують помилки, стають ще більш важливим інструментом, тому що дозволяють лініям зв'язку надійно працювати при наявності завад. У військовій справі такі коди часто використовуються для захисту інформації від навмисно організованих супротивником перешкод [3].

У багатьох системах зв'язку є обмеження на потужність, що передається. Наприклад, у системах ретрансляції через супутники збільшення потужності обходиться дуже дорого. Коди, що коректують помилки, є чудовим засобом зниження необхідної потужності, тому що з їхньою допомогою можна правильно відновити отримані повідомлення, які були послаблені в процесі передачі. Передача інформації в обчислювальних системах звичайно чутлива навіть до дуже малої частки помилок, тому що одиночна помилка може порушити програму обчислення. Завадостійке кодування стає в цих додатках досить важливим. Для деяких носіїв обчислювальної пам'яті використання кодів, що контролюють помилки, дозволяє досягти більш щільного впакування бітів [2, 4].

Іншим типом систем зв'язку є система з багатьма користувачами та розподілом за часом, у якій кожному з користувачів заздалегідь запропоновані деякі часові вікна (інтервали), протягом яких йому дозволяється здійснювати передачу даних. Довгі двійкові повідомлення розділяються на пакети, й кожен пакет передається у відведене для нього часове вікно. Через порушення синхронізації або дисципліни обслуговування деякі пакети можуть бути загублені. Коди, що контролюють помилки, захищають дані від таких втрат, тому що загублені пакети можна відновити по відомих пакетах [3].

У сучасних складних цифрових системах можуть виникнути великі потоки даних між підсистемами. Цифрові автопілоти, цифрові системи

керування процесами, цифрові перемикальні системи та цифрові системи оброблення радарних сигналів – це системи, що містять великі масиви цифрових даних, які повинні бути розподілені між багатьма взаємно пов'язаними підсистемами. Ці дані повинні бути передані або по спеціально призначеним для цього лініям зв'язку, або за допомогою більш складних систем з шинами передачі даних та з розподілом за часом. У кожному разі важливу роль грають методи завадостійкого кодування, тому що вони дозволяють гарантувати відповідні характеристики систем передачі інформації [2].

1 ОГЛЯД ЛІТЕРАТУРИ

1.1 Основи завадостійкого кодування. Принципи побудови завадостійких кодів

Завадостійкими кодами називаються коди, які дозволяють виявити або виправити в дискретних повідомленнях помилки, що виникають при передачі повідомлень по каналах зв'язку з перешкодами [3].

Застосування завадостійких (коригувальних) кодів, як правило, пов'язане з розбиванням повідомлень на блоки з k елементів, що називаються k -елементними комбінаціями. У загальному випадку кожний елемент може приймати одне з q різних значень, тому такий код називається q -ічним. Параметр q називається основою коду.

У цей час найбільше поширення при передачі даних одержали коди з основою $q = 2$. Тому надалі розглядаються переважно двійкові коди, а у необхідних випадках результати узагальнюються на випадок довільних значень q [3, 5].

Нехай z_i – довільна n -елементна двійкова кодова комбінація. Тоді загальна кількість різних кодових комбінацій довжини n дорівнює 2^n . Якщо з множини $Z = \{z_i\}$ обирати за деяким правилом $N_k < 2^n$ комбінацій, то отримана множина $V = \{v_i\}$, $i = (1, \dots, N_k)$, $V \subset Z$ буде коригувальним кодом. Комбінації v_i називаються дозволеними кодовими комбінаціями, а комбінації $z_i \notin V$ – забороненими кодовими комбінаціями.

Застосування коригувальних кодів при підвищенні вірогідності пов'язане з рішенням завдань кодування або декодування повідомлень при виявленні та (або) виправленні помилок.

Завдання кодування – це завдання одержання при передачі повідомлень для кожного k -елементного блоку відповідної йому комбінації з множини V . При цьому однозначна відповідність між блоками та кодовими комбінаціями є можливою якщо кількість блоків не перевищує N_k [5].

Завдання декодування – це завдання одержання k -елементного блоку з прийнятих комбінацій Z при одночасному виявленні або виправленні помилок.

При декодуванні коригувальний код може використовуватися в трьох режимах:

- виявлення помилок,
- виправлення помилок;
- одночасне виправлення та виявлення помилок.

Виявлення помилок ілюструється рисунком 1.1 [6].

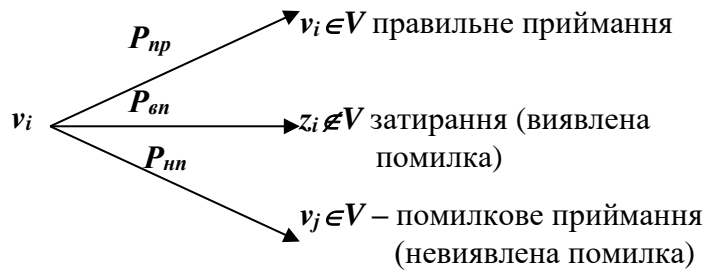


Рисунок 1.1 – Схема виявлення помилок

При передачі по каналу зв'язку деякої дозволеної комбінації на приймальній стороні можливі три неспільних результати [6]:

- правильне приймання, що відповідає відсутності помилок у повідомленні, переданому по дискретному каналу зв'язку (імовірність події P_{pr});
- приймання з виявленою помилкою, коли кодова комбінація під впливом помилок трансформується в заборонену (імовірність події P_{en}), при цьому формується сигнал затирання;
- приймання з невиявленою помилкою (імовірність події P_{nn}), що відповідає переходу під впливом помилок однієї дозволеної комбінації в іншу дозволена.

Так як перелічені три результати становлять повну групу подій, то є справедливим співвідношення [3]

$$P_{pr} + P_{en} + P_{nn} = 1.$$

Виправлення помилок ілюструється рисунком 1.2 [6].

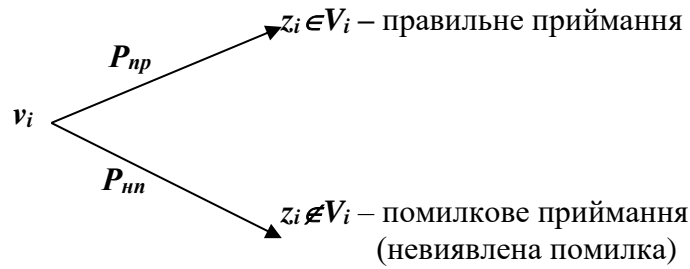


Рисунок 1.2 – Схема виправлення помилок

При виправленні помилок множина Z розбивається на підмножини V_i , $i=(1, \dots, N_k)$ такі, що $\bigcup_i V_i = Z$, $\forall i, j \rightarrow V_i \cap V_j = \emptyset$. У кожній підмножині V_i є одна дозволена комбінація v_i та деяка кількість заборонених комбінацій. Якщо прийнято комбінацію $z_i \in V_j$, то вважається, що передавалася комбінація v_j . Підмножина V_j називається захисною зоною комбінації v_i [3, 6].

При передачі по каналу зв'язку деякої дозволеної комбінації v_i можливі наступні результати [3]:

- правильне приймання, коли при передачі по каналу зв'язку деякої дозволеної комбінації z_i вона попадає в захисну зону V_i (імовірність події $P_{пр}$);
- помилкове приймання, коли перекручена під впливом помилок комбінація не попадає в захисну зону V_i й ідентифікується як інша дозволена комбінація (імовірність події $P_{нп}$).

Очевидно, що

$$P_{пр} + P_{нп} = 1.$$

При одночасному виправленні та виявленні помилок також формуються підмножини V_i , у кожній з яких є одна дозволена комбінація й деяка кількість заборонених, причому $\bigcup_i V_i \supset V$. Якщо прийнята комбінація $z_i \in V_i$, то вважається, що передавалася комбінація v_i . Якщо $z_i \in Z \setminus \bigcup_i V_i$, то помилка виявляється, а комбінація затирається. Виправлення та виявлення помилок ілюструється рисунком 1.3 [6].

Практична реалізація принципів завадостійкого кодування повинна забезпечити побудову множини дозволених комбінацій V таким чином, щоб максимізувати $P_{пр}$ та мінімізувати складність кодерів [7].

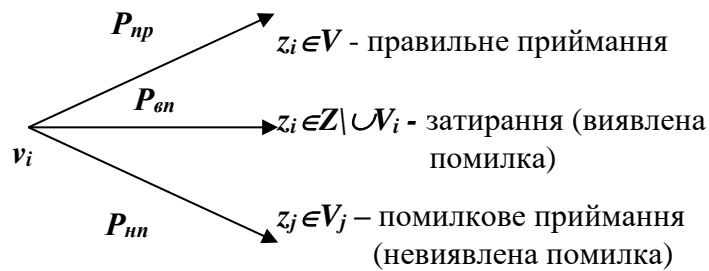


Рисунок 1.3 - Схема виявлення та виправлення помилок

1.2 Класифікація завадостійких кодів

Класифікація завадостійких кодів наведена на рисунку 1.4 [8].

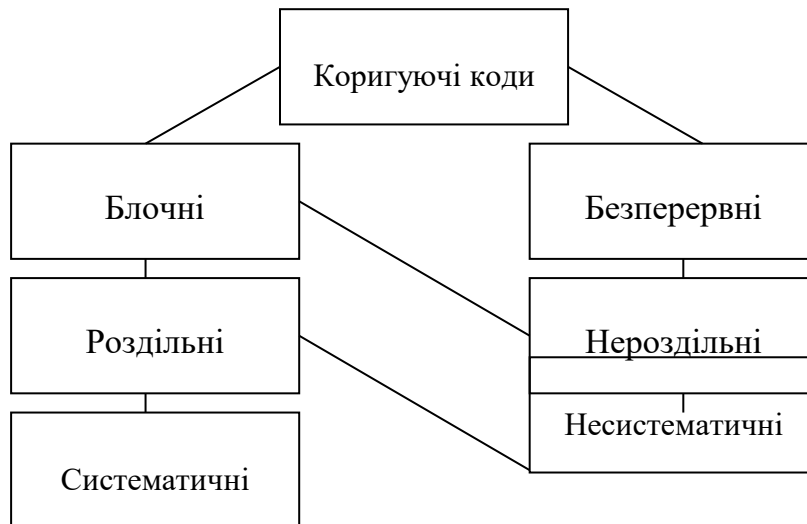


Рисунок 1.4 – Класифікація завадостійких кодів

Блокові коди – це коди, у яких кожному k – елементному блоку, отриманому з повідомлення, відповідає кодова комбінація.

Безперервні коди – це коди, при використанні яких кодування та декодування являють собою операції, які здійснюються над безперервними послідовностями елементів. Розподіл на блоки при цьому є неможливим.

Роздільні коди – це блокові коди, які мають складну структуру, при якій n - елементна кодова комбінація складається з k інформаційних розрядів та $(n - k)$ надлишкових (перевірочних) розрядів. Для таких кодів існує правило визначення перевірочних елементів (розрядів) шляхом аналізу інформаційних розрядів [5].

Нероздільні коди не поділяються на інформаційні та перевірочні частини.

Систематичні коди – це роздільні блокові коди, у яких перевірочні та інформаційні розряди пов'язані між собою системами лінійних рівнянь [5].

1.3 Основні параметри коригувальних кодів

Кодова комбінація $A=(a_0 a_1 \dots a_{n-1})$, де a_i – елементи, значення яких дорівнюють 0 або 1, та характеризуються вагою $W = \sum_i^{n-1} a_i$, тобто кількістю одиниць [8].

Нехай $A=(a_0 a_1 \dots a_{n-1})$, $B=(b_1 b_2 \dots b_{n-1})$ – n - елементні комбінації. Відстань Хеммінга між A і B – d_{AB} – вага суми цих комбінацій по $\text{mod}2$, тобто [5]

$$d_{AB} = W(A \oplus B) = \sum_{i=0}^{n-1} (a_i \oplus b_i) \quad (1.1)$$

Для оцінки властивостей коригувального коду використовують наступні основні параметри [7]:

- довжина кодової комбінації n ;
- кількість дозволених комбінацій N_k . Для роздільних кодів $N_k=2^k$, тому кількість дозволених комбінацій повністю визначається кількістю інформаційних елементів k , яка також є параметром коду;
- абсолютна надмірність $r=\log_2 2^n \cdot N_k^{-1}=n-\log_2 N_k$. Для роздільних кодів $r = n - k$;
- відносна надмірність (або швидкість передачі коду) $R = k / n$. R характеризує ефективність використання каналу зв'язку;

- мінімальна кодова відстань d_{\min} – найменша з попарних відстаней Хеммінга в множині дозволених комбінацій коригувального коду;
- d_{\min} однозначно визначає максимально гарантовану кратність помилок, що виявляються S та виправляються t .

Для режиму виявлення помилок (рисунок 1.5) $S=d_{\min}-1$ або $d_{\min}=S+1$, тому що зміна внаслідок помилок елементів комбінації в кількості d_{\min} та більше може привести до переходу однієї дозвolenої кодової комбінації в іншу. На рисунку 1.5 показано, що при $d_{\min}=5$ вплив помилок кратності 4 та менше приводить до приймання заборонених комбінацій, а п'ятикратна помилка може привести до приймання дозвolenої комбінації В при передачі кодової комбінації А [6].

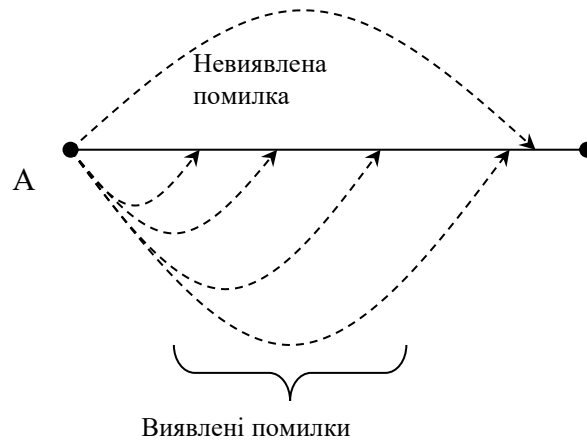


Рисунок 1.5 - Режим виявлення помилок

Для режиму виправлення помилок $t=(d_{\min}-1)/2$ або $d_{\min}=2t+1$ при непарних d_{\min} , $t=(d_{\min}-2)/2$ або $d_{\min}=2t+2$ при парних d_{\min} .

На рисунку 1.6 ці співвідношення пояснюються для $d_{\min}=5$ [6].

Однократні та двократні помилки не ведуть до переходу прийнятих комбінацій у зону V_B й тому виправляються. Помилки кратності більшої ніж два визначають виникнення комбінацій з помилками, що не виправляються.

При одночасному виправленні та виявленні помилок d_{\min} , t та S зв'язані співвідношенням $d_{\min}=t+S+1$, $S \geq t$, що є узагальнюючим для розглянутих вище випадків. Цей режим використання коду ілюструється рисунком 1.7 [6].

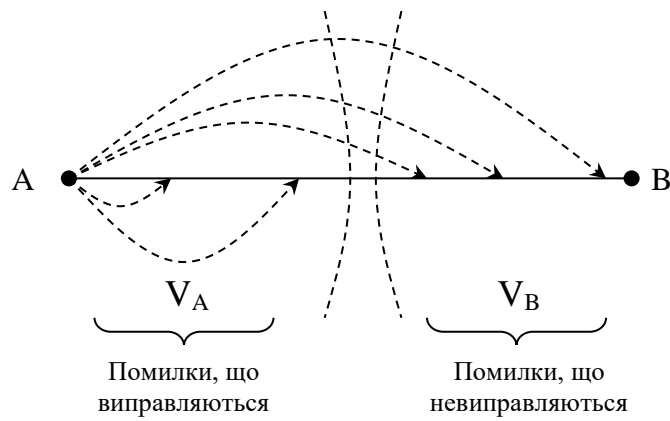


Рисунок 1.6 - Режим виправлення помилок

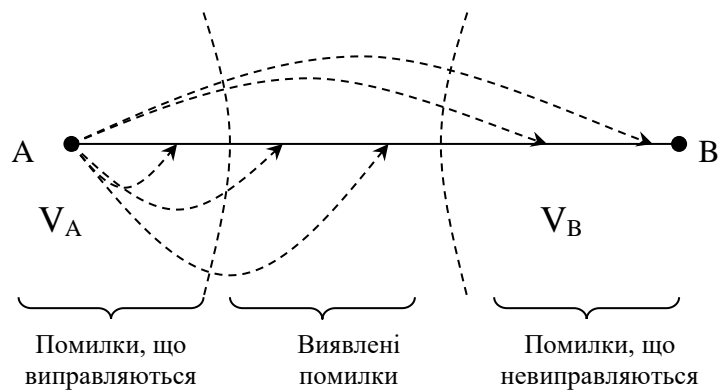


Рисунок 1.7 - Режим виявлення й виправлення помилок

Приклад. При $d_{\min}=5$ можливі наступні режими використання коду [5]:

- виявлення помилок кратності 4 та менше ($t=0, S=4$);
- виправлення однократних та двократних помилок ($t=S=2$);
- одночасне виправлення однократних ($t=1$) та виявлення помилок кратності ≤ 3 ($S=3$).

1.4 Граничні співвідношення між параметрами коригувальних кодів

Однією з найважливіших завдань побудови коригувального коду з заданими характеристиками є встановлення співвідношення між його здатністю виявляти або виправляти помилки та надмірністю коду [4].

Розглянемо граничні оцінки, що зв'язують d_{\min} , n та k . Нехай код призначений для виправлення помилок кратності t . Тоді в кожній з 2^k захисних зон має дорівнювати $\sum_{i=0}^t C_n^i$, а кількість різних комбінацій має дорівнювати 2^n , тобто $2^n \geq 2^k \sum_{i=0}^t C_n^i$ або $n - k \geq \log_2 \sum_{i=0}^t C_n^i$ [3].

Таке співвідношення називається границею Хеммінга. Для q -ічного коду границя Хеммінга має вигляд [4]

$$\sum_{i=0}^t C_n^i (q-1)^i \leq q^{n(1-R)}. \quad (1.2)$$

Можна показати, що границя Хеммінга є досить грубою в області малих R . Тому при малих R доцільно використовувати границю Плоткіна, яка визначається формулою [5]

$$d_{\min} \leq \frac{(q-1)nq^k}{q(q^k-1)}, \quad (1.3)$$

а для двійкового коду

$$2^k(n-2d_{\min}) \leq 2d_{\min}. \quad (1.4)$$

Границі Хеммінга та Плоткіна задають мінімальну надмірність, при якій існує коригувальний код, який має мінімальну кодову відстань d_{\min} та такий, що гарантовано виправляє t -кратні помилки [5, 7].

Нехай у сферу радіусом $2t$, проведену навколо будь-якої дозволеної комбінації, не попадає ніяка інша дозволена комбінація [7]. Тоді цей код здатний виправити усі помилки кратності не менш ніж t . Кількість дозволених кодових комбінацій визначається співвідношенням $2^n \geq 2^k \sum_{i=0}^{2t} C_n^i$ або

$$n - k \geq \log_2 \sum_{i=0}^{2t} C_n^i \quad [8].$$

Таке співвідношення зветься границею Варшавова-Гілберта. Для q -ічного коду границя Варшавова-Гілберта має вигляд [5]

$$n - k \geq \log_q \sum_{i=0}^{2t} C_n^i (q-1)^i. \quad (1.5)$$

Границя Варшамова-Гілберта показує при якому значенні $(n - k)$ обов'язково існує код, гарантовано виправляючий помилки кратності t [5].

1.5 Декодування завадостійких кодів

Розглянемо основні принципи декодування блокових та безперервних кодів, які використовуються у режимі виправлення або одночасного виправлення та виявлення помилок [4].

Найпоширенішим способом виправлення помилок блоковим кодом є декодування по максимуму правдоподібності.

Спосіб декодування по методу максимальної правдоподібності заснований на наступному очевидному положенні. Ймовірність перекручення кодової комбінації зменшується зі зростанням кратності помилок, тобто $P(i, n) > P(i+1, n)$. Тому представляється природним ототожнювати прийняту комбінацію з кодовою комбінацією, що відстоїть від неї на найменшу відстань [5].

Процедура виправлення помилок реалізується в наступній послідовності [4]:

- обчислюється відстань Хеммінга d_i між прийнятою комбінацією z та всіма дозволеними комбінаціями $\{v_i\}$, $i=(1, \dots, 2^k)$;
- отримана комбінація z ототожнюється з кодовою комбінацією, для якої є справедливим $d = \min_i d_i$.

Слід зазначити, що рішення може бути не єдиним, тому необхідно передбачити додаткову процедуру вибору єдиної кодової комбінації з декількох можливих.

При використанні безперервних кодів виправлення помилок здійснюється такими способами [8]:

- спосіб граничного декодування;
- спосіб послідовного декодування;
- спосіб декодування по максимуму правдоподібності.

Через те, що довжина комбінацій безперервного коду є досить великою (а в загальному випадку напівнескінченою) прийняття рішень про наявність та виправлення помилок здійснюється на основі аналізу відрізків (сегментів) комбінацій кінцевої довжини з врахуванням взаємозалежності цих сегментів .

Граничне декодування полягає в тому що при прийманні комбінацій безперервного коду формується перевірочний вектор спеціального виду та аналізується його структура. У результаті визначаються та інвертуються розряди кодової комбінації, перекручення яких приводить до ідентичної або схожої структури вектору перевірок. Аналіз комбінацій та ідентифікація перекручених розрядів здійснюється за допомогою граничних схем, що працюють по мажоритарному принципу [6-8].

Спосіб граничного декодування заснований на простих ідеях й знаходить досить широке застосування на практиці. Алгоритми декодування, як правило, розробляються для конкретних типів кодів, принципи побудови яких допускають ефективну реалізацію цих алгоритмів.

Послідовне декодування засноване на інтерпретації процесу формування кодових комбінацій як процедури побудови деякого дерева, при цьому кожному шляху відповідає дозволена комбінація. При декодуванні прийнятої комбінації по кодовому дереву послідовно визначається шлях, а, отже, деяка кодова комбінація v_i , така що умовна ймовірність $P(v_i/z)$ є максимальною, тобто $P(v_i/z):\max_i$. Особливістю практичної реалізації способу є необхідність наявності в декодері запам'ятовуючого пристрою досить великої ємності для зберігання прийнятих послідовностей та результатів пошуку найбільш імовірного шляху. Причому, чим нижче якість дискретного каналу зв'язку, тим більший обсяг пам'яті буде задіяним [7].

Виправлення помилок по максимуму правдоподібності виконується за аналогією з декодуванням блокових кодів. Відмінність полягає в тому, що рішення приймаються на послідовності взаємозалежних відрізків прийнятої комбінації [5].

У цей час найбільш ефективним алгоритмом декодування вважається алгоритм Вітербі, що здійснює ітеративну обробку послідовності сегментів та який базується на методах динамічного програмування.

1.6 Способи побудови адаптивних систем кодування

Існує кілька основних способів побудови адаптивних систем кодування інформації. Розглянемо коротко суть кожного способу [3, 4].

Перший спосіб включає зчитування байту інформації, перетворення в послідовний код, перетворення відповідно до правил модуляції та передачу в канал зв'язку. Він відрізняється тим, що на передавальній стороні додатково формують тестову послідовність нулів та одиниць, передають каналом зв'язку тестову комбінацію в послідовному коді, одержують з приймальної сторони кількість перекручених нулів та одиниць, перетворюють значення з послідовного коду в паралельний, розраховують ентропії передачі нулів та одиниць, вибирають оптимальний алгоритм завадостійкого кодування з урахуванням проведених розрахунків, здійснюють завадостійке кодування інформації відповідно до обраного алгоритму. Далі переупаковують інформацію в повнобайтний формат, передають у послідовному коді службову інформацію про тип обраного алгоритму та кількість помилок, які повинні виправлятися, перетворюють інформацію в байтах з паралельного коду в послідовний та передають байти в канал зв'язку. На приймальній стороні одержують з каналу зв'язку тестову послідовність нулів та одиниць у послідовному коді, підраховують кількість перекручених нулів та одиниць, передають на іншу сторону результати проведеного тестування в послідовному коді, одержують з каналу зв'язку умови завадостійкого кодування в послідовному коді та перетворюють їх у паралельний код, одержують з каналу зв'язку байти інформації в послідовному коді та перетворюють їх у паралельний код. Потім переупаковують інформаційні байти відповідно до отриманих від передавача умов передачі, здійснюють завадостійке декодування сформованого масиву значень, записують отриману інформацію на носій [3].

Другий спосіб містить наступні дії. На передавальній стороні – зчитування з носія інформації, яка повинна передаватися, передача в послідовному коді, на приймальній стороні – зчитування інформації з паралельного інтерфейсу та запис отриманої інформації на носій персонального комп'ютера. Він відрізняється тим, що на передавальній стороні додатково передають у канал зв'язку тестову послідовність "нулів", передають у канал зв'язку тестову послідовність "одиниць" та приймають від приймальної частини

результати тестування каналу зв'язку. На приймальній стороні одержують з каналу зв'язку тестову послідовність "нулів", одержують з каналу зв'язку тестову послідовність "одиниць", визначають коефіцієнт несиметричності каналу зв'язку, розраховують граничний рівень ідентифікації сигналу. Далі передають на передавальну сторону результати, визначають у відповідності зі швидкістю передачі часові інтервали ідентифікації інформаційних сигналів та фіксують їх у програмувальному таймері, вимірюють протягом фіксованого інтервалу часу рівень напруги в каналі зв'язку та порівнюють певну напругу з граничним рівнем та ідентифікують рівень сигналу [3].

Третій спосіб відрізняється тим, що на передавальній стороні додатково визначають швидкість передачі інформації, визначають період вимірів напруги перешкод у каналі зв'язку, визначають середнє значення напруги перешкод у каналі, розраховують перевищення рівня перешкод над базовим m , розраховують значення $(\log_2 m + 1)$, приводять отримане значення до непарного числа округленням у більшу сторону. Потім передають каналом зв'язку на приймальну сторону умови передачі (кількість повторів) та передають байт інформації каналом зв'язку певну кількість разів. На приймальній стороні з каналу зв'язку отримують умови передачі (кількість повторів однієї й тієї інформації), одержують з каналу зв'язку байт інформації певну кількість разів, перетворюють його в паралельний код та записують значення в оперативний запам'ятовуючий пристрій, здійснюють побітне визначення вмісту розрядів байту за максимальною кількістю їх значень (нулів та одиниць) [3].

Четвертий спосіб полягає у наступному. На передавальній стороні додатково реєструють масив значень напруги перешкод у каналі зв'язку та розраховують його середнє діюче значення, визначають максимальну швидкість передачі за умови збереження працездатності пристрою та необхідних рівнів логічних "нулів" та "одиниць" передачі сигналів. Далі вибирають стандартну швидкість передачі інформації, передають умови зв'язку до приймальної частини, перетворюють дані з паралельного формату в послідовний з певною швидкістю, перетворюють сигнал відповідно до певного закону модуляції та логічних рівнів сигналів. На приймальній стороні одержують з каналу зв'язку умови передачі інформації, ініціалізують пристрій на певну швидкість обміну інформацією та коефіцієнт підсилення, одержують з каналу зв'язку інформаційні сигнали, послабляють їх відповідно до отриманого

коефіцієнта та демодулюють. Потім перетворюють інформацію з послідовного формату в паралельний, зчитують інформацію в паралельному форматі та записують отриману інформацію на носій персонального комп'ютера [3, 5].

2 ПОСТАНОВКА ЗАВДАННЯ

Метою роботи є розробка пристрою адаптивного рівноважного кодування, що дозволить вибирати параметри коду для досягнення кращої завадостійкості та кращої швидкості передачі. Необхідно синтезувати алгоритм роботи, а також структурну, функціональну та принципову схеми пристрою кодування.

Розроблювальний пристрій повинен виконувати наступні функції:

- перетворення двійкового коду в біноміальний із заданими параметрами коду n та k ;
- перетворення біноміального коду в рівноважний із заданими параметрами n та k ;
- можливість зміни параметрів кодування для досягнення кращої швидкості та завадостійкості.

3 РОЗРОБКА АЛГОРИТМУ ФУНКЦІОНУВАННЯ ТА СТРУКТУРНОЇ СХЕМИ ПРОЕКТОВАНОГО ПРИСТРОЮ

3.1 Синтез алгоритму функціонування кодуючого пристрою

Рівноважні коди є рівномірними та характеризуються довжиною кодових комбінацій n та постійним числом одиниць у кодових комбінаціях k . Якщо отримана на виході з каналу зв'язку кодова комбінація є помилковою, але з невиявленими помилками, то така кодова комбінація є рівноважною. Для помилкового переходу, який не виявляється, однієї рівноважної комбінації до іншої необхідно, щоб перехід k одиниць в нулі супроводжувався переходом такої ж кількості g нулів в одиниці. У протилежному випадку помилка буде виявлена простим підрахунком одиниць у кодовій комбінації [5].

Завдання цієї роботи полягає в тому, щоб перетворити вихідну двійкову послідовність у рівноважну [9].

Це завдання вирішується у два етапи:

- перетворення двійкового коду в біноміальний;
- перетворення біноміального коду в рівноважний.

У свою чергу перший етап можна розбити на наступні кроки.

1. Обчислення коефіцієнта a_{j-1} . Для цього обчислюється ваговий коефіцієнт $(j-1)$ -го розряду $C_{n-1}^{k-q_i}$, що порівнюється з двійковим числом X . Якщо $X \geq C_{n-1}^{k-q_i}$, то $a_{j-1}=1, q_{j-1} = q_j + 1 = 1$. У протилежному випадку $a_{j-1}=0, q_{j-1} = q_j + 0 = 0$.

2. Визначається залишок R , який дорівнює $X - C_{n-1}^{k-q_i}$, якщо a_{j-1} дорівнює 1 або $R = X$, якщо a_{j-1} дорівнює 0.

3. Обчислення коефіцієнта a_{j-2} . Для цього залишок R порівнюється з вагою $(j-2)$ -го розряду $C_{n-2}^{k-q_{i-1}}$. Якщо $R \geq C_{n-2}^{k-q_i}$, то $a_{j-2}=1, q_{j-2} = q_{j-1} + 1 = 1$. У протилежному випадку $a_{j-2}=0, q_{j-2} = q_{j-1} + 0 = 0$.

4. За аналогією з пунктами 2,3 визначаються коефіцієнти a_{j-3}, a_{j-4}, \dots , доти поки кількість одиниць в отриманій послідовності не буде дорівнює k або

кількість нулів не буде дорівнювати $n - k$. Виконання однієї з цих умов буде свідчити про закінчення процесу перетворення.

Перетворення біноміального коду в рівноважний здійснюється шляхом додавання нулів або одиниць до біноміальної послідовності. Якщо кількість одиниць у біноміальній послідовності є меншою ніж заданий параметр k , то додаємо необхідну кількість одиниць до заданого значення k . Якщо кількість нулів в біноміальній послідовності є меншою ніж $n - k$, то додаємо необхідну кількість нулів.

Синтезована блок-схема алгоритму показана на рисунку 3.1.

3.2 Синтез структурної схеми проектованого пристрою

Виходячи з алгоритму, описаного вище, проектований пристрій буде містити наступні блоки: блок лічильників, вхідний регістр, блок підсумовування, блок керування, блок обчислення вагового коефіцієнта, блок порівняння, вихідний регістр. Синтезована структурна схема показана на рисунку 3.2.

Принцип роботи пристрою кодування наступний. По сигналу “Дозвіл” з виходу блоку керування вихідне повідомлення X записується у вхідний регістр та передається на входи блоку підсумовування та блоку порівняння. Блок лічильників по сигналу від блоку керування визначає значення $n - i$ та $k - q_i$, які надходять у блок обчислення вагового коефіцієнта.

У блоці порівняння відбувається порівняння обчисленого вагового коефіцієнта a_{j-i} i -го розряду із двійковим числом X , а результат передається блоку керування, де аналізується отримане значення. Якщо $X \geq a_{j-i}$, то у вихідний регістр записується значення розряду кодової комбінації, що дорівнює 1, а в блоці підсумовування визначається залишок $X - a_{j-i}$, що надходить у блоки 1 та 6. Якщо $X \leq a_{j-i}$, то у вихідний регістр записується нуль, а визначення залишку не відбувається.

Одночасно блок 6 підраховує кількість нулів та одиниць, яка передається у вихідний регістр.

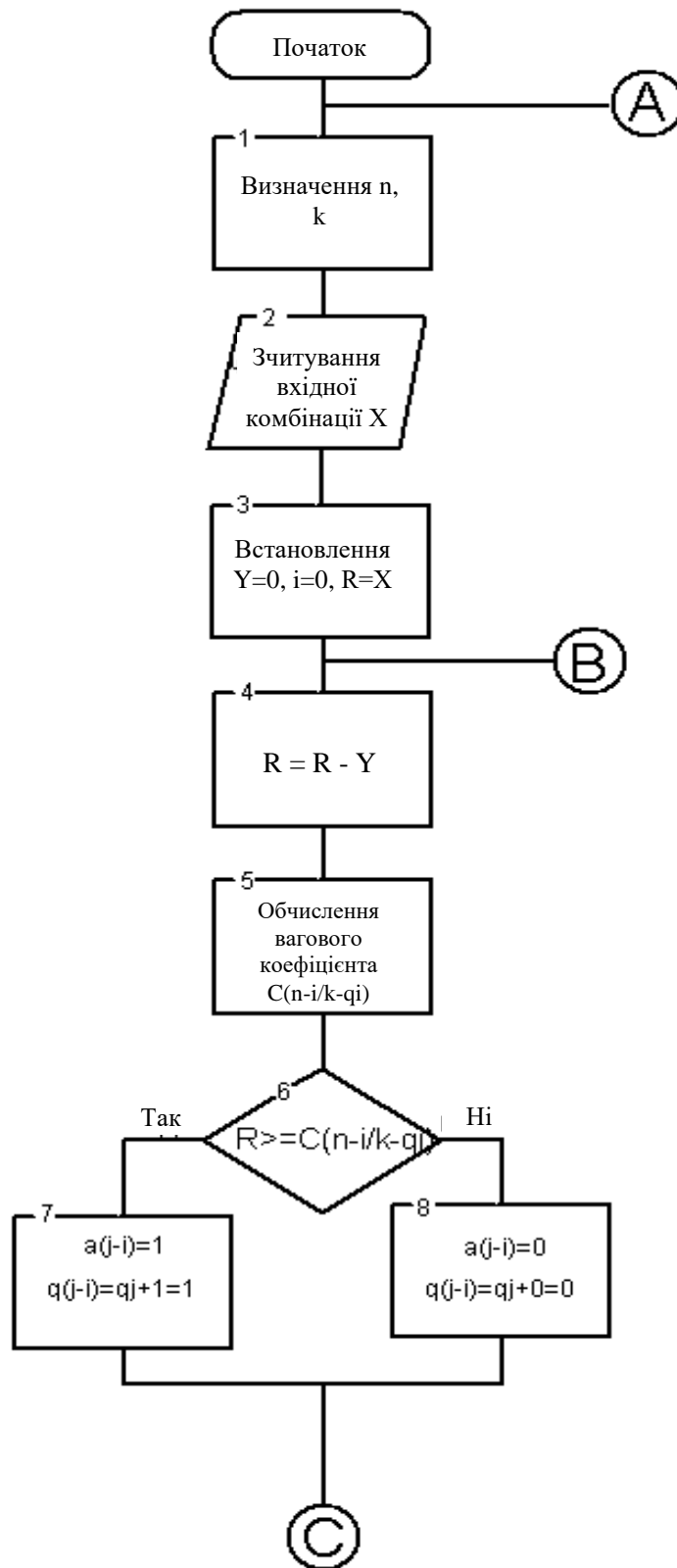
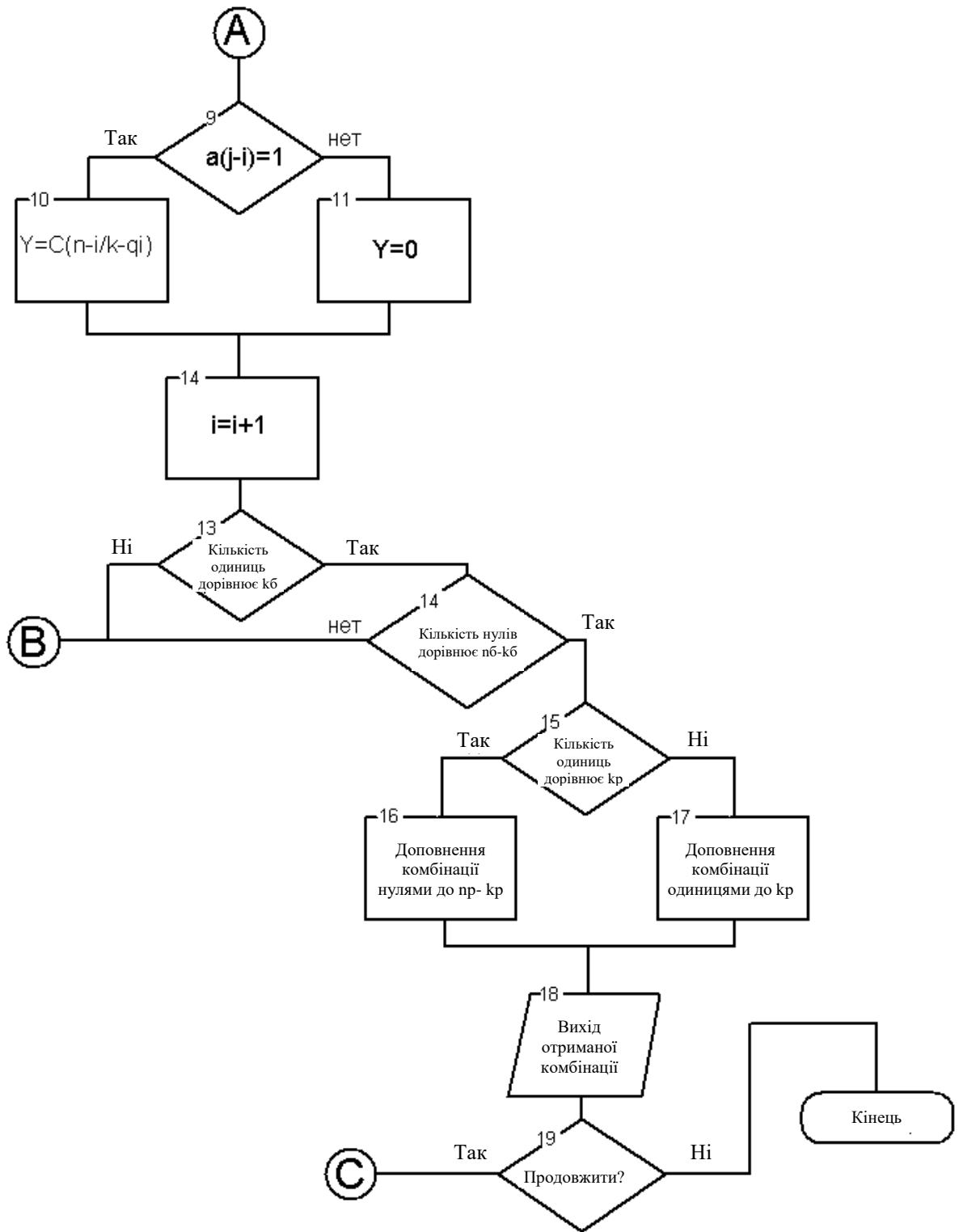


Рисунок 3.1 – Блок-схема алгоритму роботи пристрою рівноважного кодування



Продовження рисунку 3.1

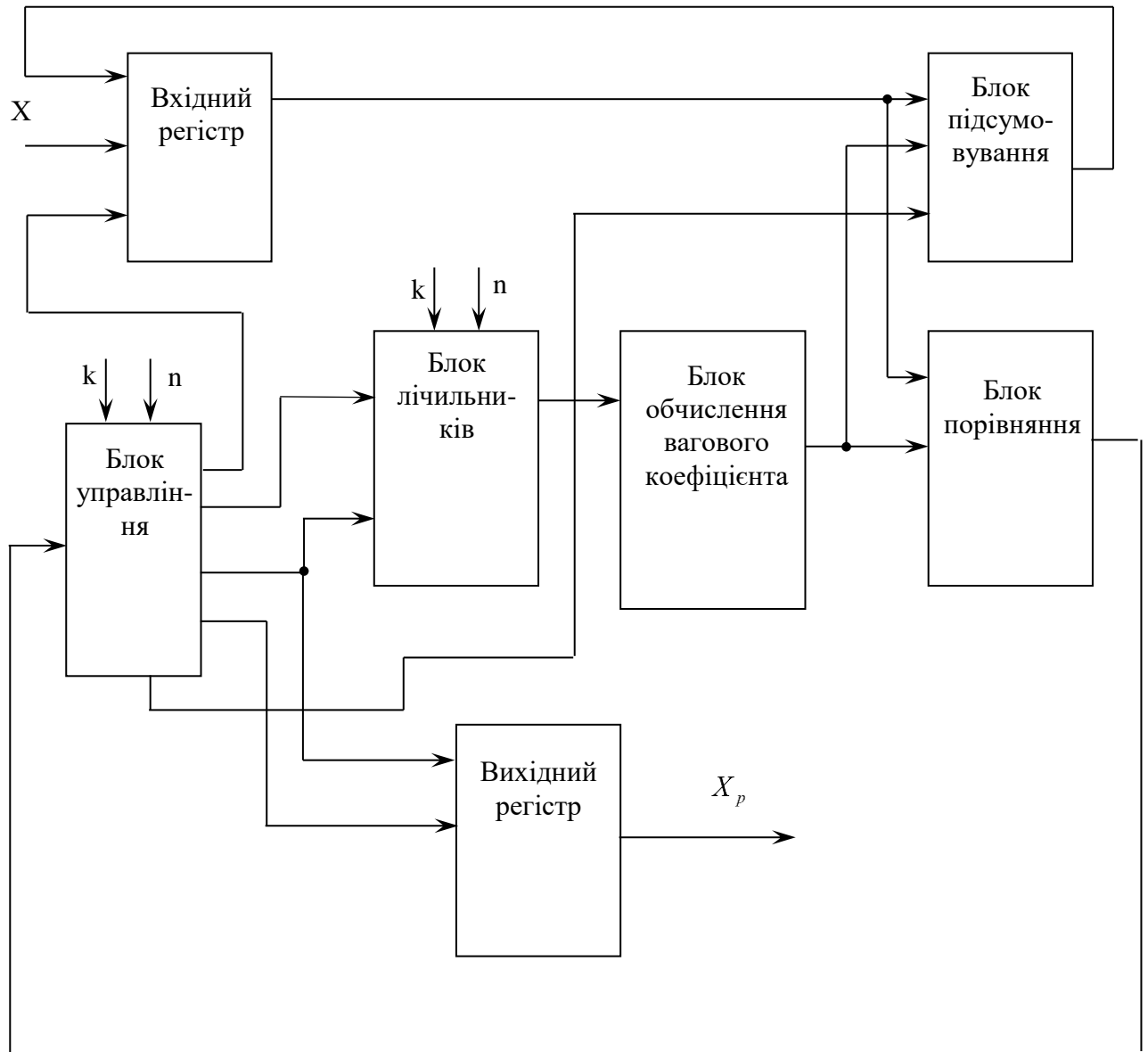


Рисунок 3.2 – Структурна схема кодуючого пристрою

Процедура формування наступних розрядів кодової комбінації буде повторюватися доти, поки кількість одиниць в отриманій послідовності не буде дорівнювати параметру k ($q_0 = k$) або кількість нулів не буде дорівнювати $n - k$ ($r - q_0 = n - k$). Після цього відбувається доповнення біноміальної комбінації нулями або одиницями. У випадку коли $q_0 = k$ додаються одиниці до значення k . А коли $r - q_0 = n - k$ - додаються нулі.

4 СИНТЕЗ ФУНКЦІОНАЛЬНОЇ СХЕМИ ПРИСТРОЮ КОДУВАННЯ

На функціональному рівні проєктований пристрій складається із програмної та апаратної частини. Синтез пристрою було вирішено проводити на основі мікропроцесорного комплекту. У результаті деякі блоки з структурної схеми будуть реалізовані з використанням мікросхем із мікропроцесорного комплекту, а частина блоків буде реалізована програмно [9].

Апаратну частину можна розділити на наступні функціональні блоки [9]:

- мікропроцесорний модуль;
- буфер адресів;
- буфер даних;
- операційний запам'ятовувальний пристрій (ОЗП);
- постійний запам'ятовувальний пристрій (ПЗП);
- дешифратор адресів;
- дешифратор зовнішніх пристроїв;
- синхронно-асинхронний приймач/передавач;
- паралельний інтерфейс;
- контролер переривань.

Мікропроцесор. Цей блок є основою всієї обчислювальної системи. Він призначений для обробки даних, які надходять на його виходи та для надсилання даних на потрібні пристрої. Даними можуть бути команди мікропроцесора або операнди для цих команд. Ці команди становлять яку-небудь програму. Обробкою даних називається процес виконання цієї програми. У функції мікропроцесора входить керування потоками даних, робота із зовнішніми пристроями [10, 11].

Буфер адресів. Його завданням є посилення сигналів адреси, що виходить з мікропроцесора, тому що з мікропроцесора виходить сигнал порівняно малої потужності. Буфер адресів являє собою регістр, який має входи даних, а також вхід дозволу записування в регістр STB. Сигнал дозволу записування надходить від мікропроцесора коли він починає працювати із запам'ятовувачами пристроями [10].

Буфер даних. Він служить для посилення та стабілізації сигналів даних від мікропроцесора. Так саме як й буфер адреси, він являє собою регістр з входами даних [10].

Оперативний запам'ятовуючий пристрій (ОЗП), у який буде записуватись вихідна двійкова комбінація, проміжні дані перетворення та вихідна рівноважна комбінація. Також необхідно помітити, що особливо високої швидкості обміну нам не потрібно, тому можна обмежитися восьмирозрядною шиною. Звичайно, можна організувати 16-розрядний обмін з нашим пристроєм, що дає більший вигреш у часі в порівнянні з 8-розрядним обміном, але зате він вимагає додаткових апаратурних витрат. Мікропроцесор виставляє на вхід запам'ятовуючого пристрою адресу, на якій перебувають дані. Також мікросхема містить вхід вибірки CS та вхід дозволу запису в ОЗП WE. Активним рівнем для обох входів є рівень нуля. При надходженні на вхід CS логічного нуля відбувається вибірка кристала. При логічній одиниці мікросхема не реагує на дані, що надійшли на входи. Вхід дозволу запису WE дозволяє вибрати режим роботи з ОЗП: при логічній одиниці дані зчитуються з адреси, що виставив мікропроцесор, та видаються на вхід мікросхеми, при логічному нулі дані пишуться по зазначеній адресі [10].

Постійний запам'ятовуючий пристрій (ПЗП). В нього буде записана мікропрограма для обробки даних. Ця мікросхема також містить входи CS та OE призначення яких відповідно у вибірці мікросхеми та дозволу роботи із пристроєм [10].

В якості способу обміну даних обираємо переривання, тому що їх використання дозволяє усунути необхідність виконання мікропроцесором неефективних операцій по перевірці готовності зовнішніх пристроїв до обміну та знижує витрати часу на очікування їхньої готовності. Адже між приходом першої двійкової послідовності та наступної, може минути досить великий час. Доцільно передбачити можливість виконання в цей період інших завдань [11].

При надходженні запиту переривання на один з входів IR та при задоволенні цього запиту контролер переривань виробляє вихідний сигнал запиту переривання, який надходить на процесор.

Контролер переривань може виконувати наступний набір операцій [11]:

- маскування запитів на переривання, тобто тимчасова заборона реакції на переривання;
- встановлення пріоритетів запитів по різних входах, тобто дозвіл конфліктів при одночасному надходженні декількох запитів на переривання.

Для маскування переривань використовується внутрішній регістр контролера, програмно доступний процесору як по запису, так й по читанню. Замаскованим може бути кожний запит (по кожній з ліній IRQ), шляхом встановлення відповідного біта маскування в записаному в контролер керуючому байті [11].

Схема пріоритетів переривань може бути задана процесором програмним шляхом. У базовому варіанті усі пріоритети є фіксованими (тобто IR0 має вищий пріоритет, а IR7 — нижчий). Але, вищий рівень пріоритету задається для кожного з входів запитів. Можна також встановити циклічне перемикання пріоритетів (останній запит, що був обслуговуваний, одержує нижчий пріоритет), забезпечуючи тим самим усім запитам рівні пріоритети. Якщо під час обробки переривання з меншим рівнем пріоритету надходить запит з вищим пріоритетом, то процесор переходить на програму обробки більш пріоритетного запиту, а після її виконання повертається до програми обробки запиту з меншим пріоритетом [11].

Завершивши виконання програми обробки переривання, процесору необхідно за допомогою спеціальної команди сповістити про це контролер переривань, щоб дозволити йому подальшу роботу, зокрема, знову обробляти той же самий запит.

Для обміну даними із зовнішніми пристроями використовується універсальний асинхронно-синхронний приймач-передавач, який приймає від центрального процесора інформацію в паралельному виді по шині даних та перетворює її в потік даних для передачі в послідовному форматі. Одночасно цей пристрій може приймати потоки даних у послідовному форматі та перетворювати їх у паралельний формат. При цьому пристрій послідовного введення-виводу сповіщає процесор про готовність прийняти новий байт даних для передачі або про приймання байту для процесора. В проектуваному пристрої використовується два приймачі-передавача: один на вхідну інформацію, а інший – на вихідну. У першому використовуються наступні входи: Rx – вхід запису інформації, що надійшла, для подальшого перетворення в паралельний код, який буде записаний в ОЗП; Rх – використовується для інформування схеми про готовність приймача прийняти дані; CTS – використовується для синхронізації із зовнішнім передавачем; DTR – на цей вхід надходять запити про готовність передавача; RTS – використовується для

запитів про готовність приймача; CLK – на цей вхід надходять тактові імпульси від мікропроцесора; RD – вхід дозволу читання (сигнал про дозвіл читання встановлює мікропроцесор); WR – вхід дозволу запису в приймач-передавач; CS – вхід вибірки мікросхеми; A0 – вхід для подання сигналу записування/зчитування даних (рівень «нуль» на вході показує на можливість запису або зчитування даних, рівень «одиниця» дозволяє записувати в пристрій керуючі слова або зчитувати інформацію про його стан). У другому приймачі-передавачі замість RxD та RxR використовуються виходи TxD та TxR. Перший вихід використовується для передачі даних, а другий для інформування про готовність передавати дані [11].

Дешифратор адреси використовується для вибору запам'ятовуючих пристроїв. На входи 1-7 подаються сигнали з шини адреси з одинадцятого до п'ятнадцятого розряду, а також сигнал зчитування RD та сигнал запису WR від мікропроцесора. Після аналізу розрядів дешифратор видає на вихід сигнали вибору оперативного або постійного запам'ятовувального пристрою. Схема дешифратора наведена на рисунку 4.1 [10, 11].

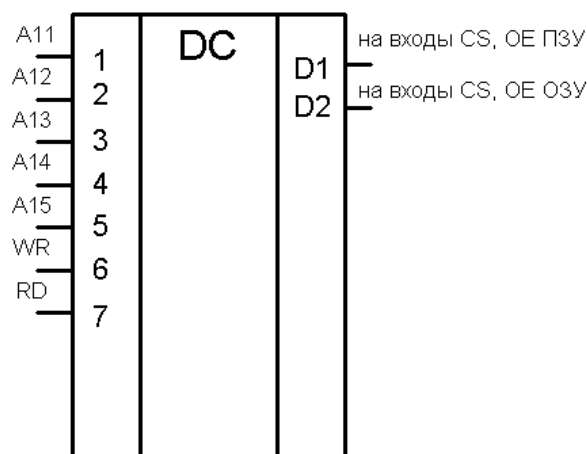


Рисунок 4.1 – Дешифратор адреси

Дешифратор зовнішніх пристроїв використовується для керування доступом мікропроцесорної системи до зовнішніх пристроїв – регістрів, універсального асинхронного приймача-передавача, інтерфейсного блоку.

На входи дешифратора подаються сигнали з шини адреси з сьомого по нульовий розряд, а також сигнал зчитування RD та сигнал запису WR від мікропроцесора. У результаті перетворень, на виході дешифратор видає сигнали

вибірки мікросхем, які надходять на вхід CS приймача-передавача, паралельного інтерфейсу та контролера переривань. Схема дешифратора показана на рисунку 4.2 [11].

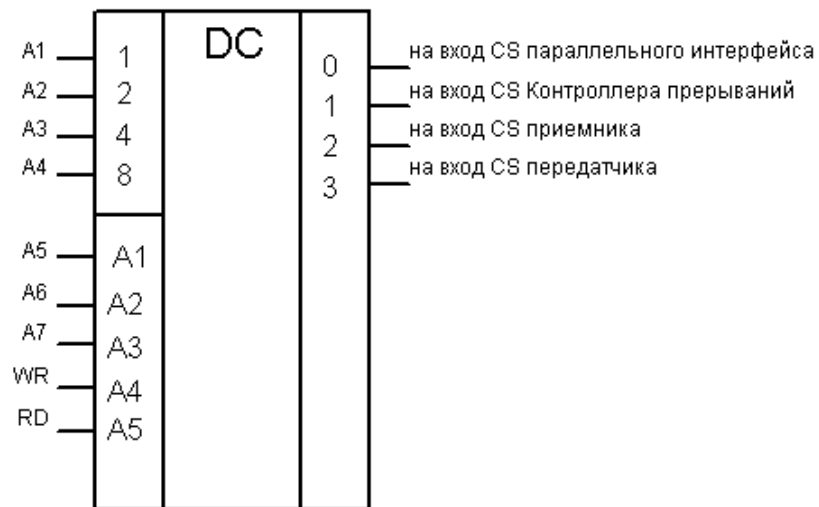


Рисунок 4.2 – Дешифратор зовнішніх пристроїв

Паралельний інтерфейс являє собою універсальний приймач/передавач, що приймає від центрального процесора інформацію в паралельному виді по шині даних та перетворює її в потік даних для передачі в паралельному форматі до периферійних пристроїв. Він забезпечує обмін інформацією між мікропроцесорною системою та зовнішніми пристроями по трьох 8-розрядних каналах А, В та С. У цьому випадку ми використовуємо порти А та В для роботи з двома універсальними асинхронними приймачами-передавачами, для зчитування інформації та видачі її в закодованому виді. Два розряди каналу С використовується для формування запитів переривань, які надходять на входи IR0 та IR1 контролера переривань [11].

5 РОЗРОБКА ПРИНЦИПОВОЇ СХЕМИ КОДУЮЧОГО ПРИСТРОЮ

5.1 Вибір елементної бази

Практична реалізація розроблювального пристрою повинна припускати оптимальний вибір реальних мікросхем, а також аналогових елементів. Основними критеріями вибору є: забезпечення необхідної швидкодії системи, енергоспоживання, а також мінімальної вартості пристрою, що проектується. У тих випадках, коли варто обирати одночасно між двома факторами, наприклад, між швидкістю роботи та завадостійкістю системи, варто прийняти або компромісне рішення, або врахувати той критерій, який є ключовим та забезпечує функціонування системи.

У роботі проектованого пристрою використовуються цифрові мікросхеми серії 1533, мікропроцесорний комплект серії 580, а також аналогові елементи, такі як резистори, конденсатори та діоди [11].

Основними параметрами мікросхем є вхідний струм логічного нуля $I_{вх}^0$, вхідний струм логічної одиниці $I_{вх}^1$, максимальний вихідний струм логічного нуля $I_{вих\ max}^0$, максимальний вихідний струм логічної одиниці $I_{вих\ max}^1$, максимальне значення вихідної напруги, що відповідає рівню логічного нуля, при якому забезпечується нормальна робота наступних ІС $U_{вих}^0$, мінімальне значення вихідної напруги, що відповідає рівню логічної одиниці, при якому забезпечується нормальна робота наступних ІС $U_{вих}^1$, коефіцієнт розгалуження по виходу визначає кількість входів елементів даної серії, яка може бути без порушення працездатності підключена до виходу попереднього логічного елемента, час затримки переходу ІС зі стану логічного нуля в стан логічної одиниці $t_{зад}^{0/1}$, час затримки переходу ІС зі стану логічної одиниці в стан логічного нуля $t_{зад}^{1/0}$, потужність, яка споживається базовим логічним елементом від джерела живлення $P_{спож}$, максимально припустиме значення статичної перешкоди $U_{пер}$, максимальна частота перемикання f_{max} . Ці основні параметри наведені в таблиці 5.1 [11].

Таблиця 5.1 – Основні параметри мікросхем серії КР1533

Параметр	Значення
$I_{вх}^0$, мА, \leq	-0,2
$I_{вх}^1$, мА, \leq	0,02
$I_{вих\ max}^0$, мА	4
$I_{вих\ max}^1$, мА	-0,4
$U_{вих}^0$, В, \leq	0,4
$U_{вих}^1$, В, \geq	2,5
$K_{роз}$	20
$t_{зад}^{0/1}$, нс, \leq	4
$t_{зад}^{1/0}$, нс, \leq	4
$P_{спож}$, мВт, \leq	1
$U_{пер}$, В, \leq	0,8
f_{max} , МГц, \leq	100

Висока швидкодія в сполученні з низькою споживаною потужністю та великою навантажувальною здатністю, широкий набір логічних та інтерфейсних мікросхем серії КР1533 дозволяють створювати обчислювальні пристрої цифрової автоматики з якісно новими характеристиками та високими техніко-економічними показниками [11].

Істотною особливістю серії К1533 є наявність інтерфейсних та буферних мікросхем, що мають підвищену навантажувальну здатність по виходу в стані високого та низького рівня й меншу, у порівнянні із серією К531, потужністю споживання при практично порівнянній швидкодії. У порівнянні з відомими серіями логічних ТТЛ-мікросхем, вона має мінімальне значення добутку швидкодії на потужність, що розсіюється [11].

Також приведемо основні параметри для серії 580 [11].

Основні електричні параметри МПК КР580:

- вихідна напруга логічної одиниці $U_{вих1}$, В $\geq 2,4$;
- вихідна напруга логічного нуля $U_{вих0}$, В $\leq 0,45$;

- вхідна напруга логічної одиниці $U_{\text{вх1}}, V \geq 2,0$;
- вхідна напруга логічного нуля $U_{\text{вх0}}, V \leq 0,8$;
- вихідний струм високого рівня $I_{\text{вих в}}, m \leq 2,0$;
- вихідний струм низького рівня $I_{\text{вих н}}, m \geq 2,0$.

5.2 Вибір мікропроцесорного модуля

При виборі елементів мікропроцесорної системи природньо починати саме з вибору самого процесора. Для вирішення завдань кодування достатньо використовувати восьмирозрядний процесор. Також бажано, щоб обраний процесор мав входи обробки переривань. Зазначеним вимогам відповідає мікропроцесор КМ1821ВМ85А (аналог – і8085А). Схематичне зображення процесора наведено на рисунку 5.1 [11].



Рисунок 5.1 – Мікропроцесор КМ1821ВМ85А

Живленням даного мікропроцесора є одне джерело +5 В. Виконаний мікропроцесор по КМОП технології. Процесор адресує простір запам'ятовуючого пристрою обсягом 64 Кбайт та простір зовнішніх пристроїв обсягом 256 Байт. Структура КР1821ВМ85А організована подібно структурі більш ранніх версій мікропроцесорних пристроїв, але має ряд відмінностей. Насамперед, даний мікропроцесор має вбудований системний генератор, який виконує функцію синхронізації мікропроцесора та часове прив'язування сигналів з входів початкового встановлення RESI та готовності RDY. Частота генератора визначається зовнішнім кварцовим резонатором, RC - ланцюжком, LC - контуром або зовнішнім джерелом, що підключається до входів X1, X2. На входи CLKO та RESO подаються сигнали синхронізації та початкового встановлення пристроїв введення/виводу, розташованих на системній шині [10, 11].

Внутрішній системний контролер формує загальні для запам'ятовуючих та зовнішніх пристроїв сигнали зчитування/запису на виходи WR, RD, а для розподілу доступу використовується вихід ІО/М. Адресна шина напівсполучена – на виводи A15...A8 виставляється адреса запам'ятовуючого пристрою або адреса зовнішнього пристрою A7...A0 на весь інтервал звертання до запам'ятовуючого або зовнішнього пристрою, а на виводи AD7...AD0 – у режимі розподілу часу спочатку адреса зовнішнього/запам'ятовуючого пристрою A7...A0, а потім дані. Адреса підключається до входів X1, X2. На входи CLKO та RESO видаються сигнали синхронізації та початкового встановлення пристроїв введення/виводу, які розташовані на системній шині [11].

Адреса на шині AD7...AD0 супроводжується адресним стробом на виході ALE, по якому він фіксується в зовнішньому регістрі. Стан процесора видається на виходи ST1, ST0, де ST0 визначає цикл запису, а ST1 – цикл зчитування.

Процесор має засоби обслуговування переривань з п'яти входів. Вхід INT призначений для маскуємих векторних переривань при спільному використанні контролера переривань. По входах TRAP, RST5.5, RST6.5, RST7.5 обіг здійснюється по фіксованих адресах запам'ятовуючого пристрою 0024, 002C, 0034, 003C відповідно. Входи

RST5.5, RST6.5, RST7.5 є програмно маскуємими, вхід TRAP немаскуємий та використовується для обслуговування аварійних ситуацій. Пріоритет входів переривань наступний: в порядку убубання – TRAP, RST7.5, RST6.5, RST5.5, INT [11].

5.3 Блок пам'яті

Для нормального функціонування проектованого пристрою досить невеликого обсягу ПЗП, тому доцільно використовувати ПЗП К573РФ2 обсягом 2 Кбайт (DD12). ІМС має 11 адресних входів, 8 виходів даних, входи дозволу програмування WE, вибірки кристала CS, дозволу виходів OE [10].

Адресні входи А0-А10 підключаються до шини адреси контролера, виходи даних D0-D7 – до шини даних. Вибірка кристала та дозвіл виходу управляються дешифратором ОЗП/ПЗП. Вхід дозволу програмування з'єднаний з джерелом живлення, тому що передбачається, що ІМС ПЗП споконвічно містить керуючу програму. При включенні живлення й після скидання, мікропроцесор завжди починає зчитувати код команди, розташований в комірці з адресами 0000H-087FH [10].

В якості проектованого ОЗП обираємо статичний ОЗП КР537РУ10 обсягом 2 Кбайт (DD13). Застосування ОЗП статичного типу дозволяє вирішити завдання збереження даних у пам'яті (на відміну від динамічного ОЗП статичне не вимагає циклів регенерації пам'яті). Це дозволяє істотно спростити апаратну частину контролера. Для запису даних з входів D0-D7 у мікросхему необхідно на входах А0-А9 встановити необхідну адресу комірки пам'яті, подати на входи CS та WR напругу низького рівня. Для зчитування даних з пам'яті необхідно встановити адресу комірки, на вхід CS подати напругу низького рівня, а на вхід WR – високого [10].

5.4 Дешифратор адреси ПЗП та ОЗП

Для розподілу області ПЗП та ОЗП необхідно дешифрувати верхні розряди адреси. Після подання сигналу скидання на процесор лічильник команд приймає значення 0, тобто виконання програми починається з адреси 0. Отже, область ПЗП, у якому зберігається код програми, повинна починатися з адреси

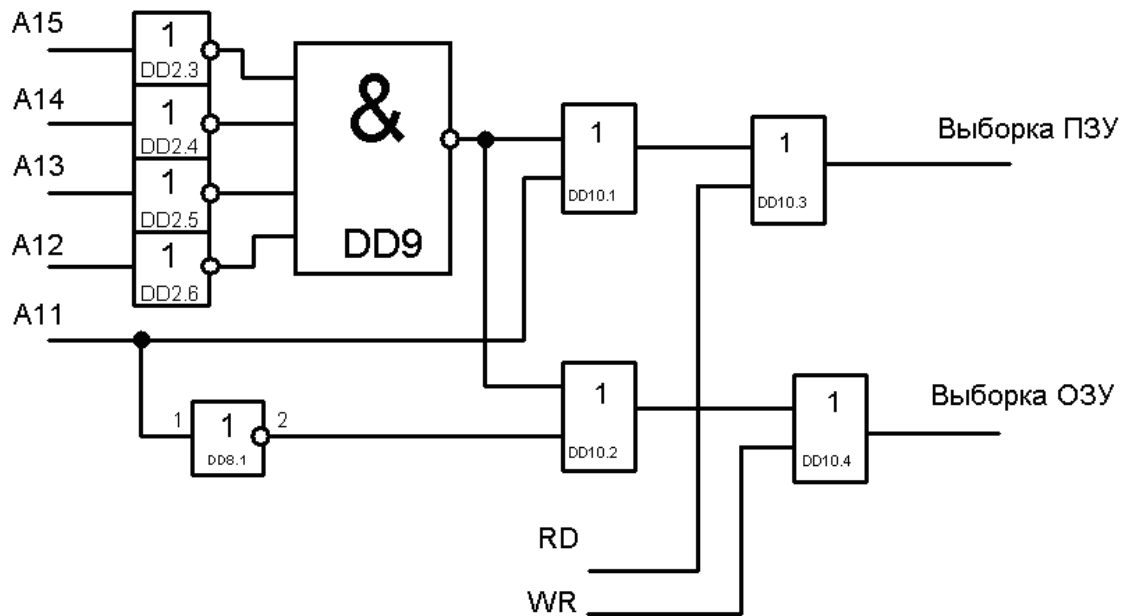


Рисунок 5.2 - Схема дешифратора адреси пам'яті

5.5 Дешифратор зовнішніх пристроїв

Для керування доступом мікропроцесорної системи до зовнішніх пристроїв (ЗП) (регістрів, універсального приймача-передавача, інтерфейсного блоку) необхідно застосовувати дешифратор адреси зовнішніх пристроїв. Процесор КР1821ВМ85А може адресуватися до 256 зовнішніх пристроїв, використовуючи для передачі адреси молодший байт шини адреси. У проектуваному пристрої для звертання до регістрів блоку індикації, регістру блоку виводу та до блоку введення аналогових даних буде використовуватися дешифратор К555ИДЗ (DD16) [10].

Дешифратор має вісім виходів з активним рівнем 1, чотири інформаційних входи, які підключаються до системної шини адреси А0-А3, та два входи дозволу, які використовуються для розподілу звертання до пам'яті, до інтерфейсного блоку та до інших зовнішніх пристроїв.

Дешифратор підключається до входів дозволу та повинен задавати адресний простір зовнішніх пристроїв, розділяти звертання до пам'яті та пристроїв. Таблиця 5.3 показує, за якою адресою організовується звернення МП до зовнішніх пристроїв [10].

Таблиця 5.3 – Адресний простір для зовнішніх пристроїв

Зовнішній пристрій, що підключається	Адреса		Розряди адресної шини							
	<10>	<16>	A7	A6	A5	A4	A3	A2	A1	A0
<i>KP580BB55 (port A)</i>	16	10	0	0	1	0	0	0	0	0
<i>KP580BB55 (port B)</i>	17	11	0	0	1	0	0	0	0	1
<i>KP580BB55 (port C)</i>	18	12	0	0	1	0	0	0	1	0
<i>KP580BB55 (унр.)</i>	19	13	0	0	1	0	0	0	1	1
<i>KP580BH59 (унр.)</i>	20	14	0	0	1	0	0	1	0	0
<i>KP580BH59 (маск.)</i>	21	15	0	0	1	0	0	1	0	1
<i>KP580BB51 (унр.)</i>	22	16	0	0	1	0	0	1	1	0
<i>KP580BB51 (введення)</i>	23	17	0	0	1	0	0	1	1	1
<i>KP580BB51 (унр.)</i>	24	18	0	0	1	0	1	0	0	0
<i>KP580BB51 (вивід)</i>	25	19	0	0	1	0	1	0	0	1

У стовпцях A4, A6, A7 дані звернення дорівнюють нулю, тому ці адреси заводяться на елемент DD14.1 прямо. A5 заводиться на DD14.1 через інвертор, тому що стовпець дорівнює одиниці на всіх використовуваних наборах. У такий спосіб ми перевіряємо, що відбувається звернення саме до зовнішніх пристроїв, а не до запам'ятовуючих пристроїв [10].

Порти з 16 по 19 (виходи дешифратора 1-4) ставляться до паралельного інтерфейсу (KP580BB55A). Порт А має адресу 10H, у схемі він використовується для введення двійкової послідовності. Порт В, що має адресу 11H, використовується для виводу закодованої послідовності на передавальний пристрій, що перебуває поза нашою схемою. Порт С, з адресою 12H, використовується для обслуговування переривань від KP580BB51. Адреса 13H використовується для записування керуючого слова в паралельний інтерфейс.

Адреси 14H та 15H резервуються за контролером переривань. Адреса 14H використовується для записування слів ініціалізації ICW1, ICW2, а також керуючого слова OCW1. Порт 15H використовується для маскуванню переривань [10].

Адреси 16Н,17Н,18Н,19Н зарезервовані за універсальним приймачем-передавачем. Перші дві зарезервовані за приймальним пристроєм, а інші – за передавальним. 16Н и 18Н використовуються для записування керуючих слів, а 17Н и 19Н – для введення та виведення даних відповідно [10].

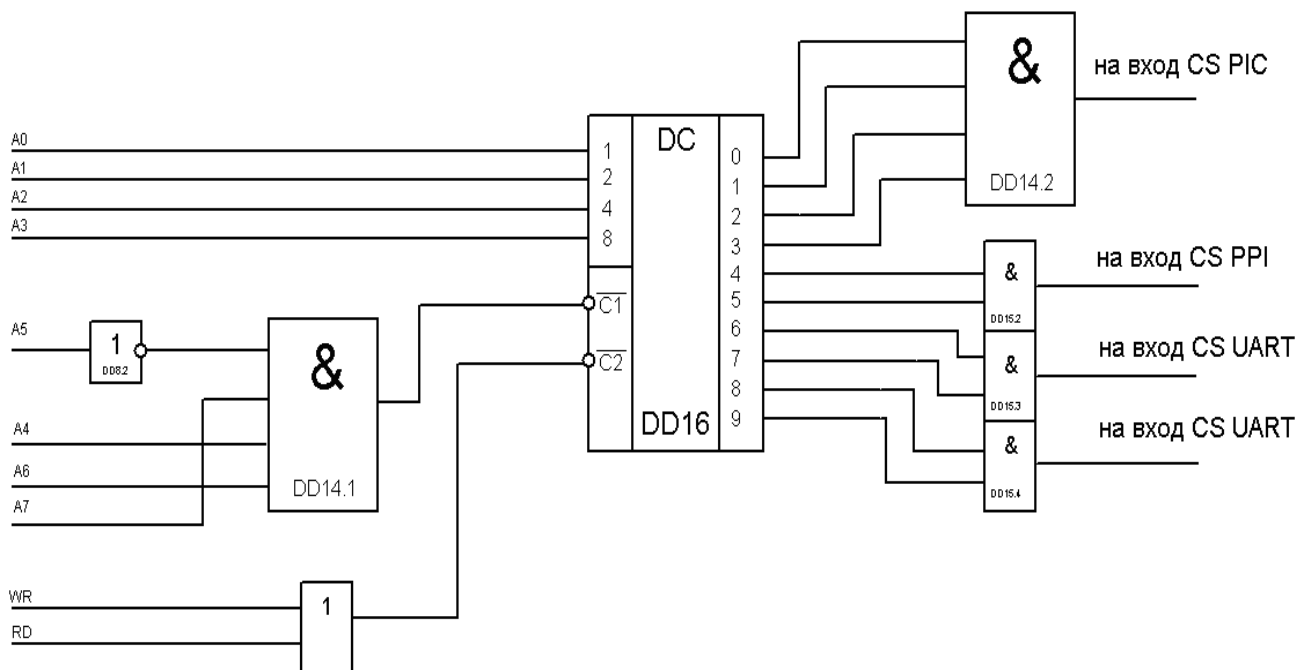


Рисунок 5.3 – Схема дешифратора зовнішніх пристроїв

5.6 Інтерфейсний блок

Для керування зовнішніми пристроями, а також для забезпечення необхідної швидкості введення отриманого коду в пам'ять, необхідно застосувати контролер KP580BB55 (DD11) – паралельний інтерфейс, що дозволяє організувати введення/вивід паралельної інформації різного формату [11].

Ця мікросхема складається із трьох восьмирозрядних портів введення/виводу PA, PB, PC. До того ж, порт PC може також працювати як два незалежних 4-х розрядних порти. Призначення виводів [11]:

- A1, A0 – адреси. Задають адресу поточного порту: 00 - PA, 01 - PB, 10 - PC, 11 - реєстр керуючого слова. Підключаються до 2-х молодших битів адресної шини;
- D7..D0 – шина даних;
- PA7..PA0, PB7..PB0, PC7..PC0 – відповідні виводи портів PA, PB, PC;
- RD – зчитування. Нульовий рівень означає, що процесор читає дані із шини даних, яка у цей момент підключена до порту (PA, PB, PC) залежно від адреси, обумовленої виводами A1, A0. Підключається до виводу системного контролера IOR;
- WR – запис. Нульовий рівень означає, що процесор видає дані на шину даних, підключену в цей момент до порту (PA, PB, PC) залежно від адреси, обумовленої виводами A1, A0. Підключається до виводу системного контролера IOW;
- CS – вибірка мікросхеми. Одиничний рівень переводить входи мікросхеми в Z-стан. При подачі сигналу з дешифратора ЗП задається адреса порту.

Порт А програмується на введення даних с DD7; порт В – на виведення даних на DD18; PC4, PC5 – на оброблення переривань від DD7 та DD18. Виходячи з цих міркувань, слово встановлення режимів: 10011000Y = 98H [11].

5.7 Розробка програмного забезпечення проектованого пристрою

Розроблений пристрій адаптивного рівноважного кодування для системи передачі даних побудовано на основі мікропроцесорної системи, тому для його функціонування необхідно програмне забезпечення. Програма складається мовою асемблер, використовуючи блок-схему алгоритму функціонування [9, 11].

У програмі є модулі підпрограм, що виконують наступні функції: обчислення сполучень, порівняння двох чисел.

Текст програми роботи проектованого пристрою:

```
CODE SEGMENT
ORG 0
DATA_n      DBO
DATA_k      DBO
DATA_koi    DBO
DATA_R      DBO
DATA_0      DBO
DATA_1      DBO
ORG 0F400H
START: MOV SP, 3FFH
BEGIN: IN AX, 01H
      MOV DATA_n, AN
      MOV DATA_k, AL
      IN AL, 02H
      MOV DATA_kOI, AL
      DEC AN
MЧ:   MOV BN, AN
      MOV BL, DATA_k
      MOV AX, [BX]
      MOV CL, DATA_n
      MOV DATA_R, CL
      CMP DATA_R, AX
      INB M1
      CLC
      RCL DX
      INC DATA_0
      JMP M2
M1:   STC
      RCL DX
      INC DATA_1
M2:   MOV AL, DATA_K
      CMP AL, DATA_1
```

```

        JZ M3
        MOV AN, DATA_n
        SUB AN, AL
        CMP AN, DATA_0
        MOV AN, DATA_n
        DEC AN
        JNZ M4
        JZ M7
M6:    MOV AL, DATA_1
        MOV AN, DATA_0
        ADD AN, AL
M3:    CMP AN, DATA_n
        JZ M5
        CLC
        RCL DX
        INC DATA_0
        JMP M6
M7:    MOV AL, DATA_1
        MOV AN, DATA_0
        ADD AN, AL
        CMP AN, DATA_n
        JZ M5
        STC
        RCL DX
        INC DATA_1
        JMP M8
M5:    OUT 03H, DX
        JMP BEGIN
        ORG 0FFFF0H
        JMP START
CODE ENDS
END

```

ВИСНОВКИ

У роботі був розглянутий спосіб кодування дискретної інформації з адаптацією до умов передачі, при якому забезпечується вибір оптимальних для даних умов передачі параметрів рівноважного коду.

У процесі виконання роботи була розглянута інформація про принципи завадостійкого кодування та способи побудови адаптивних систем.

Був розроблений пристрій адаптивного кодування, в основі якого лежить мікропроцесорний комплект. Особливістю даного пристрою є те, що алгоритм кодування реалізується програмним шляхом, що дозволяє розширювати функціональні можливості пристрою за рахунок написання нових модулів. Недоліком є збільшення при цьому часу кодування інформації, що знижує швидкість всієї системи в цілому. Розроблений пристрій може входити до складу мікропроцесорної системи збору, обробки та передачі даних. Запропонований спосіб кодування та пристрій для його реалізації доцільно будувати на базі сполучення з персональним комп'ютером IBM PC або відомими мікроконтролерами.

При використанні даного пристрою досягається значний позитивний ефект, оскільки вибираються оптимальні параметри завадостійкого коду. Це дозволяє скоротити час використання каналу зв'язку користувачем та підвищити ефективність експлуатації лінії в цілому.

СПИСОК ЛІТЕРАТУРИ

1. Signals Transmission In Network Communication. – June 2018. – <http://ecestudyaid.blogspot.com/2018/07/signals-transmission-in-network.html>.
2. 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.
3. 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.
4. Information Theory And Coding. – March 2016. – <https://educheer.com/essays/information-theory-and-coding>.
5. Варфоломеєва 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.
6. Беркман Л.Н., Варфоломеєва 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.
7. Coding theory. – July 2018. - https://wiki2.org/en/Coding_theory.
8. Information theory and coding. – 2019. – <https://uclouvain.be/en-cours-2019-LINGI2348>.
9. Ковалюк Т.В. Алгоритмізація та програмування: підручник / Т.В. Ковалюк. – Львів: Магнолія, 2013. – 400 с.
10. Швець Є.Я. Матеріали і компоненти електроніки: навчальний посібник / Є.Я. Швець, І.Ф. Червоний, Ю.В. Головка. – Запоріжжя: ЗДІА, 2014. – 278 с.
11. 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.