

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра електроніки і комп'ютерної техніки

«До захисту допущено»

Завідувач кафедри

_____ Анатолій ОПАНАСЮК

_____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття освітнього ступеня бакалавра

зі спеціальності 171 «Електроніка»,
освітньо-професійної програми «Електронні системи та компоненти»
на тему:

Генератор рівноважних кодових комбінацій на основі двійкових
біноміальних чисел

Здобувача групи ЕС-91 Бородіна Владислава Миколайовича

Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають
посилання на відповідне джерело.

Владислав БОРОДІН

Керівник, старший викладач,
кандидат фізико-математичних наук, доцент

Олексій Д'ЯЧЕНКО

Сумський державний університет

Факультет ЕлІТ **Кафедра** електроніки і комп'ютерної техніки
Спеціальність 171 Електроніка

ЗАТВЕРДЖУЮ:

Зав. кафедри ЕКТ

Анатолій ОПАНАСЮК

«__» _____ 2023 р.

Завдання

на кваліфікаційну роботу бакалавра

Бородіну Владиславу Миколайовичу

1. Тема проєкту: Генератор рівноважних кодових комбінацій на основі двійкових біноміальних чисел

затверджено наказом університету від «30» березня 2023 р. № 0310-VI

2. Термін здачі студентом закінченого проєкту 31 травня 2023 р.

3. Вихідні дані до проєкту: використання в якості проміжної системи числення біноміальної системи числення з двійковим алфавітом; розрядність двійкового коду – 4 розряди, довжина біноміального коду – 8, кількість одиниць – не більше 4, кількість нулів – не більше 4.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які підлягають розробці)

Вступ, Розробка та обґрунтування алгоритму функціонування та структурної схеми, Розробка та розрахунок принципових електричних схем, вузлів та блоків пристрою, Висновки.

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

1. Схема алгоритму функціонування;

2. Схема електрична структурна;

3. Схема електрична функціональна;

4. Схема електрична принципова

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Найменування етапів дипломного проєкту	Термін виконання етапів проєкту	Примітка
1.	Огляд літератури по тематиці проєкту	17.04.2023	
2.	Розробка схеми алгоритму	24.04.2023	
3.	Розробка схеми електричної структурної	02.05.2023	
4.	Розробка схеми електричної функціональної	08.05.2023	
5.	Розробка схеми електричної принципової	15.05.2023	
6.	Оформлення графічної частини	22.05.2023	
7.	Оформлення пояснювальної записки	29.05.2023	
8.	Рецензування роботи та підготовка до захисту	01.06.2023	

Студент-дипломник _____
(підпис)

Керівник проєкту _____
(підпис)

АНОТАЦІЯ

Пояснювальна записка до випускної кваліфікаційної роботи на тему: «Генератор рівноважних кодових комбінацій на основі двійкових біноміальних чисел» містить 51 сторінку, 18 рисунків, 7 таблиць, 13 джерел. Графічна частина представлена структурною схемою, блок-схемою алгоритму функціонування, функціональною та принциповою схемою генератора рівноважних кодових комбінацій на основі двійкових біноміальних чисел, а також принциповою схемою формувача рівноважних кодів, побудованого за системою реалізації булевих функцій. Виконано порівняння розроблених структур.

Розроблені пристрої характеризуються незначним споживанням потужності і достатньою швидкістю в результаті застосування інтегральних схем ТТЛ логіки останнього покоління - серії К 1533.

Ключові слова: біноміальна система числення, генератор рівноважних кодів, двійкові коди, діапазон, завадостійке кодування, лічильник нулів, лічильник одиниць, метод порозрядного зваження, схемотехнічна реалізація, функціональний базис.

ЗМІСТ

		стор.
Вступ		4
1	Розробка та обґрунтування алгоритму функціонування та структурної схеми	7
1.1	Огляд літератури та постановка задачі проектування	7
1.1.1	Кодування та двійкові коди	7
1.1.2	Рівноважні коди	9
1.1.3	Біноміальні системи числення з двійковим алфавітом	12
1.1.4	Постановка задачі проектування	13
1.2	Розробка та обґрунтування алгоритму функціонування та структурної схеми пристрою, що проектується	14
1.3	Розробка функціональної схеми генератора рівноважних кодових комбінацій	21
1.3.1	Перетворювач двійкових кодів в рівноважні кодові комбінації	21
1.3.2	Розробка функціональної схеми формувача рівноважних кодових комбінацій з застосуванням біноміальних кодів	24
2	Розробка та розрахунок принципових електричних схем, вузлів та блоків пристрою	33
2.1	Вибір елементної бази	33
2.1.1	Метою вибору елементної бази є	33
2.1.2	Інтегральні мікросхеми (ІМС) серії К1533	34
2.2	Вибір та розрахунок принципових схем пристрою, що проектується	34
2.2.1	Синтез перетворювача кодів	34
2.2.2	Розробка віднімача на основі двійкового суматора	44
2.2.3	Розробка блоку порівняння	46
2.2.4	Вибір регістрів	47
	Висновки	48
	Література	50

<i>ЕЛІТ 6.171.00.10.047 ПЗ</i>				
<i>Зм.</i>	<i>Арк</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>
<i>Розроб.</i>		<i>Бородін В.М.</i>		
<i>Перевір.</i>		<i>Д'яченко О.В.</i>		
<i>Реценз.</i>				
<i>Н. Контр.</i>				
<i>Затверд.</i>		<i>Опанасюк А.</i>		
<i>Генератор рівноважних кодових комбінацій на основі двійкових біноміальних чисел. Пояснювальна записка</i>				
		<i>Літ.</i>	<i>Аркуш</i>	<i>Аркушів</i>
			3	51
СумДУ зр.ЕС-91				

ВСТУП

Розробка і впровадження цифрових методів і пристроїв зберігання, обробки, перетворення і передачі сигналів є в даний час головним напрямком розвитку телебачення, систем зв'язку, інформаційно-вимірювальних та інформаційно-керуючих систем різного призначення. На сьогодні найбільш актуальним є розробка систем керування, які працюють в режимі реального часу, але все одно забезпечують високі показники завадостійкого керування.

При побудові систем автоматичного управління та контролю як у транспортній галузі, так і в промисловості широко застосовуються методи завадостійкого кодування. Кодування використовують при виборі архітектури керуючих комплексів та способів внесення надмірності в апаратні та програмні засоби для досягнення високих показників надійності. Крім того, завадостійке кодування застосовують при передачі даних між керуючими обчислювальними комплексами та периферійними об'єктами автоматики. Кодування використовують і при виборі способів синтезу контролепридатних структур компонентів систем управління, а також при виборі засобів діагностичного забезпечення окремих блоків і пристроїв. Способи кодування, які при цьому застосовуються, вкрай різноманітні, а вибір того чи іншого визначається «вихідними даними» - набором інформації про те, від яких видів несправностей або помилок захищають технічний об'єкт і які при цьому є обмеження на складність апаратних засобів і часу на виконання обчислювальних процедур.

Часто при побудові систем автоматичного керування та контролю використовуються блочні рівномірні коди. Серед них застосовуються як коди, орієнтовані на корекцію помилок (коди Хеммінга, Ріда-Соломона та ін.), так і коди, орієнтовані тільки на їх виявлення (коди Бергера, коди рівноважні та ін.). Для корекції помилок у кодових векторах у загальному випадку потрібна більша надмірність, ніж для їх виявлення, а значить, більші апаратні або часові витрати на реалізацію «самого захисту». Крім того, в деяких технічних завданнях вимагається використання перешкодостійкого кодування для виправлення помилок (наприклад, при передачі даних до периферійних об'єктів), а в деяких така властивість коду може призвести до накопичення відмов і зниження надійності (наприклад, при організації діагностичного забезпечення системи або пристроїв). Кодування особливо важливе у вирішенні сучасних завдань побудови систем автоматичного управління та контролю.

Основним способом створення високонадійних та безпечних систем управління відповідальними технологічними процесами є внесення надмірності в апаратні та програмні засоби їх компонентів. Це потребує наявності розвинених технічних засобів діагностування, якими постачаються окремі блоки систем управління. Широко застосовуються також методи виявлення несправностей в процесі експлуатації технічних об'єктів, у тому числі методи робочого (функціонального) діагностування. У системі функціонального діагностування контрольований блок (об'єкт діагностування) забезпечується технічними засобами діагностування, що дозволяють визначати технічний стан у процесі виконання ним своїх функцій. Найпростіше вирішується завдання не ідентифікації конкретного об'єкта діагностування, що відмовив, а контролю правильності обчислень, або контролю справності об'єкта діагностування. Саме таке завдання вирішують системи функціонального контролю (ФК).

Системами ФК забезпечуються, зазвичай, комбінаційні складові логічних пристроїв автоматики, а для компонентів, які мають пам'ять (наприклад, регістрів пам'яті), організується періодичне тестування. При синтезі систем ФК набули поширення завадостійкі двійкові коди. З усього різноманіття таких кодів для синтезу систем ФК можна використовувати лише рівномірні коди – коди з постійною довжиною кодових слів. Це зумовлено специфікою розв'язуваного завдання. Надмірність коду безпосередньо визначає і надмірність системи ФК. Прийнято порівнювати надмірність систем ФК із надмірністю систем дублювання, при організації яких використовуються коди з повторенням. Чим менша надмірність системи ФК, тим ефективніша вона в порівнянні з системою дублювання за показниками складності технічної реалізації. Однак при дублюванні виявляються будь-які поєднання несправностей в об'єкті діагностування, а при використанні кодів з меншою надмірністю частина несправностей може бути не виявлена.

Оскільки контроль стану пристроїв здійснюється за результатами обчислень, а їх структури можуть бути такими, що за певних обмежень на моделі несправностей виникнення ряду помилок на виходах неможливо, ефективними можуть виявитися завадостійкі коди зі зменшеною надмірністю порівняно з кодами з повторенням. До таких кодів належать рівноважні коди, різні коди з підсумовуванням та їх модифікації, коди Хеммінга, Ріда - Соломона, Геллагера (з малою щільністю перевірок на парність, LDPC-коди) та ін.

При побудові надійних та стійких до відмови систем автоматики досить ефективним буде застосування рівноважних кодів. Дані коди призначені для виявлення та корекції одноразових помилок у кодових словах за умови односпрямованого (монотонного) прояву спотворень. Їх можна використовувати як на етапі проектування пристроїв та систем автоматики при виборі контролепридатних архітектур та способів реалізації технічних засобів діагностування, так і при захисті даних на різних рівнях організації систем (у тому числі при взаємодії різнорідних систем, наприклад, стаціонарної системи керування рухом поїздів) та бортових систем локомотивів).

Таким чином, розробка генераторів (формуваців) рівноважних кодів на сьогодні є актуальним завданням, а вибір найкращої схемотехнічної реалізації таких формуваців – ще й цікавим дослідженням.

1 РОЗРОБКА ТА ОБГРУНТУВАННЯ АЛГОРИТМУ ФУНКЦІОНУВАННЯ ТА СТРУКТУРНОЇ СХЕМИ

1.1 Огляд літератури та постановка задачі проектування

1.1.1 Кодування та двійкові коди. Код – це взаємно однозначне відображення кінцевої впорядкованої множини символів, що належить деякому кінцевому алфавіту, в іншу, не обов'язково впорядковану множину, яка, як правило, має більшу кількість символів для кодування, передачі, зберігання або перетворення інформації.

Процес перетворення повідомлення у комбінацію символів відповідно до коду називається кодуванням, процес відновлення повідомлення з комбінації символів називається декодуванням.

Коди можна класифікувати за різними ознаками:

1. За основою (кількості символів в алфавіті):
бінарні (виконавчі $m = 2$) і **не бінарні** ($m \neq 2$).
2. За довжиною кодових комбінацій (слів): **рівномірні** - якщо всі кодові комбінації мають однакову довжину; **нерівномірні** - якщо довжина кодової комбінації не постійна.
3. За способом передачі: **послідовні** і **паралельні; блокові** - дані спочатку вкладаються у буфер, а потім передаються в канал і бінарні **безперервні**.
4. За завадостійкістю: **прості** (примітивні, повні) - для передачі інформації використовують всі можливі кодові комбінації (без надмірності); **коригувальні** (перешкодозахисні) - для передачі повідомлень використовують не всі, а тільки частина (дозволених) кодових комбінацій.
5. У залежності від призначення і застосування умовно можна виділити наступні типи кодів: **Внутрішні коди** - це коди, що використовуються усередині пристроїв. Це машинні коди, а також коди, що базуються на використанні позиційних систем числення (двійковий, десятковий, двійково-десятковий, вісімковий, шістнадцятковий та ін.) Найбільш поширеним кодом в ЕОМ є двійковий код, який дозволяє просто реалізувати апаратні пристрої для зберігання, обробки і передачі даних в двійковому коді. Він забезпечує високу надійність пристроїв і простоту виконання операцій над даними в двійковому коді. Двійкові дані, об'єднані в групи по 4, утворюють шістнадцятковий код, який

добре узгоджується з архітектурою ЕОМ, що працює з даними кратними байту (8 біт). **Коди для обміну даними та їх передачі каналами зв'язку.** Широке поширення в ПК отримав код ASCII (American Standard Code for Information Interchange). ASCII - це 7-бітний код буквено-цифрових та інших символів. Оскільки ЕОМ працюють з байтами, то 8-й розряд використовується для синхронізації або перевірки на парність, або розширення коду. У ЕОМ фірми ІВМ використовується розширений двійково-десятковий код для обміну інформацією EBCDIC (Extended Binary Coded Decimal Interchange Code). У каналах зв'язку широко використовується телетайпний код МККТТ (міжнародний консультативний комітет з телефонії та телеграфії) і його модифікації (МТК та ін.) При кодуванні інформації для передачі каналами зв'язку, в тому числі всередині апаратних трактів, використовуються коди, що забезпечують максимальну швидкість передачі інформації, за рахунок її стиснення та усунення надмірності (наприклад: коди Хаффмана і Шеннона-Фано), і коди які забезпечують найкращу вірогідність передачі даних, за рахунок введення надмірності в повідомлення, що передаються (наприклад: групові коди, Хеммінга, циклічні та їх різновиди). **Коди для спеціальних застосувань** - це коди, призначені для вирішення спеціальних завдань передачі і обробки даних. Прикладами таких кодів є циклічний код Грея, який широко використовується в АЦП кутових і лінійних переміщень. Коди Фібоначчі використовуються для побудови швидкодіючих і завадостійких АЦП.

Таким чином, **мету кодування** можна сформулювати наступним чином:

1. Підвищення ефективності передачі даних, за рахунок досягнення максимальної швидкості передачі даних.
2. Підвищення завадостійкості при передачі даних, а також при вирішенні завдання керування в системах автоматичного управління.

У відповідності з цими цілями теорія кодування розвивається у двох основних напрямках:

1. **Теорія економічного (ефективного, оптимального) кодування** займається пошуком кодів, що дозволяють в каналах без перешкод підвищити ефективність передачі інформації за рахунок усунення надмірності джерела і найкращого узгодження швидкості передачі даних з пропускнуною спроможністю каналу зв'язку.

2. **Теорія завадостійкого кодування** займається пошуком кодів, що підвищують вірогідність передачі інформації в каналах з перешкодами та покращують ефективність керування в САУ.

Отже, кожен код має свої характеристики, наприклад:

- довжина коду;
- основа коду;
- потужність коду;
- повне число кодових комбінацій;
- число інформаційних символів;
- число перевірних символів;
- збитковість коду;
- швидкість передачі;
- вага кодової комбінації і т.і.

Ці та інші характеристики кодів наведені на рисунку 1.1.

В залежності від того, які властивості має код, кожен з них отримав свою назву і сферу застосування. На рисунку 1.2 показана класифікація двійкових кодів.

1.1.2 Рівноважні коди. Рівноважні коди відносяться до завадостійких кодів. Завадостійкі коди – один з найбільш ефективних засобів забезпечення високої вірності як при зберіганні і при передачі дискретної інформації., так і при застосуванні в системах керування складними технологічними процесами. Досить ефективно рівноважні коди застосовуються при вирішенні питання діагностики пристроїв. Під завадостійкими кодами розуміють коди, що дозволяють виявляти і виправляти помилки, які виникають у результаті впливу завад. Завадостійкість кодування забезпечується за рахунок введення надмірності в кодові комбінації, тобто за рахунок того, що не всі кодові комбінації використовуються для передачі інформації або при вирішенні задач керування.

Коди з постійною вагою або рівноважні кодові комбінації відносять до класу рівномірних кодів. Дані коди є блочними нероздільними кодами, оскільки у них відсутня можливість розділити інформаційну та перевірочну частини кодових комбінацій. Всі комбінації цього коду мають однакову вагу (однакова кількість одиниць). Прикладом є відомий код «3 з 7».

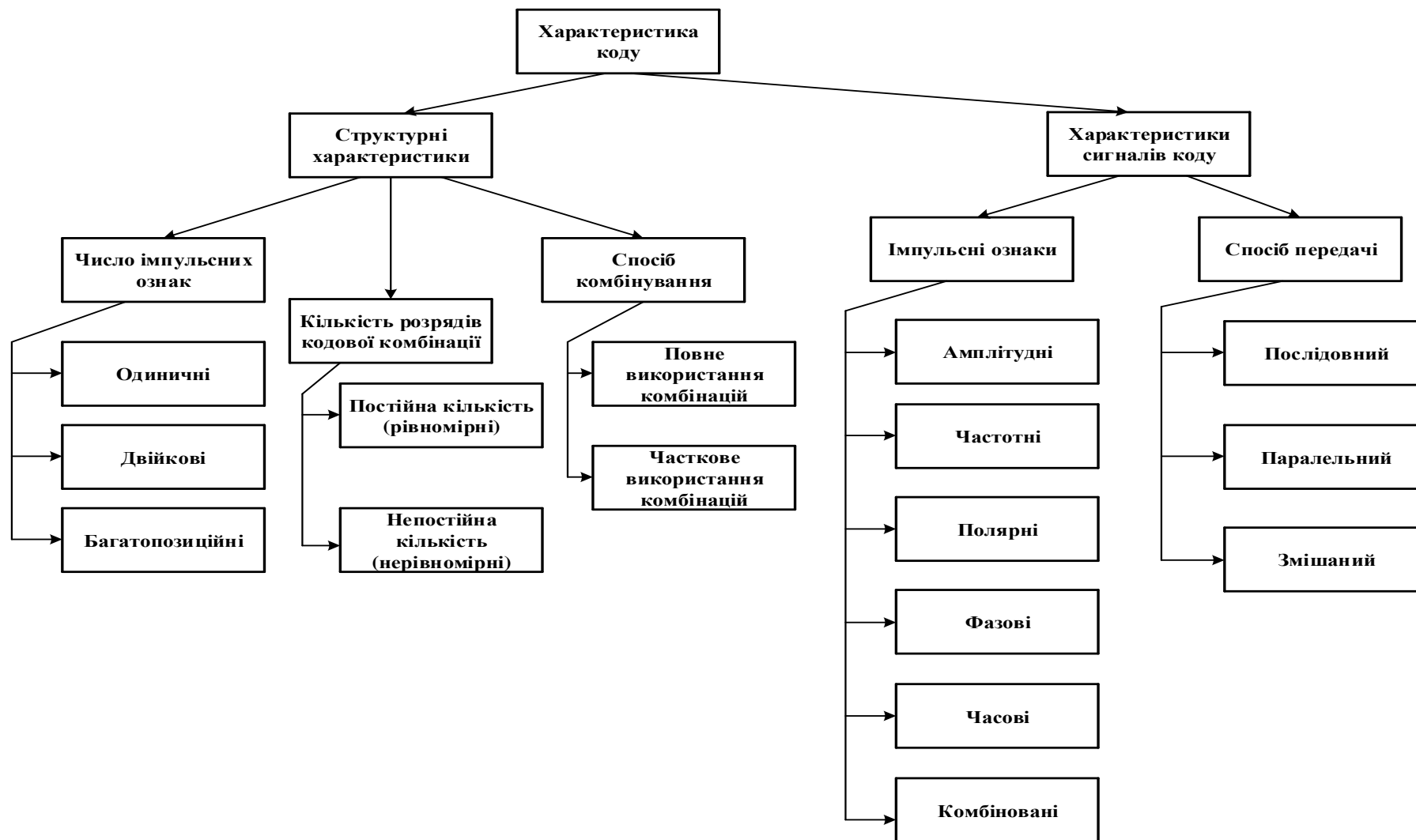


Рисунок 1.1 - Характеристики кодів

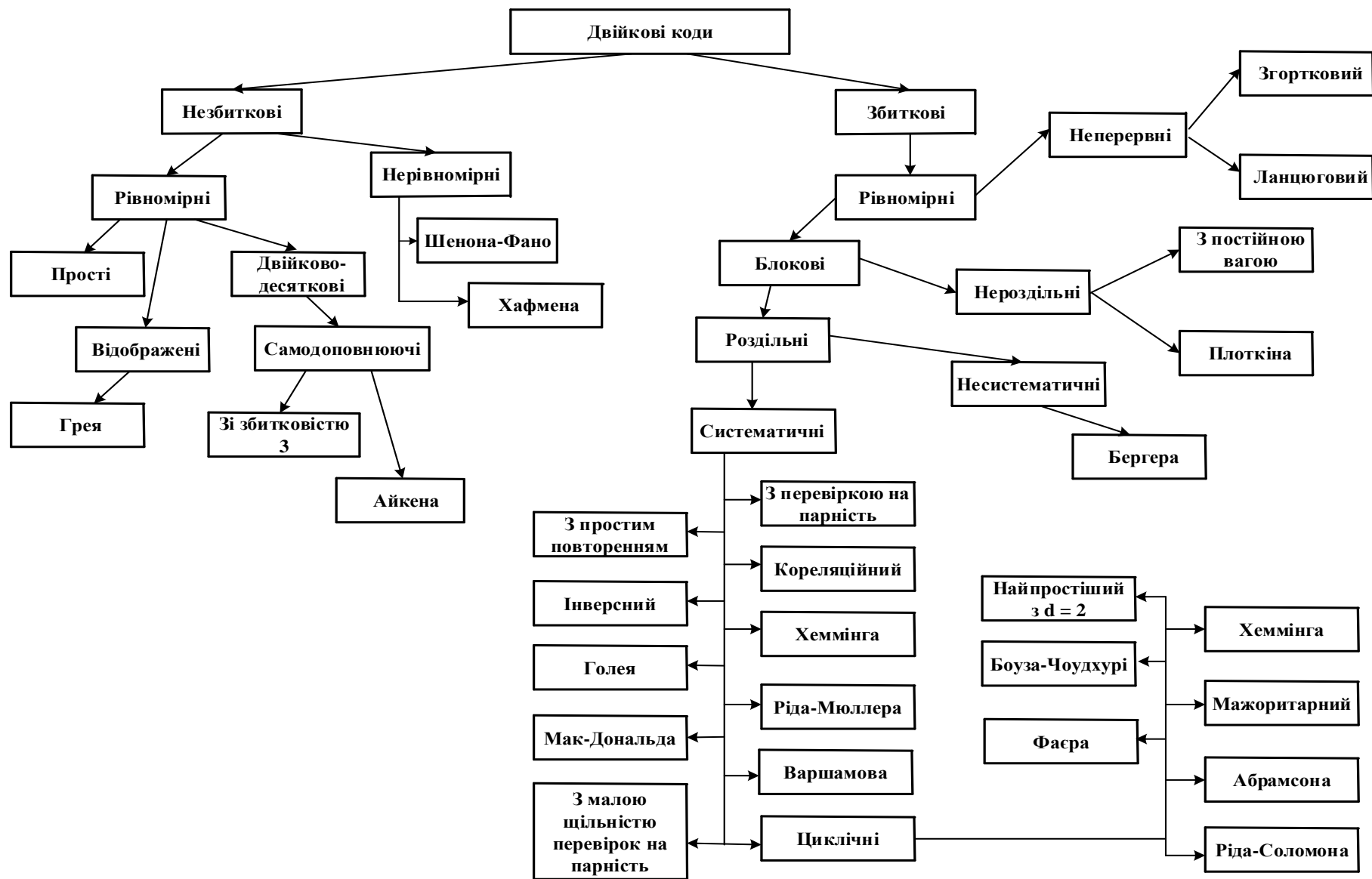


Рисунок 1.2 - Класифікація двійкових кодів

Головною з причин застосування цих кодів як в каналах зв'язку, так і в системах керування є здатність рівноважних кодів виявляти всі можливі комбінації непарної кількості помилок, з яких вони не виявляють лише ті, при яких перетворення $0 \rightarrow 1$ компенсується перетворенням $1 \rightarrow 0$.

Існує думка, що головним недоліком рівноважних кодів є складність їх генерації. На сьогодні застосовуються два найпоширеніших методи: перший – це побудова генератора рівноважних кодів як стандартного перетворювача довільних кодів, при цьому забезпечується максимальна швидкодія перетворення, але для різних параметрів коду треба будувати окремий перетворювач. А другий – більш ефективний з точки зору універсальності – один і той пристрій може генерувати рівноважні кодові комбінації з різними параметрами – це формувач на базі біноміальних двійкових чисел. При цьому перетворення виконується за два кроки – спочатку формується біноміальне число, а потім здійснюється перехід від нього до рівноважного коду.

1.1.3 Біноміальні системи числення з двійковим алфавітом. В основі біноміальних систем числення лежить розбиття вихідної множини з кількістю чисел у них, що задаються біноміальними коефіцієнтами.

Біноміальною системою числення з двійковим алфавітом називається функція:

$$F = x_{r-1} C_{n-1}^{k-q_r} + \dots + x_i C_{n-r+i}^{k-q_{i+1}} + \dots + x_0 C_{n-r}^{k-q_1} \quad (1.1)$$

з системами обмежень:

$$\begin{cases} q_0 = k_1 \\ r \leq n - 1 \\ n - k = r - q_0 \\ q_0 \leq k - 1 \end{cases}$$

де $x_i \in \{0,1\}$;

r - кількість розрядів біноміального числа;

k - максимальна кількість одиниць у біноміальному числі;

i - порядковий номер розряду;

n - параметр системи числення;

q_i - сума одиничних значень цифр від $(r-1)$ -го розряду до i -го включно;

$$q_i = \sum_{j=i}^{r-1} x_j.$$

Дві системи обмежень двійкової біноміальної системи числення визначають правила формування біноміальних чисел та їх особливості. Всі двійкові біноміальні числа з зазначеними параметрами можна розділити на дві групи – перші містять k – одиниць, а другі - $n - k$ – нулів. В таблиці 1.1 наведено приклад біноміальних кодових комбінацій з параметрами $n = 6, k = 4$.

Таблиця 1.1 – Біноміальні числа з параметрами $n = 6, k = 4$ та діапазоном

$$P = C_n^k = \frac{n!}{k!(n-k)!} = C_6^4 = \frac{6!}{4!2!} = 15$$

№	Біноміальне число	№	Біноміальне число	№	Біноміальне число
0	0 0	5	1 0 0	10	1 1 0 1 0
1	0 1 0	6	1 0 1 0	11	1 1 0 1 1
2	0 1 1 0	7	1 0 1 1 0	12	1 1 1 0 0
3	0 1 1 1 0	8	1 0 1 1 1	13	1 1 1 0 1
4	0 1 1 1 1	9	1 1 0 0	14	1 1 1 1

При подальшому перетворенні біноміальні числа доповнюються нулями або одиницями таким чином, щоб в кожній кодовій комбінації було k одиниць.

1.1.4 Постановка задачі проектування. Виконаємо розробку формувачів рівноважних кодів двома способами – виконаємо синтез перетворювача довільних кодів, а також побудуємо формувач рівноважних кодів на основі двійкових біноміальних кодів і порівняємо складність отриманих схем.

1.2 Розробка та обґрунтування алгоритму функціонування та структурної схеми пристрою, що проектується

Розробимо алгоритм функціонування та структурну схему формувача рівноважних кодів з застосуванням у якості проміжної біноміальної системи числення з двійковим алфавітом.

В таблиці 1.2 наведені розглянуті вже нерівномірні біноміальні числа і відповідні їм рівноважні коди.

Таблиця 1.2 – Нерівномірні двійкові біноміальні числа і відповідні рівноважні кодові комбінації

№	Нерівномірне біноміальне число	Рівноважне слово	№	Нерівномірне біноміальне число	Рівноважне слово	№	Нерівномірне біноміальне число	Рівноважне слово
0	00	001111	5	100	100111	10	11010	110101
1	010	010111	6	1010	101011	11	11011	110110
2	0110	011011	7	10110	101101	12	11100	111001
3	01110	011101	8	10111	101110	13	11101	111010
4	01111	011110	9	1100	110011	14	1111	111100

Порівняння нерівномірних двійкових біноміальних чисел і рівноважних кодових комбінацій в кожному рядку таблиці дозволяє зробити наступний висновок: для отримання рівноважного коду необхідно біноміальне число доповнити нулями або одиницями. Якщо отримана біноміальна комбінація містить k одиниць, то доповнити двійкове слово потрібно нулями, причому загальна їх кількість має стати рівним $n-k$. Якщо ж біноміальна комбінація містить $n-k$ нулів, то доповнити її необхідно одиницями таким чином, щоб їх загальна кількість дорівнювала k .

Таким чином, процедура формування рівноважних кодів розбивається на два етапи - отримання нерівномірного біноміального числа, а потім перехід від нього до відповідної рівноважної комбінації. Тому формувач рівноважних кодових комбінацій функціонально буде складатися з двох блоків - перетворювача вихідних двійкових слів в біноміальні кодові слова з двійковим

алфавітом і перетворювача отриманих біноміальних слів в кодові слова з постійною вагою.

Перетворення вихідного двійкового слова в нерівномірне біноміальне базується на методі порозрядного зважування. Початкове двійкове (що перетворюється) число порівнюється з вагою спочатку старшого розряду. Якщо дана вага вміщується в числі, то відповідний розряд буде дорівнювати одиниці, число, що переводиться відповідно необхідно зменшити на величину ваги даного розряду. Подальшому аналізу буде піддаватися решта числа (різниця). Якщо ж вага розряду в числі не вміщується, то розряду присвоюється значення нуля. Величина числа при цьому не змінюється. Наступний крок перетворення полягає в порівнянні з вагою наступного розряду частини вихідного числа, що залишилася після перетворення. Аналогічно розряду присвоюється значення нуля або одиниця. Процедура повторюється до тих пір, поки не будуть отримані значення всіх розрядів числа. При цьому повинні виконуватися наступні умови:

- різниця вихідного числа і суми значень ваг всіх значущих розрядів повинна бути рівною нулю,

- обов'язково повинна виконуватися одна з систем обмежень біноміальної системи числення, що застосовується з вибраними або заданими параметрами.

Алгоритм перетворення двійкового числа в біноміальну систему числення розпишемо детально по кроках:

1) визначається перший коефіцієнт біноміального числа a_{j-1} шляхом порівняння двійкового числа A_i , що переводиться з вагою $(j-1)$ -го (старшого) розряду $C_{n-1}^{k-q_j}$.

Якщо $A_i \geq C_{n-1}^{k-q_j}$, то $a_{j-1} = 1$, підраховується поточне значення кількості одиниць в біноміальному числі, що отримуємо $q_{j-1} = q_j + 1$.

В іншому випадку $a_{j-1} = 0$, $q_{j-1} = q_j$;

2) визначається залишок R , що дорівнює $A_i - C_{n-1}^{k-q_j}$, якщо $a_{j-1} = 1$, або $R = A_i$ при $a_{j-1} = 0$.

3) порівнюється залишок R з вагою наступного $(j-2)$ -го розряду $C_{n-2}^{k-q_j-1}$.

Якщо $R \geq C_{n-2}^{k-q_j-1}$, то $a_{j-2} = 1$, $q_{j-2} = q_{j-1} + 1$.

В іншому випадку $a_{j-2} = 0$, $q_{j-2} = q_{j-1}$.

4) визначаються коефіцієнти a_{j-3} , a_{j-4} ... аналогічно до тих пір, поки кодова комбінація не стане задовольняти одну з систем обмежень.

Коефіцієнт a_l , при якому для кодового слова, що формується, будуть виконуватися зазначені обмеження, є коефіцієнтом a_0 .

Таким чином, виконується перший етап – отримання біноміального числа.

Для отримання коду з постійною вагою до біноміального кодового слова справа дописують нулі, якщо в ньому міститься k одиниць, або одиниці, якщо в ньому міститься $(n - k)$ нулів, до тих пір, поки розрядність коду не стане рівною n .

Розробимо пристрій, що виконує перетворення двійкових кодових комбінацій в кодові комбінації, що характеризують біноміальну систему числення з двійковим алфавітом.

Структурна схема перетворювача кодових двійкових комбінацій в біноміальний код, що реалізує описаний алгоритм, може мати вигляд, наведений на рисунку 1.3.

Перетворювач кодів містить наступні основні вузли:

- 1 – реєстр поточної вибірки;
- 2 – лічильник довжини слів;
- 3 – лічильник одиниць;
- 4 – лічильник нулів;
- 5 – формувач кодів;
- 6 – блок синхронізації;
- 7 – суматор;
- 8 – блок порівняння;
- 9 – реєстр числа;
- 10 – реєстр маски значущості;
- 11 – схема перевірки;
- 12 – блок індикації.

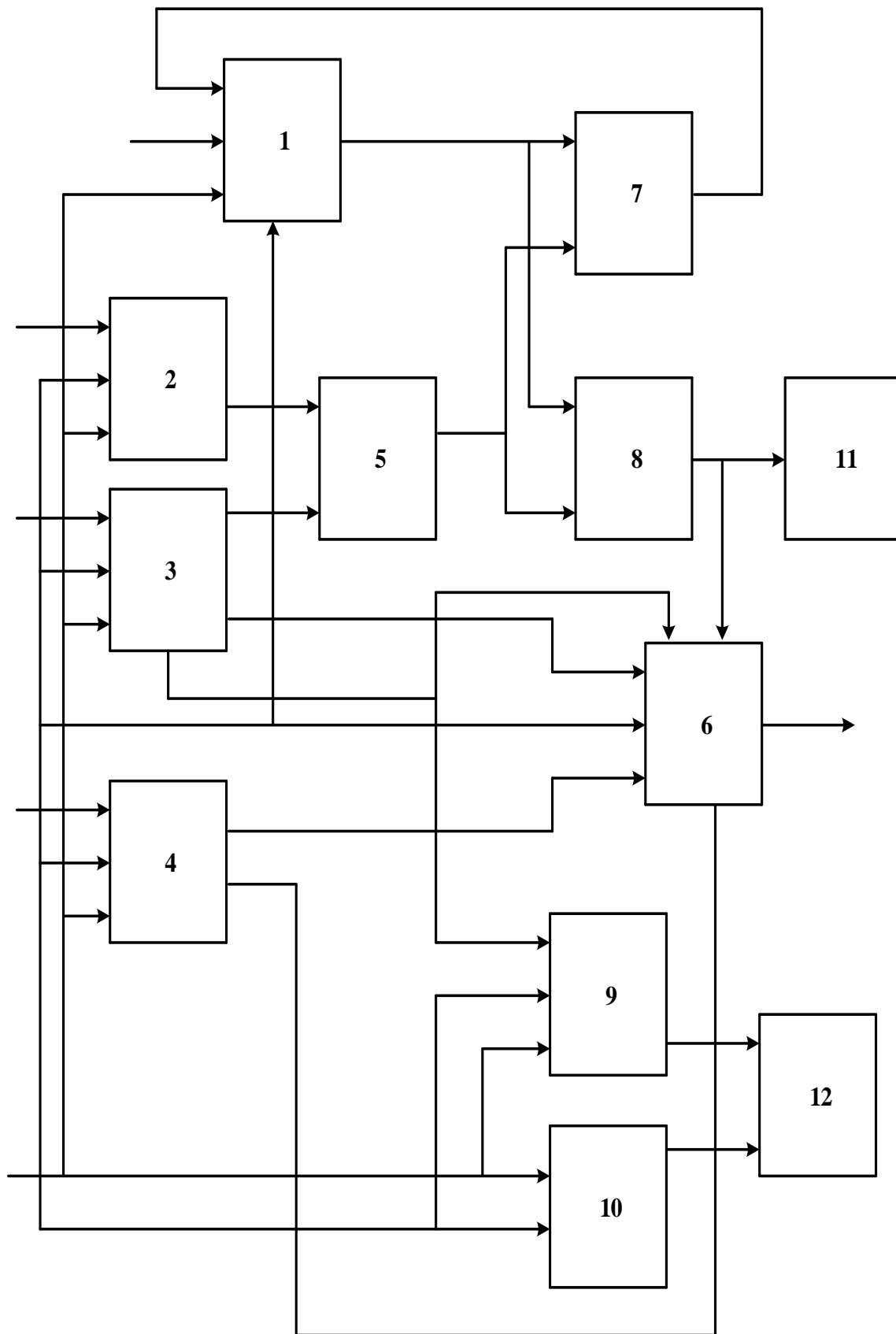


Рисунок 1.3 – Структурна схема формувача рівноважних кодових комбінацій

Вихідне двійкове число в паралельному виді записується в реєстр поточної вибірки 1.

Переведення двійкового вихідного числа використовує принцип порозрядного зваження. Вага кожного розряду біноміального числа визначається величиною сполучення $C_{n-1-j}^{k-q_j}$, де q_j - число одиниць в попередніх розрядах; $j = 0, 1, 2, \dots, n - 1$. Сполучення формується формувачем кодів 5, який при подачі на одну групу входів значення n , а на іншу значення k формує на виході значення C_n^k .

Для підрахунку числа розрядів $n - 1 - j$, числа одиниць и числа нулів ($k - q_j$), ($n - k$) використовуються лічильники довжини слова 2, одиниць 3, нулів 4.

Порівнюють слово, що переводиться з вагою поточного розряду. Цю функцію виконує блок порівняння 8, який порівнює кодові комбінації на двох групах входів та видає сигнали «більше або дорівнює» або «менше». Визначають різницю між цими числами. Для цього застосовують суматор 7. При додатному значенні різниці або при рівності цього числа розряду призначають значення "1", при від'ємному - "0".

Визначають залишок отриманої різниці, який при додатному значенні дорівнює абсолютній величині цієї різниці, а при від'ємному – вихідному значенню числа, що переводиться і порівнюють з вагою наступного розряду. Цю процедуру виконують до тих пір, поки кодова комбінація не стане задовольняти умовам біноміального числа, тобто вміст лічильника нулів або лічильника одиниць повинен стати рівним нулю.

Для отримання коду з постійною вагою до біноміального коду справа дописують нулі, якщо в ньому міститься k одиниць, або одиниці, якщо в ньому міститься $(n - k)$ нулів, до тих пір, поки розрядність коду не стане рівною n . Вихідні кодові комбінації накопичуються в реєстрі числа 9 та реєстрі маски значущості 10.

Після перетворення рівноважних кодових комбінацій цей код надходить в блок індикації 12.

Перетворювач кодів перед початком перетворення проводить перевірку, чи входить число, що переводиться до діапазону даної системи числення з параметрами n та k .

Алгоритм функціонування перетворювача кодів представлено на рисунку 1.4.

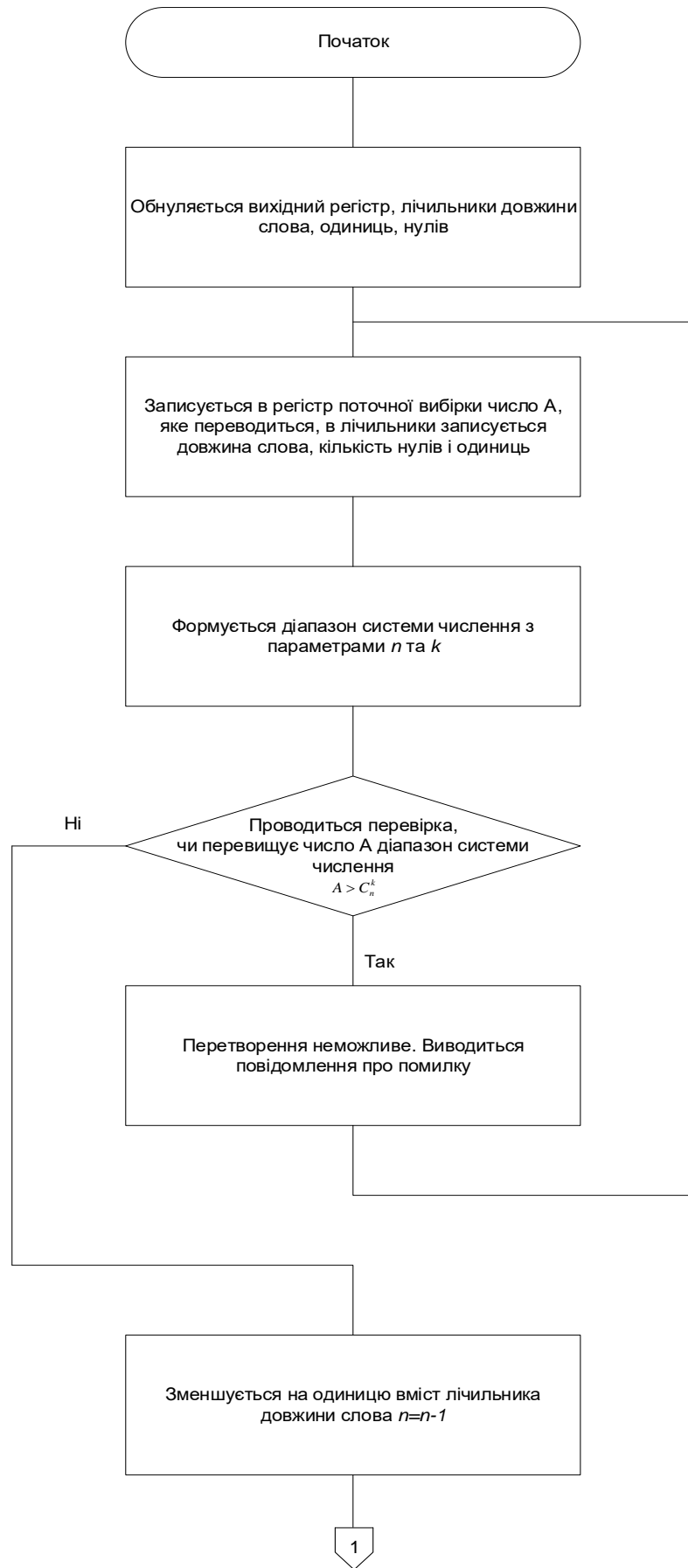
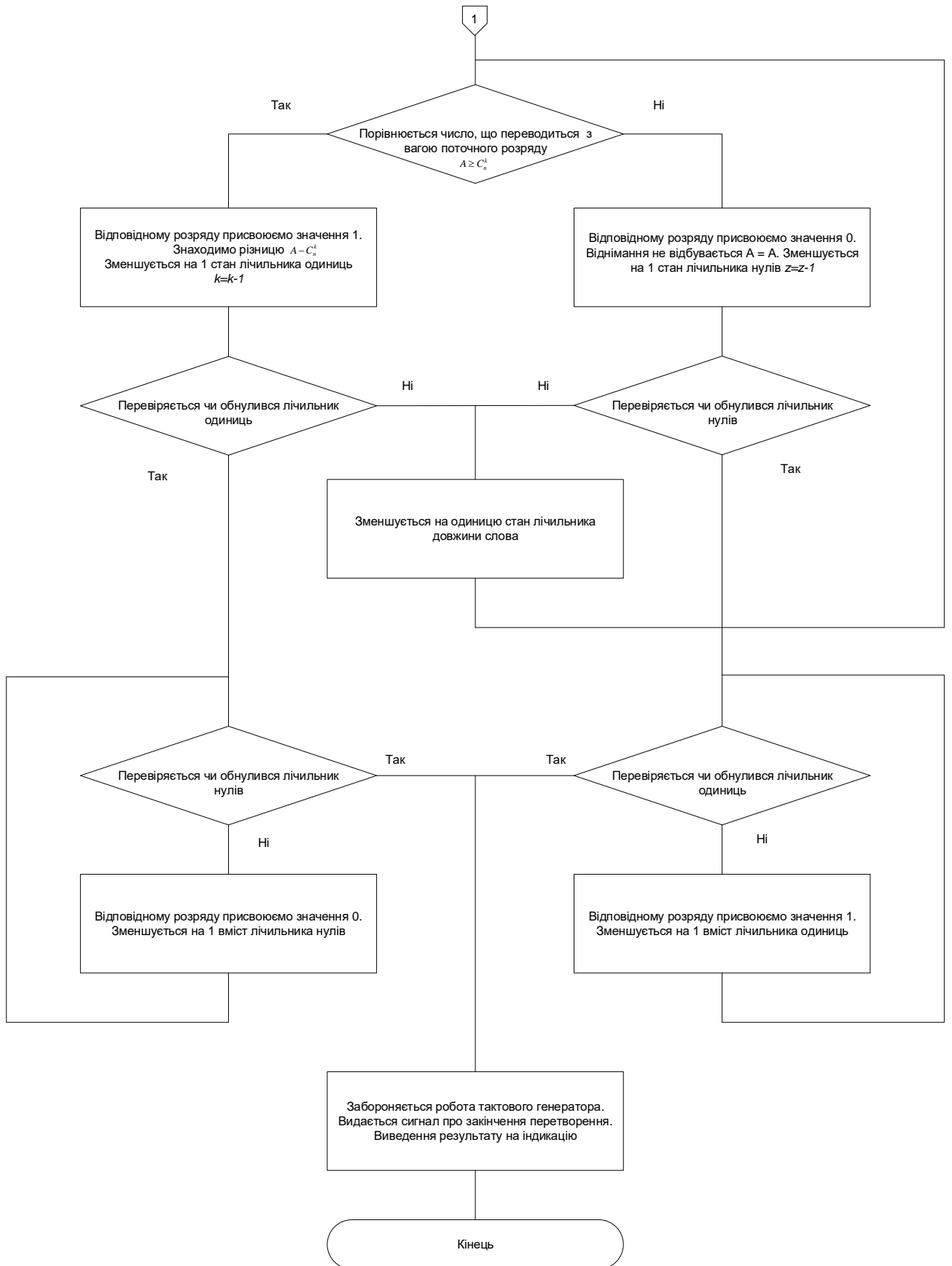


Рисунок 1.4 - Алгоритм функціонування перетворювача кодів



Продовження рисунку 1.4 - Алгоритм функціонування перетворювача кодів

На наступному етапі проведемо розробку функціональної схеми перетворювача двійкових слів у рівноважні кодові комбінації з проміжним перетворенням в двійкові біноміальні коди та розглянемо варіанти побудови формувача рівноважних кодів в якості перетворювача довільних кодів.

1.3 Розробка функціональної схеми генератора рівноважних кодових комбінацій

1.3.1 Перетворювач двійкових кодів в рівноважні кодові комбінації.

Перетворювачі довільних кодів призначені для перетворення чисел з однієї форми подання в іншу. Відправним пунктом для побудови перетворювача коди є таблиця відповідності, в якій записується повний набір вхідних і відповідний набір вихідних кодових слів. Якщо вхідні і вихідні слова записані двійковими символами, то синтез перетворювача коду зводиться до знаходження для кожного розряду вихідного слова булевої функції, що встановлює зв'язок даного розряду з вхідними наборами двійкових змінних. Знаходження такого зв'язку і мінімізація булевого виразу здійснюється за допомогою карт Карно (діаграм Вейча). На заключному етапі отримана функція перетворюється до виду, зручного для реалізації функцій в заданому (обраному) функціональному базисі.

До побудови кодового перетворювача можна підійти з декількох позицій. При першому підході перетворювач реалізується як система булевих функцій групи аргументів. Найпростішим способом побудови схеми, що обробляє систему функцій з m -виходами, є синтез звичайними методами m -невдомих одновихідних функцій. Більш економічне рішення зазвичай можна отримати при підході до системи функцій з урахуванням її можливості. Тоді часто вдається виявити загальні логічні фрагменти, що входять до формули декількох виходів. Ці фрагменти досить реалізувати схемно лише один раз.

При другому підході до побудови кодового перетворювача він трактується як пара декодер-кодер (див. рис. 1.5). Перетворювач коду в даному випадку реалізує таблицю істинності, представлену в таблиці 1.3.

Число входів дешифратора дорівнює числу входів перетворювача, число виходів шифратора - числу виходів перетворювача. З'єднання дешифратора і шифратора виконуються відповідно до таблиці відповідностей. Частина виходів декодера і входів кодера може не використовуватися.

Таблиця 1.3 – Таблиця істинності для перетворювача кодів

A		B		
a_2	a_1	Z	J	K
0	0	1	0	0
0	1	0	0	1
1	0	0	1	1
1	1	0	0	0

Якщо декільком вхідним комбінаціям відповідає одна і та ж вихідна, то відповідні виходи декодера об'єднують на елементі АБО і вихід останнього подають на потрібний вхід кодера.

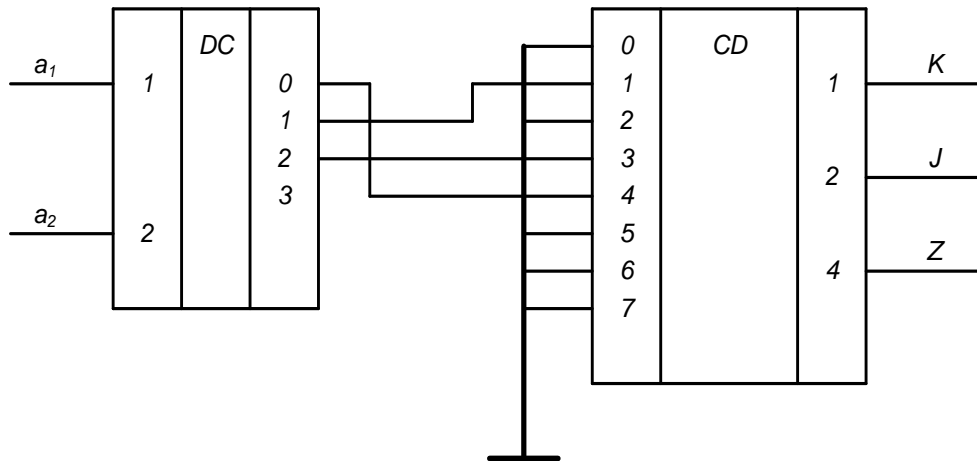


Рисунок 1.5 – Кодовий перетворювач, побудований як структура декодер-кодер

При третьому підході перетворювач коду можна реалізувати на програмованих постійних запам'ятовуючих пристроях (ППЗП) або програмованих логічних матрицях (ПЛМ). ПЛМ, як і ППЗП, належать до програмованих інтегральних схем двоступеневої структури, що складається з двох послідовних матриць «матриця І» - «матриця АБО». ПЛМ відрізняється від ППЗП тим, що в ППЗП матриця І жорстка, а матриця АБО - програмована, а в ПЛМ обидві матриці І та АБО – програмовані. Друга відмінність полягає в тому, що на ПЛМ можна реалізувати не будь-яку систему перемикальних функцій, як на ППЗП, а лише задовольняє додатковому обмеженню: довжина диз'юнктивних нормальних форм відтворюваних функцій не повинна перевищувати числа

перехідних ланцюгів між матрицями І та АБО. Структура матриць І та АБО складається з горизонтальних і вертикальних шин, в вузлах перетину яких знаходяться елементи зв'язку, які при програмуванні вводяться або усуваються (див. рис. 3). Як елементи зв'язку можуть служити наприклад, діоди в матриці І та транзистори в матриці АБО, або плавкі перемички, які при програмуванні перепалюються шляхом подачі високої напруги. Слід зазначити, що як ПЗП, так і ПЛМ реалізує найбільш розгорнуті форми представлення функцій (ДДНФ) і для них не використовується мінімізація функцій.

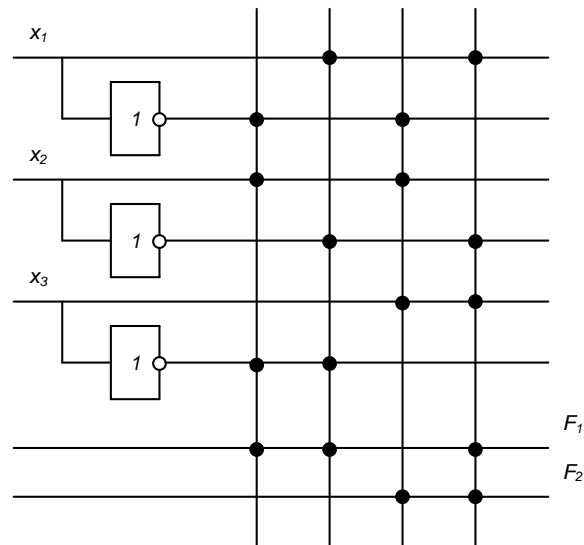


Рисунок 1.6 – Кодовий перетворювач, реалізований з використанням ПЛМ

Використовувати ПЗП для побудови перетворювачів потрібно тоді, коли перетворенню підлягають всі або майже всі комбінації вхідних змінних, а загальне число змінних більше 6-8. Якщо в заданій для реалізації функції використовується лише порівняно мала частка всіх можливих вхідних комбінацій, то раціональніше застосувати ПЛМ. Через значно меншу кількість елементів у корпусі ПЛМ вони зазвичай дешевше, мають меншу затримку та споживають меншу потужність, ніж ПЗП.

Що стосується вибору між ПЛМ та набором окремих логічних мікросхем для побудови кодових перетворювачів, то, починаючи вже з вельми простих перетворювачів, ПЛМ виявляються вигіднішими і за часом, і за витратами апаратури. Слід, однак, мати на увазі, що ПЛМ має явну перевагу перед розсипом перед реалізацією саме систем функцій, а в разі одновихідних функцій лише коли джерела аргументів і споживачів функцій розташовані компактно. Якщо ж однієї

логічної матрицею замінюється безліч розкиданих по платі окремих логічних елементів, то результуючий виграш за габаритними характеристиками стає вже сумнівним. Справа в тому, що в більшості технологій монтажу площа, зайнята зв'язками, перевищує площу, зайняту власне логічними елементами. Тому тенденція концентрувати логічні операції в одному великому елементі, а не виконувати їх на місцях, в середньому збільшує зайняту площу плати. Програшною по числу мікросхем (правда, виграшною з точки зору уніфікації) є спроба заміни програмованими матрицями спеціалізованих мікросхем типу декодерів, мультиплексорів і т.п. У спеціалізованій мікросхемі раціонально використовуються всі виходи корпусу, а при вкладенні в ПЛМ різних схем конкретних вузлів практично завжди частина виводів буде не використана. ПЛМ призначена для спрощення реалізації довільних нестандартних функцій, та, як всякий універсальний інструмент вона програє спеціалізованим мікросхемам, кожна з яких спеціально спроектована для економної реалізації саме її функцій.

При проектуванні з готових мікросхем (якщо порівнювати структуру кодер-декодер та розсип логічних елементів) більш вигідним та по числу корпусів, та за швидкодією зазвичай виявляється структура декодер-кодер. Однак споживана потужність в цьому випадку може бути більше, ніж у схеми з окремих логічних елементів. Але в той же час витрати часу інженера на логічне проектування за схемою декодер-кодер незмірно менше, ніж витрати на проектування перетворювача з розсипу.

1.3.2 Розробка функціональної схеми формувача рівноважних кодових комбінацій з застосуванням біноміальних кодів. Розглянемо детально роботу формувача рівноважних кодових комбінацій.

Перед початком перетворення сигнал СКИДАННЯ встановлює в нульовий стан лічильник довжини слова, лічильник одиниць, лічильник нулів, а також регістр поточної вибірки, регістр біноміального числа. Сигнал СКИДАННЯ одночасно скидає керуючі тригери в нульовий стан, щоб підготувати його до керування процесом перетворення числа.

По сигналу ПУСК двійковий код, що переводиться, надходить на входи регістра поточної вибірки, на установчі входи лічильника довжини слова, лічильника одиниць та лічильника нулів надходять відповідно довжина рівноважного коду n , кількість у ньому одиниць k та нулів $n - k$.

Цей же сигнал ПУСК надходить на вхід S установки керуючого тригера в одиничний стан, переводячи його тим самим в одиничний стан. При цьому на

прямому виході керуючого тригера формується сигнал логічної одиниці. Цей сигнал на прямому виході тригера буде присутнім там до тих пір, поки повинен тривати процес перетворення. Цей сигнал надходить на вхід схеми збігу – логічний елемент «І», на другий вхід якого подаються тактові імпульси. На третій вхід схеми збігу подається сигнал з інверсного виходу керуючого тригера 2. Так як тригер 2 знаходиться у нульовому стані, то на його інверсному виході знаходиться сигнал логічної одиниці. Логічні одиниці з прямих виходів керуючих тригерів дозволяють проходження тактових імпульсів на схему перетворення.

Схема готова до перетворення заданого двійкового числа в нерівномірне біноміальне з двійковим алфавітом.

Процедура перетворення складається з двох етапів – формування біноміального числа та формування рівноважного слова. Схема роботи керуючого тригера 3 наведена на рисунку 1.7. Ілюструє цей процес часова діаграма на рисунку 1.8.

Керуючий імпульс з виходу «менше» схеми збігу надходить на вхід установки тригера 3 в одиничний стан. На прямому виході тригера формується рівень логічної одиниці. Поки тригер 3 буде знаходитися в одиничному стані, то буде виконувати перетворення двійкового числа в біноміальне нерівномірне слово. Як тільки в біноміальному числі з'явиться k – та одиниця або $n - k$ – тий нуль, то процес формування біноміального числа завершено. Свідченням завершення перетворення є поява сигналу логічної одиниці на вході скидання тригера 3. Цей сигнал надходить з виходів «дорівнює» схем порівняння. Схеми порівняння постійно проводять перевірку лічильників нулів та одиниць на рівність їх вмісту одиниці. На одну групу входів схем порівняння подається вміст лічильників нулів та одиниць, а на іншу – значення логічного нуля. Як тільки один із лічильників досягне нульового значення, на виході «дорівнює» схеми порівняння з'являється сигнал логічної одиниці, який надходить через елемент «АБО» на вхід скидання тригера 3 в нульовий стан, і скидає його в нуль. Нульовий сигнал з прямого виходу тригера 3 забороняє проходження імпульсів від кільцевого лічильника. Процес формування біноміального числа на цьому завершено. З інверсного виходу тригера 3 логічна одиниця дозволяє процес додавання необхідної кількості нулів або одиниць.

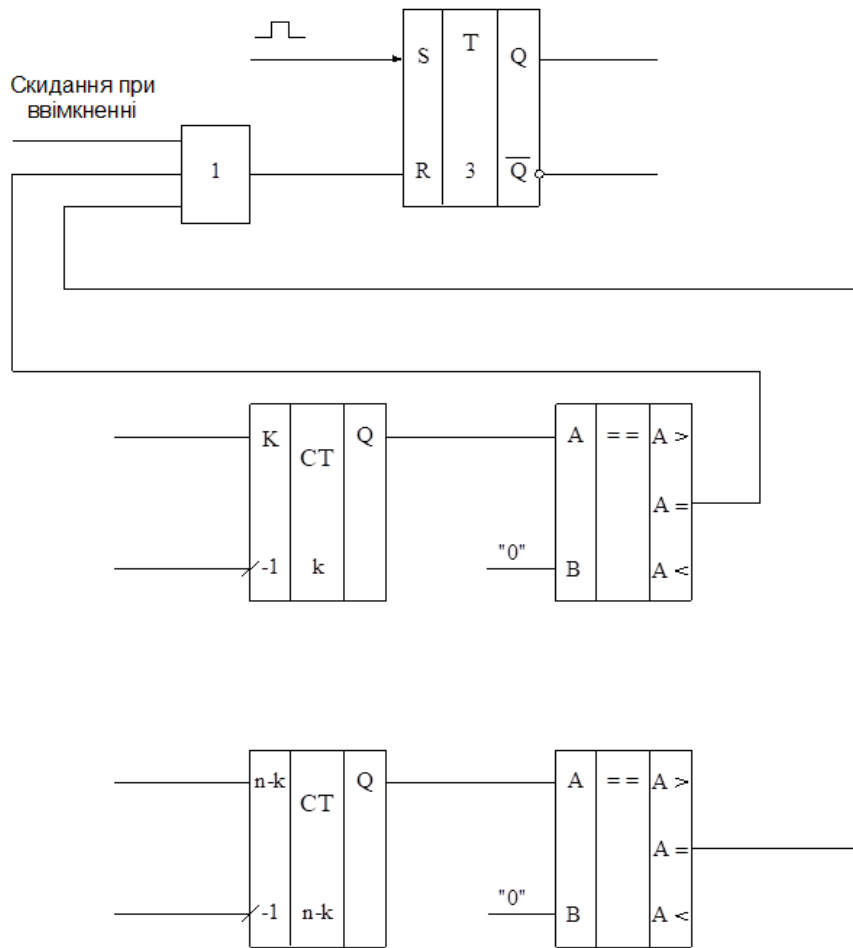


Рисунок 1.7 - Схема роботи керуючого тригера 3

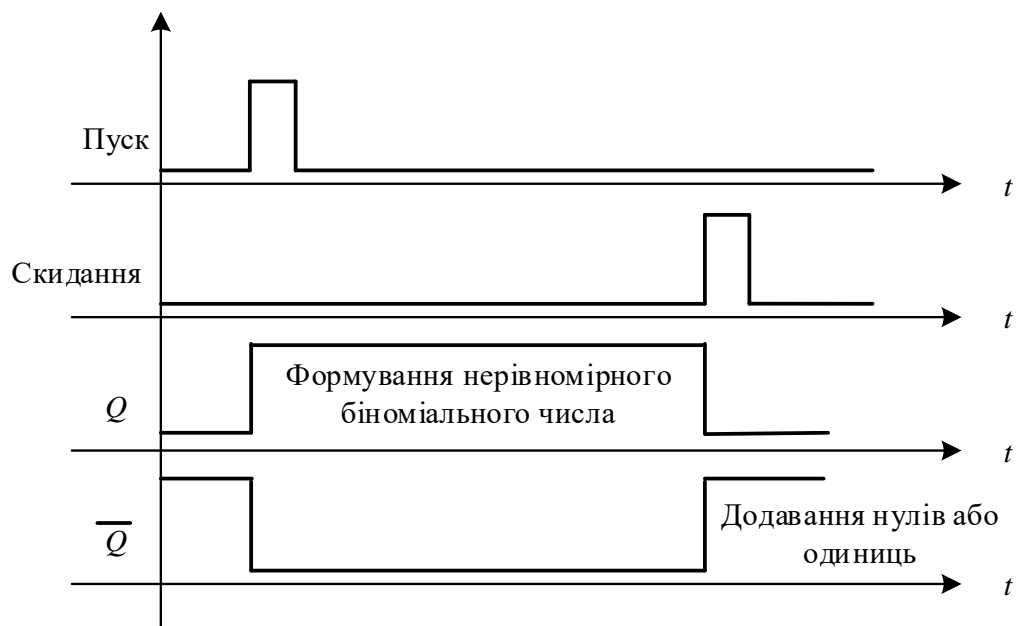


Рисунок 1.8 – Часова діаграма формування рівноважних слів

Проілюструємо процес отримання нерівномірних біноміальних чисел за допомогою прикладу. Виконаємо переведення десяткового числа 6 в біноміальну систему числення з параметрами $n = 6$ та $k = 4$.

Перевіримо можливість перетворення. Підрахуємо діапазон системи числення: $C_n^k = C_6^4 = \frac{6!}{4!*2!} = 15$. Так як 6 менше 15, то переведення можливе.

Виконаємо його. Визначення починається з знаходження старшого розряду

біноміального числа: $C_{n-1}^k = C_5^4 = \frac{5!}{4!*1!} = 5 < 6$. Тому старший розряд дорівнює

одиниці: $a_{j-1} = 1$. Після кожного кроку перетворення проводиться перевірка

виконання вимог систем обмежень біноміальної системи числення, тобто підраховується кількість нулів та одиниць в отриманому біноміальному числі.

Після першого кроку маємо одну одиницю, тому отримане число ще не є біноміальним для заданих параметрів. Визначається залишок: $R = 6 - 5 = 1$.

Переходимо до знаходження наступного розряду біноміального числа. Значення

n та k зменшуються на одиницю. $C_{n-2}^{k-1} = C_4^3 = \frac{4!}{3!*1!} = 4 > 1$. Розряд дорівнює нулю,

так як можлива вага розряду не вміщується в різниці, що залишилася: $a_{j-2} = 0$.

Залишок як і раніше дорівнює одиниці. Тепер у числі одна одиниця та один нуль, тому число поки що не є біноміальним. Переходимо до визначення наступного

розряду. Тепер треба зменшити на одиницю лише n . $C_{n-3}^{k-1} = C_3^3 = \frac{3!}{3!*0!} = 1 = 1$.

Розряд числа дорівнює 1: $a_{j-3} = 1$. Визначаємо залишок: $R = 1 - 1 = 0$. В отриманому

числі дві одиниці та один нуль, тому воно ще не є біноміальним. Визначаємо значення наступного розряду. Зменшуємо на одиницю значення обох параметрів

системи числення. $C_{n-4}^{k-2} = C_2^2 = \frac{2!}{2!*0!} = 1 > 0$. Розряд числа дорівнює нулю: $a_{j-4} = 0$

тепер в отриманому числі дві одиниці та два нуля, тобто для заданих параметрів число є біноміальним. Виконалась умова рівності кількості нулів заданому

значенню: $n - k = 6 - 4 = 2 = 2$. Розглянемо реалізацію даного алгоритму в схемі.

Перетворення можна проілюструвати за допомогою функціональної схеми (додаток).

Формування нерівномірного біноміального числа починається з формування числа сполучень C_n^k , тобто на блок формування сполучень з прямих

виходів лічильника довжини слова та лічильника одиниць надходять записані початкові значення – на першому кроці $n = n - 1$, а $k = k$. На виході формувача сполучень з'являється відповідний сигнал і надходить на другу групу входів суматора та компаратора. На першу групу входів цих блоків надходить вихідний код двійкового числа. В суматорі, який в даному випадку організований як пристрій віднімання, проводиться операція віднімання. Від значення вихідного числа віднімається вага розряду, що аналізується. На виході суматора з'являється значення різниці, яке надходить через елемент «АБО» на вхід регістра поточної вибірки. Одночасно в компараторі проходить порівняння значення записаного в регістрі поточної вибірки зі значенням старшого розряду біноміального числа з заданими параметрами. Якщо число, що переводиться буде більше ваги старшого розряду, то на виході «>» компаратора з'явиться сигнал логічної одиниці. Це значить, що вага біноміального числа, що аналізується вміщується в вихідному двійковому числі, отже, відповідний розряд нового числа дорівнює одиниці. Це значення необхідно записати в регістр біноміального числа. Сигнал логічної одиниці, який через схему «АБО» проходить на одиничний вхід четвертого керуючого асинхронного *RS* - тригера, встановлює його в одиничний стан. На прямому виході тригера з'являється одиничний сигнал, який дозволяє проходження синхроімпульса від кільцевого лічильника через верхній елемент «І». Логічна одиниця значення розряду з'являється на послідовному вході регістра біноміального числа. Тактовий сигнал по входу *C* записує цю одиницю у вихідний регістр. Одиниця з виходу верхньої схеми «І» надходить на тактуючий вхід регістра поточної вибірки і дозволяє в регістр запис нового значення – різниці, що залишилася. Ця ж одиниця надходить на віднімаючий вхід лічильника одиниць, зменшуючи його вміст на одиницю. Цей же сигнал, проходячи через елемент «АБО» надходить і на віднімаючий вхід лічильника довжини слова, зменшуючи таким чином його вміст на одиницю. Якщо ж вага старшого розряду дорівнює числу, що переводиться, то все проходить аналогічно за виключенням того, що відповідний сигнал логічної одиниці з'являється на виході компаратора «=». Через елемент «АБО» цей сигнал аналогічно надходить на одиничний вхід *RS* - тригера. Подальше проходження сигналу аналогічно описаному. Якщо ж вага старшого розряду біноміального числа виявилась більшою, то на виході компаратора «<» з'являється сигнал логічної одиниці, який надходить на вхід скидання тригера та активізує нижню схему «І». Верхня схема «І» у цьому випадку буде закрыта, так як на прямому

виході тригера з'явиться нуль. З виходу верхньої схеми «І» цей нуль надійде на вхід D регістра числа і по тактовому імпульсу запишеться в нього. Нульовий сигнал з виходу верхньої схеми «І» як і раніше надходячи на вхід C регістра поточної вибірки на цей раз не змінить його стану, тобто різниця $A-B$, отримана на виході віднімача тепер не буде записана в регістр. Нульовий сигнал, що надходить на віднімаючий вхід лічильника одиниць, також не змінить його стану. Але логічна одиниця з нижньої схеми «І» надходить на віднімаючий вхід лічильника нулів, зменшуючи його вміст на одиницю, а також через елемент «АБО» надходить на лічильник довжини слова та зменшує і його вміст на одиницю. Таким чином, проведено порівняння числа, що переводиться з вагою старшого розряду.

На наступному кроці на вхід формувача сполучень надходять змінені значення з виходів лічильників довжини слова та кількості одиниць. При чому вміст лічильника довжини слова обов'язково зменшується на одиницю з кожним тактом перетворення. Вміст лічильника одиниць зменшується лише у тому випадку, якщо вага розряду, що аналізується менше числа, записаного на даному кроці в регістр поточної вибірки. Тобто вміст лічильника одиниць зменшується тільки тоді, коли відповідний розряд нового – біноміального числа дорівнює одиниці. Якщо отриманий розряд біноміального числа дорівнює нулю, то зменшується на одиницю вміст лічильника нулів.

Порівняння вмісту регістру поточної вибірки з вагою наступного розряду біноміального числа виконується аналогічно. Процес перетворення буде тривати поки не буде отримано біноміальне число.

Запис значень біноміального числа у вихідний регістр проводиться по тактовому імпульсу кільцевого лічильника. Значення біноміального числа надходять на вхід D вихідного регістру, проходячи через верхню схему «І» та далі через елемент «АБО». Так як в процесі формування біноміального числа тригер 3 знаходиться в одиничному стані, то на його прямому виході присутня одиниця, яка дозволяє проходження через елемент «І» отриманого на прямому виході тригера 4 значення відповідного розряду біноміального числа.

На кожному кроці перетворення проходить аналіз вмісту лічильників нулів та одиниць. Як тільки встановиться в нуль один з цих лічильників, то схема виявлення нульового стану, сформувавши на своєму виході логічну одиницю, скине керуючий тригер 3 у нульовий стан. Це сигнал про закінчення формування біноміального числа. Таким чином, буде виконано перший етап перетворення –

здійснено перехід до біноміального числа. Далі необхідно буде отримати з нього рівноважне слово.

Перехід до рівноважного слова здійснюється при появі рівня логічної одиниці на інверсному виході керуючого тригера 3. При цьому в регістр вихідного числа за тактовими імпульсами, що надходять від кільцевого лічильника необхідно записати відсутні нулі чи одиниці відповідно заданим параметрам біноміальної системи числення. Відповідно до нашого прикладу отримано біноміальне число 1010, а відповідне йому рівноважне – 101011, тобто необхідно доповнити біноміальне число ще двома одиницями. З інверсного виходу тригера 3 на вхід нижньої схеми «І» надходить логічна одиниця. На другий вхід тієї ж схеми надходить сигнал з виходу схеми контролю стану лічильника одиниць. Так як вміст лічильника одиниць не дорівнює нулю, то на виході схеми контролю (вихід «більше») присутня логічна одиниця, яка проходить через елементи «І» та «АБО» і з'являється на вході *D* регістра. Поки лічильник одиниць не встановиться в нуль, на вході *D* буде присутній рівень логічної одиниці. Ця одиниця запишеться у вихідний регістр двічі, поки не встановиться в нуль лічильник одиниць. Якщо в отриманому біноміальному слові були б записані усі необхідні одиниці, і не вистачало б нулів, то на вході вихідного регістру через схеми «І» та «АБО» був би присутній логічний нуль и записаний був би стільки разів якої кількості нулів не вистачало б в біноміальному (рівноважному) числі. Схема керування записом у вихідний регістр наведена на рисунку 1.9.

Схема керування лічильником одиниць в режимі формування рівноважного слова наведена на рисунку 1.10. Вміст лічильника надходить на першу групу входів схеми порівняння. На другу групу входів надходить логічний нуль. Якщо лічильник одиниць не знаходиться у нульовому стані, то на вході компаратора «більше» присутній сигнал логічної одиниці, який надходить на перший вхід схеми збігу, на другому вході присутній сигнал логічної одиниці з інверсного виходу тригера 3 (в режимі формування рівноважного коду). На третій вхід схеми збігу надходять імпульси від кільцевого лічильника.

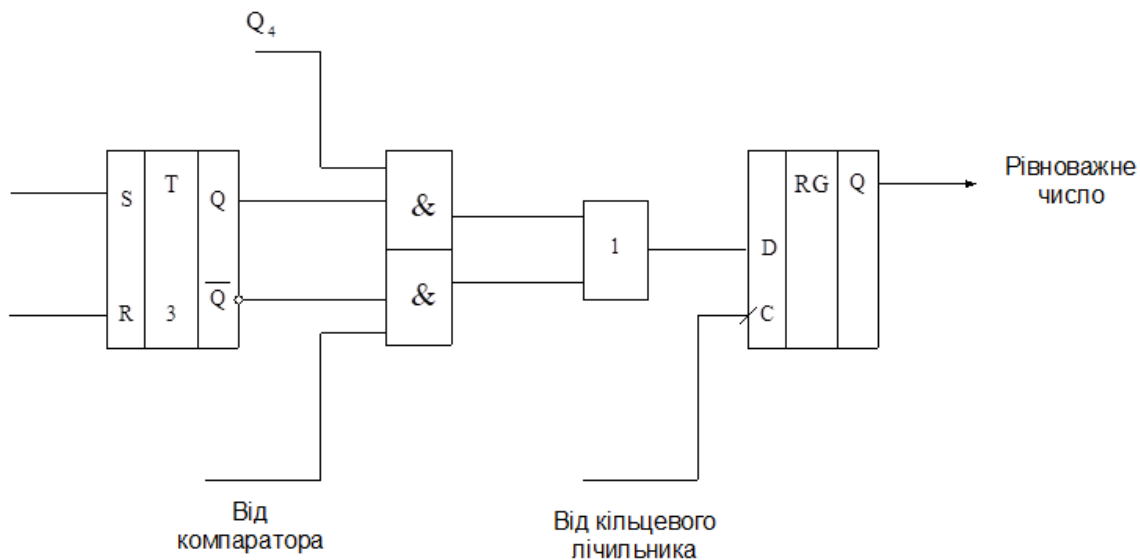


Рисунок 1.9 - Схема керування записом у вихідний регістр

З виходу схеми збігу сформовані імпульси через елемент «АБО» надходять на віднімаючий вхід лічильника одиниць, і з кожним імпульсом від кільцевого лічильника зменшують вміст лічильника одиниць. Таким чином здійснюється зменшення вмісту лічильника одиниць в режимі формування рівноважного слова. В режимі отримання біноміального числа зменшення вмісту лічильника одиниць досягається шляхом подачі керуючих імпульсів на другий вхід схеми «АБО» з виходу формувача біноміальних чисел (рис. 1.10).

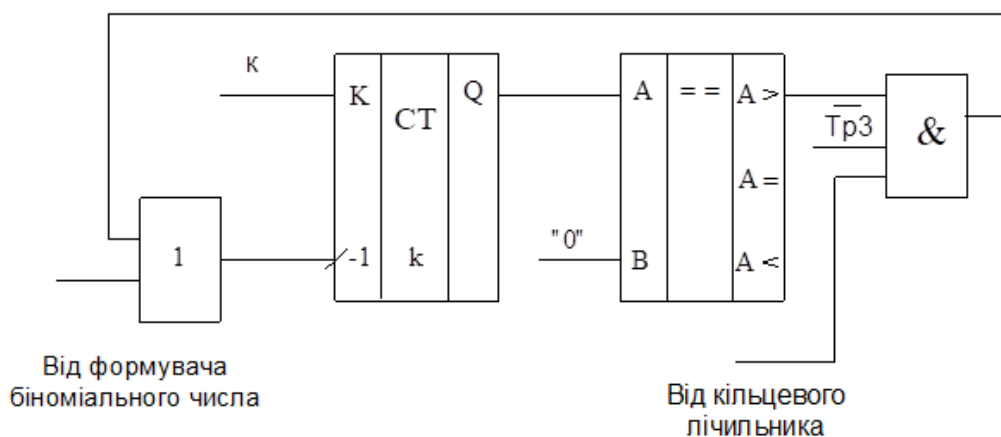


Рисунок 1.10 - Схема керування лічильником одиниць в режимі формування рівноважного слова

Аналогічно проводиться керування лічильником нулів.

При одночасному встановленню в нуль лічильника одиниць та лічильника нулів (тобто отримане слово містить k – одиниць та $n - k$ нулів) відбувається зупинка роботи схеми. Цей стан виділяється схемою «I», на входи якої надходять сигнали з виходів «дорівнює» схем перевірки вмісту лічильників. З виходу схеми сигнал надходить на вхід скидання керуючого тригера 1, на прямому виході якого рівень логічного нуля забороняє проходження імпульсів генератора на вхід формувача. Таким чином, отримано рівноважне слово, яке в послідовному коді зберігається в регістрі числа.

Докладно процес формування рівноважних кодових комбінацій з проміжним одержанням біноміальних кодів можна проаналізувати на основі функціональної схеми, яка наведена в додатку.

Деталізація функціональної схеми приводить до отримання принципової схеми. На наступному кроці виконаємо синтез основних вузлів в двох розглянутих варіантах реалізації генераторів рівноважних кодів.

2 РОЗРОБКА ТА РОЗРАХУНОК ПРИНЦИПОВИХ ЕЛЕКТРИЧНИХ СХЕМ, ВУЗЛІВ ТА БЛОКІВ ПРИСТРОЮ

2.1 Вибір елементної бази

2.1.1 Метою вибору елементної бази є обґрунтування серії (або серій) інтегральних мікросхем, а також інших електрорадіоелементів, необхідних для раціональної реалізації пристрою, що проектується.

Критеріями вибору серії (серій) ІМС є:

- наявність необхідних функціональних вузлів у складі серії ІМС;
- мала споживана потужність;
- виконання вимог по швидкодії (граничній робочій частоті) і умовам експлуатації;
- низька вартість;
- можливість керувати необхідними елементами, наприклад, індикаторами без додаткових підсилень і перетворень вихідних сигналів і т.п.

Вибір елементної бази необхідно проводити в наступній послідовності:

- за функціональною схемою пристрою визначаються необхідні функціональні вузли (лічильники, регістри, шифратори, перетворювачі коду тощо) та їх параметри;
- по довідниках визначаються серії ІМС, що містять всі або частину відповідних функціональних вузлів. При відсутності функціональних вузлів визначається можливість їх побудови за допомогою вхідних до складу серії елементів;
- на основі проведеного аналізу визначається одна або декілька серій, що застосовуються для побудови пристрою.

При виборі дискретних елементів (індикаторів, електромагнітних реле і т.д.), які входять до складу пристрою, що проектується, доцільно використовувати ті, які керуються сигналами з мікросхем або спеціальними мікросхемами сполучення, що входять до складу серій. Інакше проводиться розрахунок схем сполучення на дискретних елементах.

Вибір елементної бази доцільно ілюструвати таблицями, наприклад:

- таблиця відповідності складу серій потрібним функціональним вузлам і можливість реалізації функціональних елементів на дискретних логічних елементах серії;

- таблиця характеристик обраних серій ІМС;
- таблиця характеристик необхідних дискретних елементів.

На підставі аналізу даних таблиць проводиться вибір елементної бази.

2.1.2 Інтегральні мікросхеми (ІМС) серії К1533 призначені для організації швидкого обміну та обробки цифрової інформації, часового та електричного проходження сигналів в обчислювальних системах [1,2]. ІМС серії К1533 дозволяють отримати мінімальне значення добутку швидкодії на потужність розсіяння в порівнянні з відомими серіями ТТЛмікросхем. У наведеній нижче таблиці подані порівняльні характеристики цифрових ІМС різних серій.

Таблиця 2.1 – Порівняльні характеристики цифрових ІМС

Параметр	К155	К134	К531	КР1530	КР1531	К1533
Час затримки, нс	10	33	3	1,5	2,7	4
Потужність споживання, мВт	10	10	20	22	4	1
Фактор якості, пДж	100	33	60	33	10,8	4

ІМС серії К1533 мають функціональні аналоги в інших серіях і співпадають з ними відносно призначення виводів у корпусі. Це дозволяє виконувати повну заміну мікросхем серії К555, К533, К155, КР1531. Зарубіжний аналог – серія мікросхем SN74ALSxxxx фірми Texas Instruments (TI) США. Конструктивно мікросхеми серії К1533 виконані в 14-, 16-, 20- та 24-виводних пластмасових корпусах типу 201, 14-1, 238, 16-1, 2140, 20-8, 2142, 24-2 відповідно. Мікросхеми мають стандартні ТТЛ вхідні/вихідні рівні сигналів. Напрацювання мікросхеми ~50000 годин. Вживані аббревіатури: ТТЛ – транзисторно-транзисторна логіка; Н – високий рівень напруги; L – низький рівень напруги; Z(\tilde{N}) - високоомний стан виходу мікросхеми.

2.2 Вибір та розрахунок принципових схем пристрою, що проектується

2.2.1 Синтез перетворювача кодів. До побудови кодового перетворювача можна підійти з декількох позицій. При першому підході

перетворювач реалізується як система булевих функцій групи аргументів. Найпростішим способом побудови схеми, що обробляє систему функцій з m -виходами, є синтез звичайними методами m -невідомих одновихідних функцій. Розглянемо докладно реалізацію формувача рівноважних кодових комбінацій цим методом. Для побудови комбінаційної схеми перетворювача кодів згідно стандартної методики необхідно скласти таблицю істинності, за якою записуються формули, і в решті-решт будується схема.

Правило функціонування перетворювача кодів наведемо у вигляді таблиці істинності (див. табл. 2.2). В ній наводиться відповідність між вхідними та вихідними кодами. Вхідними кодовими комбінаціями в цьому випадку будуть двійкові кодові комбінації, які необхідно перетворювати, а вихідними, відповідно, рівноважні кодові комбінації.

Таблиця 2.2 – Таблиця істинності перетворювача двійкових кодів у рівноважні

№	x_1	x_2	x_3	x_4	f_1	f_2	f_3	f_4	f_5	f_d
0	0	0	0	0	0	0	1	1	1	1
1	0	0	0	1	0	1	0	1	1	1
2	0	0	1	0	0	1	1	0	1	1
3	0	0	1	1	0	1	1	1	0	1
4	0	1	0	0	0	1	1	1	1	0
5	0	1	0	1	1	0	0	1	1	1
6	0	1	1	0	1	0	1	0	1	1
7	0	1	1	1	1	0	1	1	0	1
8	1	0	0	0	1	0	1	1	1	0
9	1	0	0	1	1	1	0	0	1	1
10	1	0	1	0	1	1	0	1	0	1
11	1	0	1	1	1	1	0	1	1	0
12	1	1	0	0	1	1	1	0	0	1
13	1	1	0	1	1	1	1	0	1	0
14	1	1	1	0	1	1	1	1	0	0
15	1	1	1	1	*	*	*	*	*	*

За таблицею істинності можна записати декілька формул. Зазвичай синтез починають з запису досконалої диз'юнктивної нормальної форми. За ДДНФ

можна побудувати комбінаційну схему, але оскільки ДДНФ – це найбільш загальна формула, відповідно має максимальну довжину, то і схема, побудована за такою формулою буде мати максимальну кількість логічних елементів. Тому необхідно виконати мінімізацію поданої логічної функції.

Однією з основних задач, що виникають при синтезі комбінаційних схем (КС), є мінімізація переключальних функцій, які ці комбінаційні схеми реалізують. Чим простіше буде записано логічний вираз, який описує цю функцію, тим простішою, дешевшою та надійнішою буде комбінаційна схема, що реалізує цю функцію.

Аналітичні методи мінімізації в загальному випадку надзвичайно трудомісткі, тому найбільшого поширення отримав графічний метод мінімізації за допомогою діаграм Вейча, безперечною перевагою якого є наочність та простота використання при невеликій кількості змінних ($n \leq 6$). В якості критерія складності логічного виразу, що описує функцію, доцільно прийняти число первинних термів, що в нього входять. Очевидно, що будь-який метод мінімізації може базуватися лише на тотожних перетвореннях логічних виразів.

Оскільки функція залежить від чотирьох змінних, то застосуємо діаграму Вейча. Діаграма Вейча для мінімізації за одиницями має наступну нумерацію клітинок:

		x_2		$\overline{x_2}$		
		12	13	9	8	$\overline{x_3}$
x_1		14	15	11	10	x_3
$\overline{x_1}$		6	7	3	2	
		4	5	1	0	$\overline{x_3}$
		$\overline{x_4}$	x_4		$\overline{x_4}$	

Мінімальні форми одержуємо, застосувавши операції неповного склеювання та поглинання. На наступному кроці вже можна буде будувати комбінаційну схему, але необхідно вибрати найкращий для даного випадку базис. Базис – це набір елементів, за допомогою яких можна побудувати схему будь-якої складності. Якщо заздалегідь в технічному завданні не заданий необхідний функціональний базис, то необхідно проаналізувати принаймні два найпоширеніших функціональних базиси та вибрати ефективніший в даному випадку. Ефективність при побудові схеми – це найменша кількість елементів в схемі.

Найпоширенішими функціональними базисами на сьогодні є – базис Шефера та базис Пірса. Для реалізації в базисі Шефера необхідно одержувати мінімальну ДНФ, а для – базиса Пірса необхідна мінімальна КНФ.

До одержаних форм необхідно послідовно застосувати закон подвійного заперечення та правило де Моргана. Виконаємо це.

Виконаємо послідовно мінімізацію функцій з таблиці істинності.

На наступному кроці до мінімальних ДНФ застосуємо послідовно закон подвійного заперечення ($\overline{\overline{F}} = F$) та закон де Моргана ($\overline{A \vee B} = \overline{A} \wedge \overline{B}$).

Виконаємо мінімізацію функції f_1 та побудуємо схему:

		x_2				$\overline{x_2}$				
		1	12	1	13	1	9	1	8	$\overline{x_3}$
x_1		1	14	*	15	1	11	1	10	x_3
$\overline{x_1}$		1	6	1	7	0	3	0	2	
		0	4	1	5	0	1	0	0	$\overline{x_3}$
		$\overline{x_4}$		x_4		$\overline{x_4}$				

$$f_1 = x_1 \vee x_2 x_4 \vee x_2 x_3$$

$$f_1 = \overline{\overline{x_1 \vee x_2 x_4 \vee x_2 x_3}} = \overline{\overline{x_1} \wedge \overline{x_2 x_4} \wedge \overline{x_2 x_3}}$$

Побудуємо комбінаційну схему, вона представлена на рисунку 2.1.

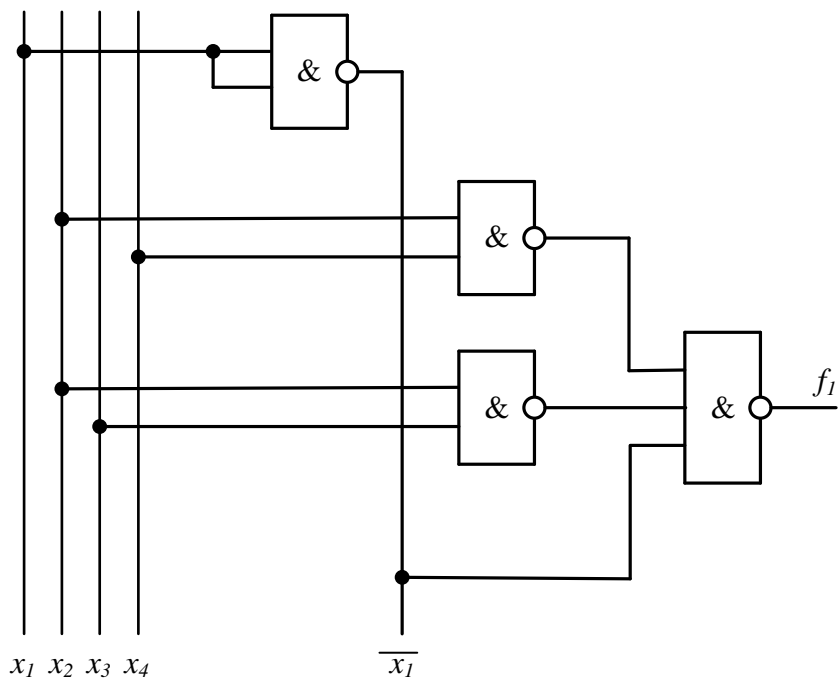


Рисунок 2.1 – Реалізація f_1 в базисі Шеффера

Виконаємо мінімізацію функції f_2 та побудуємо схему:

		x_2		\bar{x}_2		
		1 12	1 13	1 9	0 8	\bar{x}_3
x_1		1 14	* 15	1 11	1 10	x_3
\bar{x}_1		0 6	0 7	1 3	1 2	
		1 4	0 5	1 1	0 0	\bar{x}_3
		\bar{x}_4	x_4	\bar{x}_4	x_4	

$$f_2 = \bar{x}_2 x_3 \vee \bar{x}_2 x_4 \vee x_2 \bar{x}_3 \bar{x}_4 \vee x_1 x_2$$

$$f_2 = \overline{\overline{\bar{x}_2 x_3 \vee \bar{x}_2 x_4 \vee x_2 \bar{x}_3 \bar{x}_4 \vee x_1 x_2}} = \overline{\overline{\bar{x}_2 x_3} \wedge \overline{\bar{x}_2 x_4} \wedge \overline{x_2 \bar{x}_3 \bar{x}_4} \wedge \overline{x_1 x_2}}$$

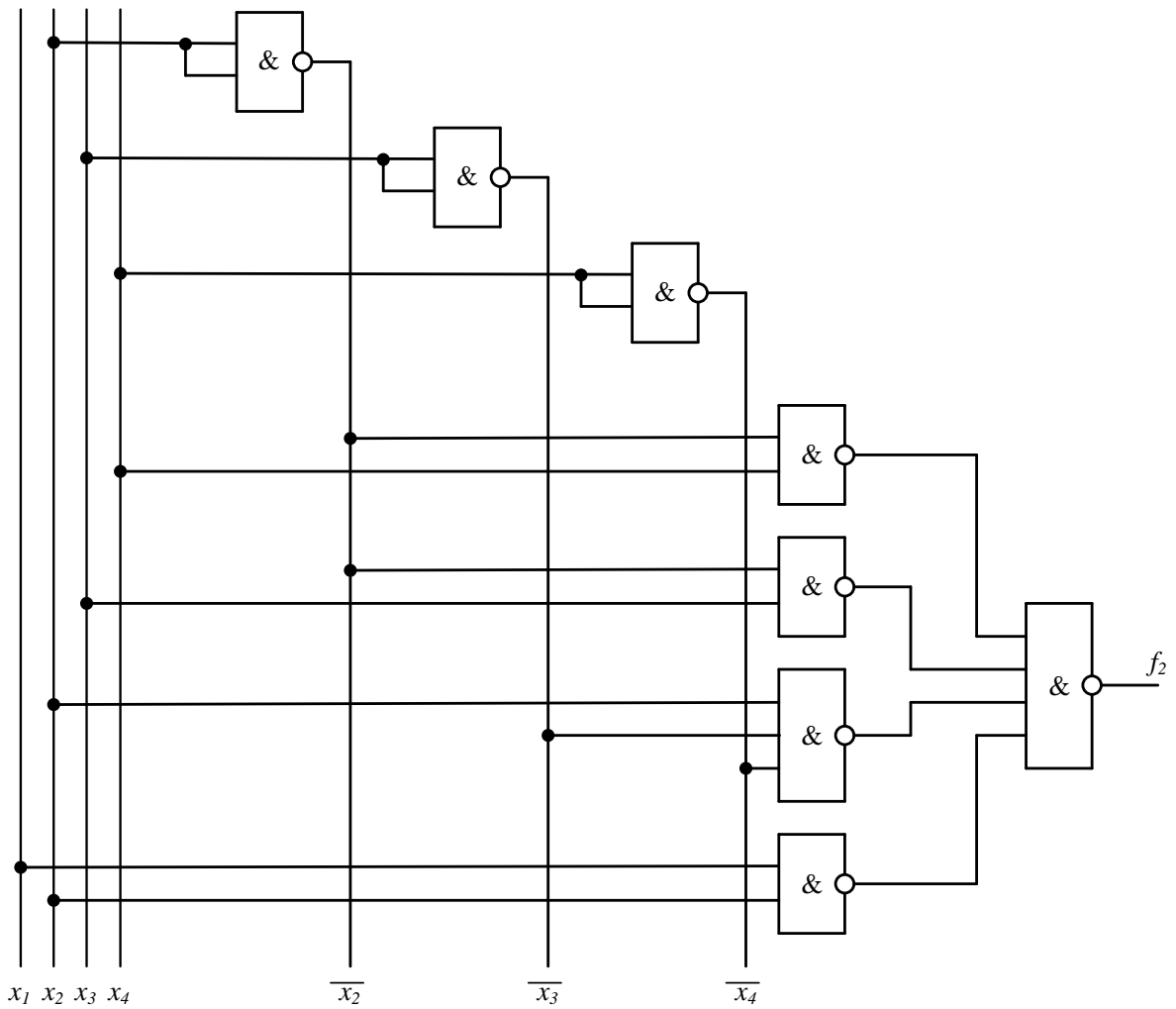


Рисунок 2.2 – Реалізація f_2 в базисі Шеффера

Виконаємо мінімізацію функції f_3 та побудуємо схему

		x_2			\bar{x}_2				
		1	12	1	13	0	9	1	\bar{x}_3
x_1		1	14	*	15	0	11	0	x_3
\bar{x}_1		1	6	1	7	1	3	1	
		1	4	0	5	0	1	1	\bar{x}_3
		\bar{x}_4		x_4		\bar{x}_4			

$$f_3 = x_1x_2 \vee \bar{x}_1x_3 \vee \bar{x}_3\bar{x}_4$$

$$f_3 = \overline{\overline{x_1x_2 \vee \bar{x}_1x_3 \vee \bar{x}_3\bar{x}_4}} = \overline{\overline{x_1x_2} \wedge \overline{\bar{x}_1x_3} \wedge \overline{\bar{x}_3\bar{x}_4}}$$

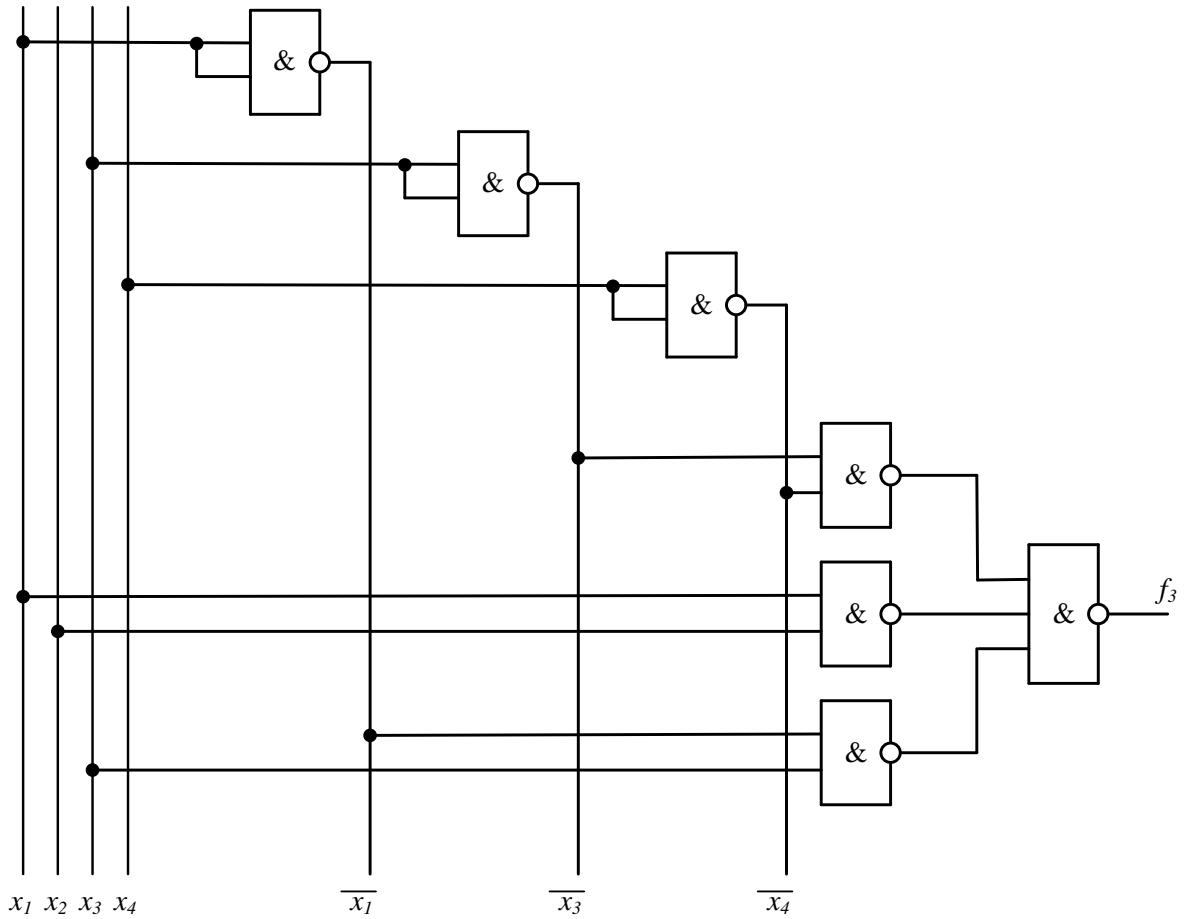


Рисунок 2.3 – Реалізація f_3 в базисі Шеффера

Виконаємо мінімізацію функції f_4 та побудуємо схему:

		x_2		\bar{x}_2			
		0 12	0 13	0 9	1 8		\bar{x}_3
x_1		1 14	* 15	1 11	1 10		x_3
\bar{x}_1		0 6	1 7	1 3	0 2		
		1 4	1 5	1 1	1 0		\bar{x}_3
		\bar{x}_4	x_4		\bar{x}_4		

$$f_4 = x_1 x_3 \vee x_3 x_4 \vee \bar{x}_1 \bar{x}_3 \vee \bar{x}_2 \bar{x}_3 \bar{x}_4$$

$$f_4 = \overline{\overline{x_1 x_3 \vee x_3 x_4 \vee \bar{x}_1 \bar{x}_3 \vee \bar{x}_2 \bar{x}_3 \bar{x}_4}} = \overline{\overline{x_1 x_3} \wedge \overline{\overline{x_3 x_4} \wedge \overline{\overline{\bar{x}_1 \bar{x}_3} \wedge \overline{\overline{\bar{x}_2 \bar{x}_3 \bar{x}_4}}}}}$$

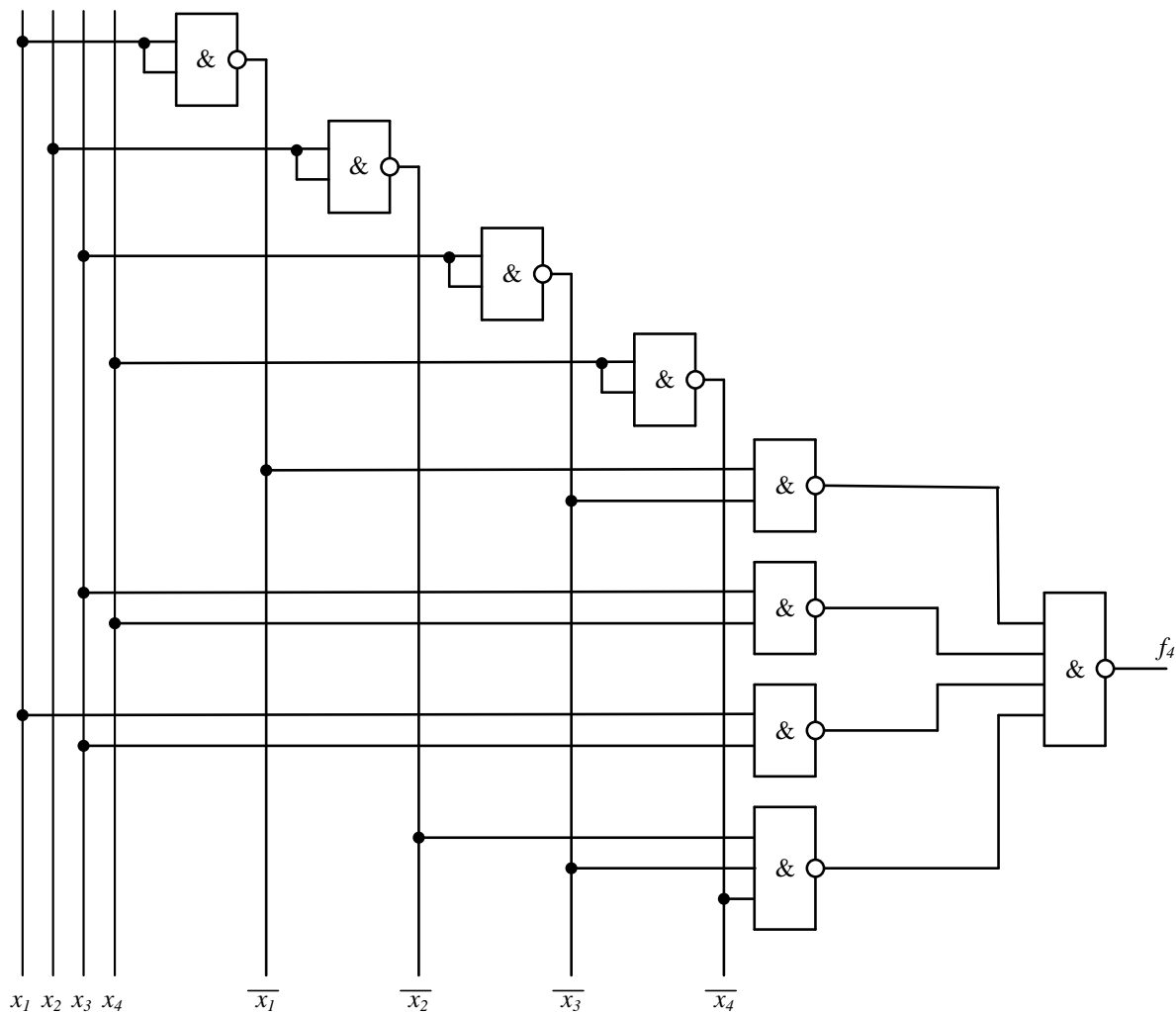


Рисунок 2.4 – Реалізація f_4 в базисі Шеффера

Виконаємо мінімізацію функції f_5 та побудуємо схему:

	x_2		$\overline{x_2}$			
x_1	0 12	1 13	1 9	0 10	$\overline{x_3}$	
$\overline{x_1}$	1 6	0 7	0 3	1 2	x_3	
	1 4	1 5	1 1	1 0	$\overline{x_4}$	
	$\overline{x_4}$		x_4			

$$f_5 = x_1x_4 \vee \overline{x_1}\overline{x_3} \vee \overline{x_1}\overline{x_4} \vee \overline{x_2}\overline{x_3}\overline{x_4}$$

$$f_5 = \overline{\overline{x_1x_4} \vee \overline{\overline{x_1}\overline{x_3}} \vee \overline{\overline{x_1}\overline{x_4}} \vee \overline{\overline{x_2}\overline{x_3}\overline{x_4}}} = \overline{\overline{x_1x_4} \wedge \overline{\overline{x_1}\overline{x_3}} \wedge \overline{\overline{x_1}\overline{x_4}} \wedge \overline{\overline{x_2}\overline{x_3}\overline{x_4}}}$$

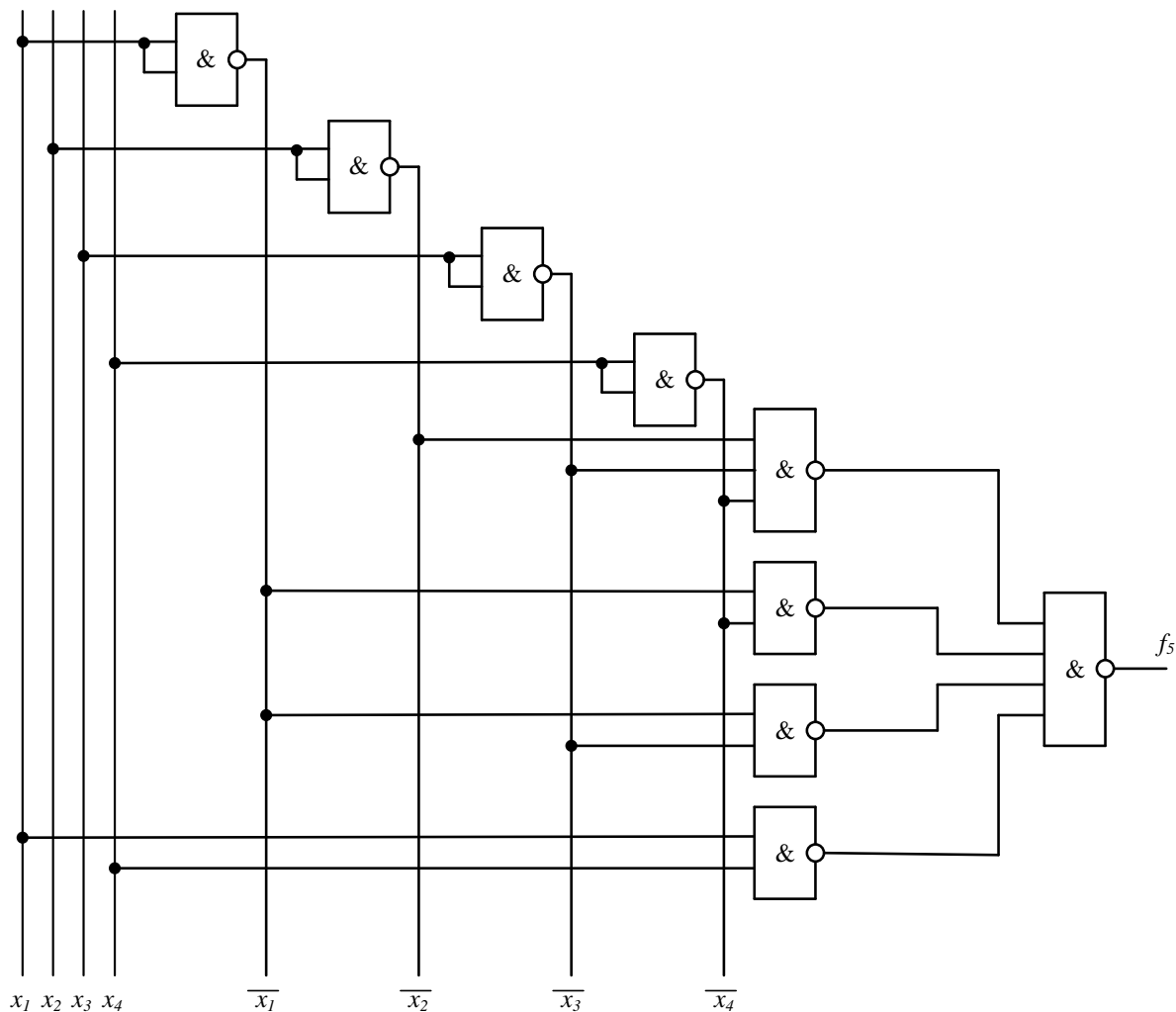


Рисунок 2.5 – Реалізація f_5 в базисі Шеффера

Виконаємо мінімізацію функції f_d та побудуємо схему:

		x_2			\bar{x}_2			
		1 12	0 13		1 9	0 8		\bar{x}_3
x_1		0 14	* 15		0 11	1 10		x_3
\bar{x}_1		1 6	1 7		1 3	1 2		
		0 4	1 5		1 1	1 0		\bar{x}_3
		\bar{x}_4	x_4		\bar{x}_4			

$$f_d = \bar{x}_1 \bar{x}_2 \vee \bar{x}_1 x_4 \vee \bar{x}_1 x_3 \vee \bar{x}_2 \bar{x}_3 x_4 \vee \bar{x}_2 x_3 \bar{x}_4 \vee x_1 x_2 \bar{x}_3 \bar{x}_4$$

$$f_d = \overline{\overline{\overline{\overline{\bar{x}_1 \bar{x}_2 \vee \bar{x}_1 x_4 \vee \bar{x}_1 x_3 \vee \bar{x}_2 \bar{x}_3 x_4 \vee \bar{x}_2 x_3 \bar{x}_4 \vee x_1 x_2 \bar{x}_3 \bar{x}_4}}} =$$

$$= \overline{\overline{x_1} \overline{x_2}} \wedge \overline{\overline{x_1} x_4} \wedge \overline{\overline{x_1} x_3} \wedge \overline{\overline{x_2} \overline{x_3} x_4} \wedge \overline{\overline{x_2} x_3 \overline{x_4}} \wedge \overline{x_1 x_2 \overline{x_3} \overline{x_4}}$$

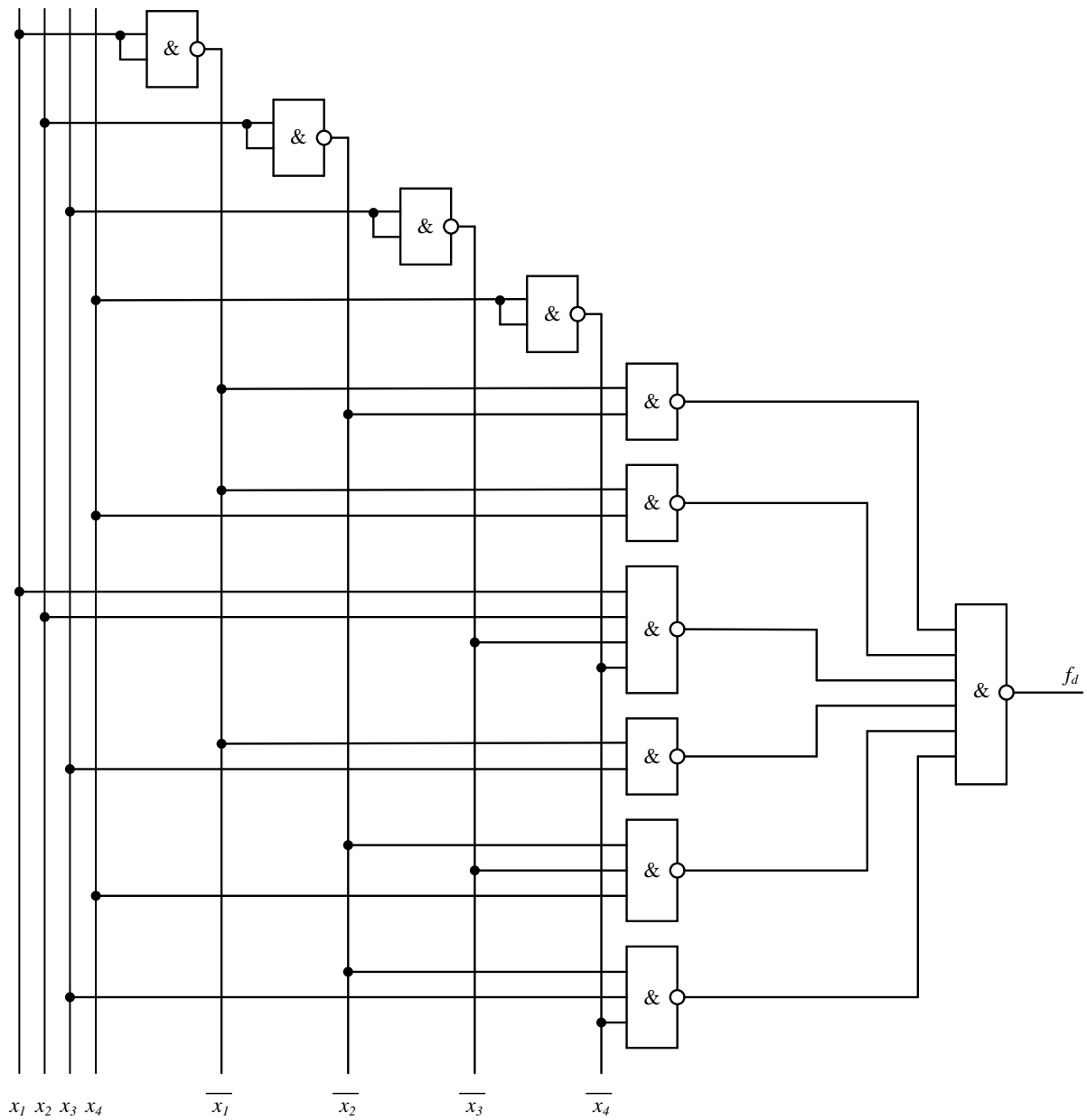


Рисунок 2.6 – Реалізація f_d в базисі Шеффера

2.2.2 Розробка віднімача на основі двійкового суматора. В пристрої, що розробляється необхідно проводити віднімання двійкових слів. Операція віднімання буде реалізована на двійковому суматорі.

Розглянемо приклад. Нехай необхідно з двійкового числа 1001110 відняти інше двійкове число 110101. Виконаємо операцію віднімання:

$$\begin{array}{r}
 1\ 0\ 0\ 1\ 1\ 1\ 0 \\
 -\quad 1\ 1\ 0\ 1\ 0\ 1 \\
 \hline
 \quad 1\ 1\ 0\ 0\ 1
 \end{array}$$

Такий же результат може бути отриманий, якщо провести складання вихідного числа зі зворотним кодом від'ємника числа. Переконаємося в цьому. Проінвертуємо друге число, тобто отримаємо його зворотний код. Це буде 1001011. Додаємо отримані двійкові числа.

$$\begin{array}{r}
 1\ 0\ 0\ 1\ 1\ 1\ 0 \\
 +\ 1\ 0\ 0\ 1\ 0\ 1\ 1 \\
 \hline
 1\ 0\ 0\ 1\ 1\ 0\ 0\ 0
 \end{array}$$

Отриманий результат відрізняється від необхідного на одиницю в старшому і в молодшому розрядах. В отриманому результаті старша одиниця є зайвою, зате бракує одиниці в молодшому розряді. Зазвичай цю проблему вирішують організацією циклічного перенесення одиниці старшого розряду і додаванням її до молодшого розряду.

Сформулюємо постановку задачі. Необхідно побудувати схему, яка повинна двічі виконувати операцію додавання - перший раз додавання першого числа з інверсним двійковим числом, а потім - додавання до молодшого розряду отриманої одиниці переносу. Дана схема може мати вигляд, представлений на рисунку 2.7.

Запропоновану схему доцільно реалізувати на комбінаційних повних суматорах. Суматори входять в номенклатуру кількох серій мікросхем ТТЛ. У складі серії К 155 випускаються три типи повних суматорів: однорозрядних К 155 ІМ1, двухрозрядних К 155 ІМ2 і чотирьохрозрядних К 155 ІМ3. Всі вони відносяться до розряду комбінаційних пристроїв, і сигнали суми і перенесення присутні на виході, поки діють вхідні сигнали.

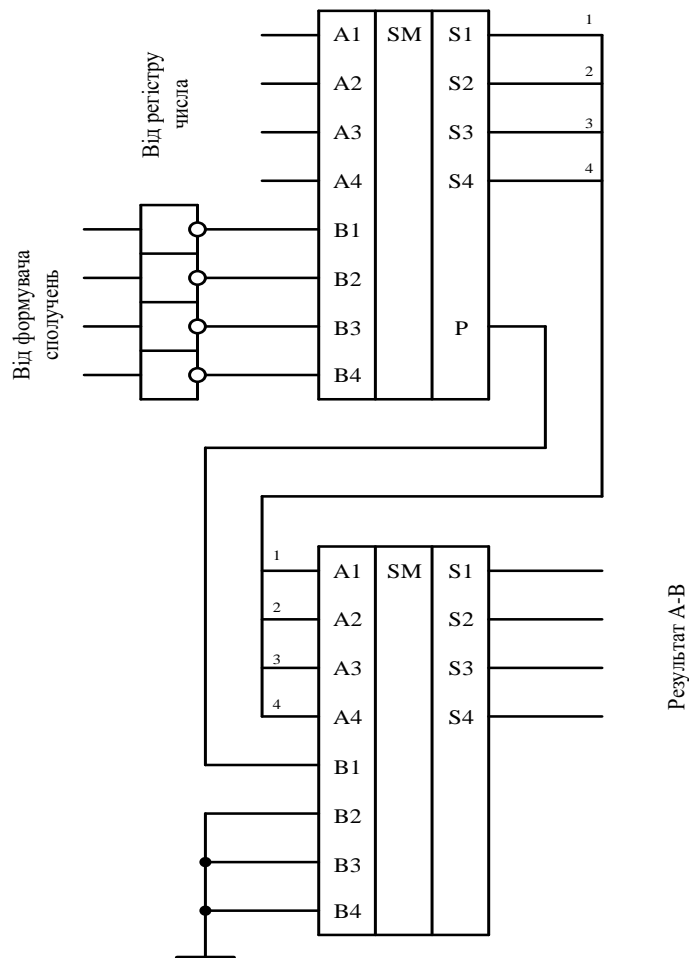


Рисунок 2.7 – Віднімач двійкових чисел

Розглянуті суматори побудовані на основі повних суматорів, які функціонують згідно з таблицею істинності (Таблиця 2.3).

Таблиця 2.3– Таблиця істинності повного суматора

№	Входи			Виходи	
	A_i	B_i	P_i	P_{i+1}	S_i
0	0	0	0	0	0
1	0	0	1	0	1
2	0	1	0	0	1
3	0	1	1	1	0
4	1	0	0	0	1
5	1	0	1	1	0
6	1	1	0	1	0
7	1	1	1	1	1

Функціонування суматора визначається наступними рівняннями:

$$S_i = A_i \bar{P}_i \vee B_i \bar{P}_i \vee P_{i-1} \bar{P}_i \vee A_i B_i P_{i-1}$$

$$P_i = B_i P_{i-1} \vee A_i P_{i-1} \vee A_i B_i.$$

У мікросхемах - суматорах в якості базового вузла використовується акумулятор, реалізований на основі отриманих формул.

2.2.3 Розробка блоку порівняння. В якості блоку порівняння вибираємо стандартний комбінаційний вузол - цифровий компаратор - мікросхема К1533СП1, умова функціонування якого наведена в таблиці 2.4.

Таблиця 2.4 – Таблиця станів цифрового компаратора К 1533СП1

Вхід порівняння даних				Вхід нарощування			Вихід		
A3, B3	A2, B2	A1, B1	A0, B0	$I(A>B)$	$I(A=B)$	$I(A<B)$	$(A>B)$	$(A=B)$	$(A<B)$
A3>B3	X	X	X	X	X	X	B	H	H
A3<B3	X	X	X	X	X	X	H	B	H
A3=B3	A2>B2	X	X	X	X	X	B	H	H
A3=B3	A2<B2	X	X	X	X	X	H	B	H
A3=B3	A2=B2	A1>B1	X	X	X	X	B	H	H
A3=B3	A2=B2	A1<B1	X	X	X	X	H	B	H
A3=B3	A2=B2	A1=B1	A0>B0	X	X	X	B	H	H
A3=B3	A2=B2	A1=B1	A0<B0	X	X	X	H	B	H
A3=B3	A2=B2	A1=B1	A0=B0	B	H	H	B	H	H
A3=B3	A2=B2	A1=B1	A0=B0	H	B	H	H	B	H
A3=B3	A2=B2	A1=B1	A0=B0	H	H	B	H	H	B
A3=B3	A2=B2	A1=B1	A0=B0	X	X	B	H	H	B
A3=B3	A2=B2	A1=B1	A0=B0	B	B	H	H	H	H
A3=B3	A2=B2	A1=B1	A0=B0	H	H	H	B	B	H

Цифрові компаратори відносяться до арифметичним пристроїв. Цифрові компаратори (від англ. *Compare* - порівнювати, звіряти) виконують порівняння двох чисел, заданих в двійковому (двійково-десятковому) коді. Залежно від схемного виконання компаратори можуть визначати рівність $A = B$ (A і B - незалежні числа з рівною кількістю розрядів) або вид нерівності: $A > B$ або $A < B$.

Результат порівняння відображається відповідним логічним рівнем на виході. Мікросхеми - цифрові компаратори - виконують, як правило, всі ці операції і мають три виходи. Цифрові компаратори широко застосовуються для виявлення потрібного числа (слова) в потоці цифрової інформації, для позначки часу в часових приладах, для виконання умовних переходів в обчислювальних пристроях.

У формувачі двійкових біноміальних чисел, що розробляється, компаратор порівнює вихідне двійкове число (чи різницю) і вагу відповідного розряду. Вміст регістру поточної вибірки подається на одну групу входів компаратора, на другу групу – надходить вага відповідного розряду біноміального числа.

2.2.4 Вибір регістрів. В якості регістру поточної вибірки і регістру біноміального числа вибираємо тригерну збірку К 1533 ТМ8, що містить чотири *D* - тригера з паралельним керуванням. Функціональне позначення тригерної збірки наведено на рисунку 2.8.

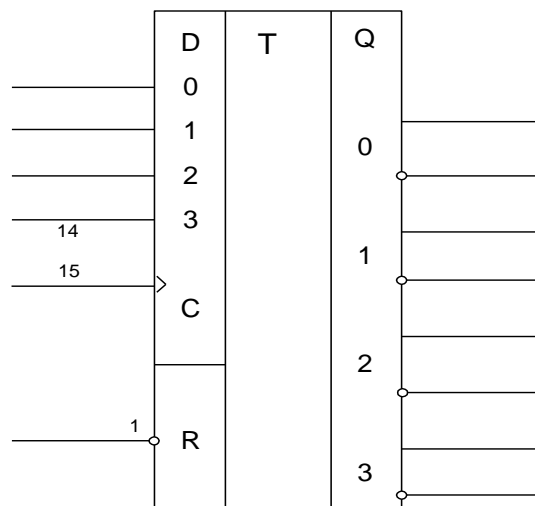


Рисунок 2.8 – Цоколювання мікросхеми К 1533 ТМ8

У регістр поточної вибірки початкове 8-ми розрядне двійкове число записується в паралельному коді, потім зберігається і обробляється також в паралельному коді, тому доцільно керувати всіма тригерами збірки одночасно.

Принципова схема формувача рівноважних кодових комбінацій в основному реалізована на мікросхемах серії К 1533, і тільки коли в цій серії не виявилось потрібних мікросхем були обрані мікросхеми серії К 155. Це чотирирозрядні суматори К 155 ІМ3 і двовходові елементи «АБО» К 155 ЛЛ1.

ВИСНОВКИ

Дана робота присвячена розробці генератора рівноважних кодових комбінацій формувача рівноважних кодових комбінацій на основі двійкових біноміальних чисел.

Рівноважні кодові комбінації широко застосовуються як в системах передачі інформації, так і в пристроях діагностики складних керуючих систем, які працюють в реальному часі в умовах високого рівня завад. Широке застосування цих кодів обумовлено їх здатністю виявляти всі помилки, крім помилок «зсуву», коли відбуваються одночасні переходи нуля в одиницю та одиниці в нуль.

В результаті проведеного аналізу літературних джерел була запропонована класифікація завадостійких кодів, проаналізовані сучасні методи генерування рівноважних кодів.

В роботі розглядаються два методи – швидкодіючий – на основі побудови структури перетворювача довільних кодів та більш універсальний – з застосуванням в якості проміжної системи числення з двійковим алфавітом. Генератор рівноважних кодів, побудований за цим методом, має властивість формувати рівноважні коди в широкому діапазоні заданих параметрів.

В процесі проектування був розроблений алгоритм функціонування пристрою, запропонована структурна схема генератора рівноважних кодових комбінацій на основі застосування біноміальних кодів з двійковим алфавітом. Подальша деталізація структурної схеми привела до розробки функціональної і принципової схем генератора рівноважних кодових комбінацій.

Також були розроблені функціональна та принципова схема перетворювача двійкових кодових комбінацій в рівноважні. Пристрій побудований з застосуванням мікросхем найпоширенішого функціонального базису – базису Шеффера. Було виконано логічний синтез та побудована принципова схема.

Були проаналізовані структури одержаних схемотехнічних рішень.

За необхідністю максимально швидкого перетворення краще застосовувати довільний перетворювач кодів, незважаючи на те, що для різних параметрів треба будувати окрему схему. У випадку необхідності побудови універсального пристрою, який може забезпечити широкий діапазон параметрів

рівноважних кодів, доцільнішим буде використання генератора рівноважних кодів на основі отримання біноміальних кодових комбінацій.

При побудові принципових схем була застосована серія 1533, для якої характерна досить висока швидкість перемикання, невелика споживана потужність, широке різноманіття функціонального складу.

Таким чином завдання виконане успішно.

СПИСОК ЛІТЕРАТУРИ

- 1 Ubar R., Raik J., Vierhaus H.-T. Design and Test Technology for Dependable System-on-Chip (Premier Reference Source). Information Science Reference, Hershey. New York, IGI Global, 2011, 578 p.
- 2 Kharchenko V., Kondratenko Yu., Kasprzyk J. Green IT Engineering: Concepts, Models, Complex Systems Architectures // Springer Book series "Studies in Systems, Decision and Control", Vol. 74, 2017, 305 p.
- 3 Lala P.K. Self-Checking and Fault-Tolerant Digital Design. San Francisco: Morgan Kaufmann Publishers, 2001, 216 p.
- 4 Ryan W.E., Lin S. Channel Codes: Classical and Modern. Cambridge University Press, 2009, 708 p.
- 5 Göessel M., Ocheretny V., Sogomonyan E., Marienfeld D. New Methods of Concurrent Checking: Edition 1. Dordrecht: Springer Science+Business Media B.V., 2008, 184 p.
- 6 Das D., Toubia N.A., Seuring M., Gossel M. Low Cost Concurrent Error Detection Based on Modulo Weight-Based Codes // Proc. of IEEE 6th International On-Line Testing Workshop (IOLTW). Spain, Palma de Mallorca, July 3-5, 2000, pp. 171—176. DOI:10.1109/OLT. 2000.856633.
- 7 Протасова Т.О., Бородін В.М., Усик А.О. Засоби кодування на основі двійкових біноміальних чисел для застосування в телекомунікаційних системах // Фізика, електроніка, електротехніка (ФЕЕ-2023). Матеріали та програма міжнародної науково-технічної конференції студентів та молодих вчених, 24-28 квітня 2023 р. Суми, СумДУ, 2023. – С. 99
- 8 Кушнір О.І. Аналіз методів завадостійкого кодування у цифрових системах зв'язку [Електронний ресурс] / О.І. Кушнір, О.І. Тимочко, О.В. Северінов // Системи обробки інформації. – 2007. – Вип. 9. – С. :63-65.
- 9 Пятін І.С. Моделювання цифрової системи зв'язку з завадостійким кодуванням [Електронний ресурс] / І.С. Пятін, В.В. Сергеев // Вісник Хмельницького національного університету. Технічні науки. – 2017. - № 6. – С. 89-91.
- 10 Гребенюк О. П. Застосування завадостійкого кодування в системах зв'язку і передачі даних комплексів радіомоніторингу для забезпечення достовірності інформаційного обміну / О.П. Гребенюк, В.Д. Меленський, В.І. Коріненко // Проблеми створення, випробування,

застосування та експлуатації складних інформаційних систем. - 2015. - Вип. 11. - С. 44-50.

- 11 V. M. Deundyak, Yu. V. Kosolapov, “On the Berger–Loidreau cryptosystem on the tensor product of codes”, *J. Comp. Eng. Math.*, **5:2** (2018), 16–33
- 12 Цифрова схемотехніка: Підручник у двох томах, том 2. Жуйков В.Я., Бойко В.І., Зорі А.А. та ін. –К.: Аверс, 2002.- 408 с.
- 13 В.М. Рябенський, В.Я. Жуйков, Ю.С. Ямненко, А.В. Заграничний. Схемотехніка: Пристрої цифрової електроніки: Електронний підручник. – НТУУ «КПІ», Київ 2016. – 399 с.