

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
Кафедра електроніки і комп'ютерної техніки

ПОЯСНЮВАЛЬНА ЗАПИСКА

ДО КВАЛІФІКАЦІЙНОЇ РОБОТИ МАГІСТРА
на тему:

**«АДАПТИВНА ЕЛЕКТРОННА СИСТЕМА ОБЧИСЛЕННЯ ДВІЙКОВИХ
ОДИНИЦЬ В ІНФОРМАЦІЙНИХ ПОСЛІДОВНОСТЯХ»**

Завідувач кафедри

Опанасюк А. С.

Керівник проекту

Кулик І. А.

Консультант

з техніко-економічної частини

Маценко О. М.

Розробив студент групи ЕС.м-21

Греков Т. В.

Суми 2023

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

Факультет _____ електроніки та інформаційних технологій

Кафедра _____ електроніки і комп'ютерної техніки

Напрямок підготовки _____ 8.171.00.10 «Електроніка»

Освітня програма _____ Електронні системи та компоненти

ЗАТВЕРДЖУЮ

Зав. кафедрою _____ Опанасюк А. С.

"__" _____ 2023 р.

ЗАВДАННЯ

на кваліфікаційну роботу магістра

1. Тема роботи _____

затверджена наказом по університету " __ " _____ 2023 р. № _____

2. Термін здачі студентом завершеної роботи _____

3. Вихідні дані до роботи _____

4. Зміст розрахунково-пояснювальної записки (перелік питань, що належить розробити) 1) Огляд літератури та поставлення задачі роботи. 2) Науково-дослідна частина. 3) Розробка електронної системи з використанням отриманих результатів дослідження. 4) Техніко-економічна частина.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) 1) Схема електрична структурна. 2) Схема алгоритму. 3) Схема електрична функціональна. 4) Схема електрична принципова.

6. Консультанти з кваліфікаційної роботи

Розділи	Консультанти	Завдання видав	Завдання прийняв
Техніко-економічна частина	Маценко О. М.		

7. Дата видачі завдання _____

8. Керівник роботи _____

9. Завдання прийняв до виконання _____

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів дипломного проекту	Термін виконання етапів роботи	Примітки
1	Огляд літератури й постановка завдання дослідження	20.09.23 – 10.10.23	
2	Науково-дослідна частина	10.10.23 – 20.10.23	
3	Розробка алгоритму функціонування	20.10.23 – 25.10.23	
4	Розробка та обґрунтування структурної схеми	25.10.23 – 30.10.23	
5	Розробка функціональної схеми	30.10.23 – 05.11.23	
6	Розробка та розрахунок принципової схеми	05.11.23 – 15.11.23	
7	Техніко-економічна частина	15.11.23 – 25.11.23	
8	Оформлення пояснювальної записки	25.11.23 – 30.11.23	
9	Оформлення графічного матеріалу	30.11.23 – 10.12.23	
10	Представлення роботи керівнику і отримання відгуку	15.12.23	
11	Представлення роботи кафедрі для отримання рецензії	19.12.23	

Студент _____

Керівник роботи _____

«___» _____ 2023 р.

РЕФЕРАТ

Пояснювальна записка містить: 80 сторінок, 22 рисунки, 14 літературних джерел, дві структурні схеми та одну принципову.

В даній кваліфікаційній роботі було спроектовано адаптивну електронну систему обчислення двійкових одиниць в інформаційних послідовностях, також було проведено огляд літератури способів підрахунку одниниць.

Пояснювальна записка містить чотири розділи. Перший розділ містить огляд літератури за вибраною темою проектування. В другому розділі приводиться науково-дослідна частина. У третьому розділі показана розробка електронної системи з використанням отриманих результатів досліджень. Четвертий розділ присвячений техніко-економічній частині.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1 ОГЛЯД ЛІТЕРАТУРИ ТА ПОСТАНОВКА ЗАВДАННЯ ПРОЕКТУВАННЯ.....	8
1.1 СПОСОБИ ПІДРАХУВАННЯ ОДИНИЦЬ В ДВІЙКОВИХ ПОСЛІДОВНОСТЯХ.....	8
1.2 ОГЛЯД ОБЛАСТЕЙ ЗАСТОСУВАННЯ СПОСОБІВ ПІДРАХУНКУ В ЕЛЕКТРОННИХ СИСТЕМАХ	16
1.3 АПАРАТНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ОБЧИСЛЕННЯ КІЛЬКОСТІ ОДИНИЦЬ В ДВІЙКОВИХ КОМБІНАЦІЯХ.....	17
1.4 ПОСТАНОВКА ЗАВДАННЯ ПРОЕКТУВАННЯ.....	21
РОЗДІЛ 2 НАУКОВО-ДОСЛІДНА ЧАСТИНА.....	23
РОЗДІЛ 3 РОЗРОБКА ЕЛЕКТРОННОЇ СИСТЕМИ З ВИКОРИСТАННЯМ ОТРИМАНИХ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ	27
3.1 РОЗРОБКА ТА ОБГРУНТУВАННЯ АЛГОРИТМУ ФУНКЦІОНУВАННЯ	27
3.2 РОЗРОБКА СТРУКТУРНОЇ СХЕМИ ПРОЕКТОВАНОЇ СИСТЕМИ.....	34
3.3 РОЗРОБКА ЕЛЕКТРИЧНОЇ ФУНКЦІОНАЛЬНОЇ СХЕМИ.....	38
3.4 ВИБІР ЕЛЕМЕНТНОЇ БАЗИ.....	40
3.5 РОЗРОБКА ПРИНЦИПОВИХ СХЕМ БЛОКІВ	52
3.5.1 Розробка блоку розподілу розрядів	53
3.5.2 Розробка блоку виявлення одиниць	56
3.5.3 Розробка блоку пам'яті	57
3.5.4 Розробка блоку арифметичних операції	58
3.5.5 Розробка блоку складання двійкових чисел	61
3.5.6 Розробка блоку лічильника одиниць	62
3.5.7 Розробка блоку керування	62
3.5.8 Розробка блоку вибору режиму	64
3.5.9 Розробка блоку зсуву	65
3.5.10 Розробка блоку генератора.....	67
РОЗДІЛ 4 ТЕХНІКО-ЕКОНОМІЧНА ЧАСТИНА.....	68

					ЕліТ 8.171.00.10.445 ПЗ			
Зм.	Лист	№ документа	Підпис	Дата				
Розробник		Греков Т.В.			Адаптивна електронна система обчислення двійкових одиниць в інформаційних послідовностях Пояснювальна записка	Лит	Лист	Листів
Перевірів		Кулик І.А.					5	80
Реценз.						СумДУ, гр. ЕС.м-21		
Н. Контр		Гапич В.М.						
Затверд.		Опанасюк А.С.						

4.1 РОЗРАХУНОК ПОВНОЇ СОБІВАРТОСТІ СИСТЕМИ.....	68
4.2 РОЗРАХУНОК ЦІНИ СИСТЕМИ	73
4.3 ВИЗНАЧЕННЯ РІЧНИХ ВИТРАТ СПОЖИВАЧА.....	74
4.4 РОЗРАХУНОК ПОВНОЇ ЦІНИ КОРИСТУВАЧА	76
4.5 ВИСНОВКИ ДО ТЕХНІКО-ЕКОНОМІЧНОЇ ЧАСТИНИ.....	76
ВИСНОВОК.....	78
СПИСОК ЛІТЕРАТУРИ.....	79
СПЕЦИФІКАЦІЯ.....	80

ВСТУП

Однією з ключових операцій у стисненні інформації, двійковому кодуванні автоматизації процесів у промисловості, аналізі дискретної інформації та захисті даних від помилок є необхідність визначення кількості одиниць у двійкових послідовностях. Це часто вимагає розробки пристроїв, які відповідають вимогам проектування. Основні вимоги включають швидкість, надійність та апаратне забезпечення. Таким чином, існують потреби у розробці більш ефективних методів та електронних пристроїв для обчислення кількості одиниць у двійкових послідовностях, що враховуватимуть вимоги швидкості та мінімізації ресурсів.

Метою кваліфікаційної роботи магістра є створення ефективної адаптивної системи для обчислення кількості одиниць у двійкових послідовностях. Основною вимогою до проектування є швидкість обчислень, що робить його важливим завданням у даному контексті.

					<i>ЕліТ 8.171.00.10.445</i>	Лист
						7
<i>Зм.</i>	<i>Лист</i>	<i>№ документа</i>	<i>Підпис</i>	<i>Дата</i>		

РОЗДІЛ 1 ОГЛЯД ЛІТЕРАТУРИ ТА ПОСТАНОВКА ЗАВДАННЯ ПРОЕКТУВАННЯ

1.1 Способи підрахування одиниць в двійкових послідовностях

Розрахунок кількості одиниць у двійкових послідовностях є важливою операцією в різних областях, таких як стиснення повідомлень, захист даних від помилок і генерація комбінаторних конфігурацій. Ця задача отримує значний акцент у розробці ефективних та економічних методів обчислення кількості одиниць, а також у реалізації електронних пристроїв, які здійснюють цей процес.

Визначення кількості одиниць у вихідних двійкових послідовностях є ключовою операцією для методів захисту і стиснення даних, включаючи техніки двійкового біномного стиснення. Це завдання важливе у контексті забезпечення безпеки та ефективності обробки інформації, і йому приділяється значна увага у вдосконаленні алгоритмів та розробці спеціалізованих електронних засобів для його виконання.

Розглянемо простий метод обчислення кількості одиниць у двійкових послідовностях 1.1, який включає послідовний перегляд кожного розряду n -бітової двійкової комбінації $V_j = n_1n_2n_3n_4\dots n_i\dots n_a$. (V_j – двійкова комбінація). У цьому методі, якщо, наприклад, $n_1 = 1$, змінна, відповідальна за підрахунок одиниць, збільшується відповідно. Хоча такий підхід має просту схемотехніку, його недоліком є значна затрата часу, еквівалентного часу n машинних тактів, для підрахунку k одиниць у найгіршому випадку.

Для деяких обчислювальних пристроїв такий метод може бути оптимальним, але для швидкодіючих пристроїв це стає неприйнятним через велику кількість часу, особливо коли значення n велике. У таких випадках виникає необхідність в ефективніших алгоритмах для обчислення кількості одиниць у двійкових послідовностях з метою зниження часових витрат.

Наступний метод є модифікацією попереднього, включаючи одночасний паралельний підрахунок одиниць у різних частині двійкової послідовності. Перед початком обчислення послідовність розбивається на частини, і підрахунок відбувається одночасно для кожної частини. Цей підхід може значно зменшити час обчислення до n/m разів, де m - кількість частин розбиття вихідної двійкової послідовності. Проте це призводить до збільшення обсягу апаратних витрат

Зм.	Лист	№ документа	Підпис	Дата

приблизно в той же самий раз, оскільки потрібно використовувати суматори для обробки результатів з кожної частини.

Цей метод є компромісом між часом виконання і обсягом апаратних витрат. Для деяких застосувань, де ефективність обчислення має пріоритет, апаратні витрати можуть бути прийнятними. Однак у випадках, де важливо мінімізувати обсяг апаратури, можливо, слід розглядати інші методи для оптимізації розрахунків кількості одиниць у двійкових послідовностях.

Використання однорозрядних матричних суматорів для паралельного підрахунку одиниць є крайнім випадком паралельності, який може забезпечити максимальну швидкодію. Однак, як виправдано зауважили, це призводить до значної кількості апаратних витрат, що ускладнює його застосування в невеликих і недорогих електронних пристроях та системах.

Використання таких матричних суматорів дозволяє одночасно обчислювати значення для кожного розряду паралельно, що в теорії прискорює операцію. Однак реальні витрати на виготовлення та використання такого апарату великі, що часто робить його непрактичним для реалізації у пристроях з обмеженими ресурсами або бюджетом.

У розробці електронних пристроїв важливо знаходити баланс між швидкодією та вартістю апаратури, щоб забезпечити оптимальний компроміс залежно від конкретних вимог та обмежень проекту.

Більш привабливими виявляються методи комбінаторних підходів для обчислення кількості одиниць, які "розв'язують" визначення часу, необхідного для знаходження кількості одиниць у вихідній інформаційній послідовності в порівнянні з прямо пропорційним зростанням часу обчислення від довжини комбінації до більш повільного логарифмічного. Суть методу полягає в застосуванні обчислювального процесу типу виду $V = V \wedge (V - 1)$. де V - двійкова комбінація, \wedge - логічна операція І, яка виконується паралельно з кожним відповідним розрядом пари аргументів V та $(V - 1)$. Цей спосіб визначення кількості двійкових одиниць використовується у пристроях, що можуть виконувати арифметичні та логічні операції над розрядами двійкових послідовностей. Зміни, які пропонуються, націлені на покращення швидкості підрахунку числа k одиниць у двійкових послідовностях [3].

											Лист
											9
Зм.	Лист	№ документа	Підпис	Дата	ЕліТ 8.171.00.10.445						

$$\begin{array}{r|l}
V & 1011000 \\
V-1 & 1010111 \\
\hline
V \wedge V-1 & 1010000
\end{array}$$

Цей алгоритм продовжуватиме свою роботу до того моменту, поки A не стане рівним нулю, тобто складатиметься лише з нулів. На відміну від розглянутого раніше методу, цей спосіб використовується тільки стільки разів, скільки одиниць є в наборі A . Іншими словами, алгоритм працює лише з одиницями, не знаючи їх точного розташування. Цей алгоритм є очевидно ефективним, особливо коли застосовується до підрахунку двійкових послідовностей, де переважають нулі.

Удосконалення попереднього способу підрахунку одиниць у двійковій послідовності A , заснованого на обчислювальній процедурі $A = A \wedge (A - 1)$, яка проводиться доти, доки $A = 0$. Удосконалення полягає в тому, що одночасно з тим, як обробляється комбінація A , проводиться паралельно така сама операція

Покращення попереднього методу підрахунку одиниць у двійковій послідовності A , що ґрунтується на обчислювальній процедурі $V = V \wedge (V - 1)$, яка виконується до того моменту, поки $V = 0$. Удосконалення полягає в тому, що одночасно з обробкою комбінації V , проводиться паралельно така сама операція, але з інверсією її значень.

Такий підхід дозволяє принаймні вдвічі прискорити процес обчислення значень k .

У загальному вигляді, цей метод можна уявити як систему з двох каналів, де перший канал обробляє комбінацію V , а другий - її інверсію \bar{V} . Основним складовими блоками такого пристрою є блоки віднімання, блоки кон'юнкції, блоки лічильника суми, блоки лічильника віднімання та блоки комутації. Ці блоки мають просту схемотехнічну реалізацію.

Розглянутий метод підрахунку кількості одиниць у двійковій послідовності, а також його практична реалізація, дозволяють значно скоротити час виконання відповідної операції.

Застосування арифметичного методу залежить від методів захисту чи стиснення інформації. У випадку використання методів перетворення двійкових даних, для яких потрібно точно знати значення k , пропонується в моделі процесу підрахунку одиниць виконувати одночасно операції $B = B \wedge (B - 1)$ та $\bar{B} = \bar{B} \wedge (\bar{B} - 1)$. Ця зміна дозволить принаймні вдвічі зменшити середній час обчислення кількості одиниць. Для методів захисту та стиснення двійкової інформації, умови роботи яких залежать від граничних значень k , пропонується додатково ввести обмежувальну змінну L , яка приймає значення кордонів a або $n - a$. У випадку перевищення кількості одиниць (нулів) значення L , подальший підрахунок одиниць припиняється, оскільки точне значення k вже не є необхідним. Це покращення також, як і попереднє, значно пришвидшує час обчислення k інформаційних одиниць.

Тоді модель процесу обчислення точного числа k одиниць для бінарної послідовності A виглядає наступним чином.

Етап 1. Генерується інверсія B :

$\bar{B} = \bar{b}_1 \bar{b}_2 \dots \bar{b}_i \bar{b}_n$, отримуючи тим самим можливість працювати як з B , так і з його інверсійною формою \bar{B} .

Етап 2. У випадку, коли B має вигляд $00\dots 0$, процес обчислення кількості двійкових одиниць завершується, і змінна k містить необхідне значення.

Етап 3. Якщо \bar{B} дорівнює $00\dots 0$, то виконується операція $k = n - k$. Після цього процес обчислення двійкових одиниць також завершується, і змінна k містить потрібне значення. В іншому випадку слід перейти до наступного етапу.

Етап 4. Проводиться операції над B та \bar{B} :

$B = B \wedge (B - 1)$ та $\bar{B} = \bar{B} \wedge (\bar{B} - 1)$ з наступним збільшенням змінної k на одиницю ($k = n - k + 1$). Після цього виконується перехід до етапу 2 для перевірки, чи рівні нулю значення B та \bar{B} .

Для методів перетворення двійкової інформації, коли точне значення k необхідно лише до граничних значень a або $n - a$ наведена вище модель при зміні етапу 4 доповнюється етапом 5, що дозволяє завершити підрахунок при перевищенні граничного значення a .

Етап 5. У випадку, коли виконується умова $k \leq L$, де $L = a$, відбувається перехід до етапу 2. В іншому випадку процес підрахунку завершується, і змінній k присвоюється шаблонне значення $k = L + 1$.

Цікавіший алгоритм, викладений далі. З алгоритмом який розглядали на початку 1.1 має одну загальну важливу властивість: повторення циклу фіксоване число разів, що залежить від n і не залежить від V . У той же час на відміну від алгоритму 1.1, що повторює свій цикл n раз, в алгоритмі 1.3 (докладного викладу якого ми не даємо) цикл повторюється лише $(\lg n)$ разів). Для типового слова довжини 32 (або 64) цикл повторюється п'ять (відповідно шість) разів, тобто алгоритм 1.3 значно швидшим, ніж алгоритм 1.1. Алгоритм 1.3 виходить із тих самих припущень про можливості обчислювального пристрою, що й алгоритм 1.2. Додатково потрібно, щоб був спосіб швидко зрушувати слово на 1, 2, 4, 8... розрядів[3].

Цей алгоритм краще пояснити на конкретному прикладі.

	b8	b7	b6	b5	b4	b3	b2	b1
V	1	1	0	1	0	0	0	1

1. Спочатку виділяються розряди з непарними номерами $b7, b5, b3, b1$ та зліва від кожного з них приписуються нулі. Отриманий таким чином набір позначимо $V_{неп}$

	b7	b5	b3	b1
$V_{неп}$	0	1	0	1

Потім виділяються парні координати і зсуваються праворуч на один розряд на місця розрядів $b7, b5, b3, b1$ відповідно. До кожного з розрядів припишемо зліва нуль, а отриманий таким чином рядок позначимо $V_{пар}$

	b8	b6	b4	b2
V _{пар}	0	1	0	0

(Нулі, приписані розрядам, виділені дрібним шрифтом, щоб відрізнити їх від нулів, що входять до набору В.) Після цього складаємо два числа, двійковими уявленнями яких є V_{неп} і V_{пар}. Через V' позначимо набір, що являє собою двійковий запис результату підсумовування

	b'8	b'7	b'6	b'5	b'4	b'3	b'2	b'1
V _{неп}	0	1	0	1	0	0	0	1
V _{пар}	0	1	0	0	0	0	0	0
V'	1	0	0	1	0	0	0	1

2. З отриманого набору виділяємо пари розрядів b'6 b'5 та b'2, b'1 і зліва від кожної пари приписуємо по два нулі. Позначимо V'_{неп} отриману таким чином послідовність.

	b'6	b'5	b'2	b'1
V' _{неп}	0	0	0	1

Потім виділяються і зсуваються на два розряди праворуч інші пари b'8 b'7 та b'4, b'3. Після такої операції вони займатимуть місця пар розрядів b'6 b'5 та b'2, b'1 відповідно. Як і раніше, ліворуч від кожної пари приписуємо по два нулі. В результаті отримаємо набір V'_{пар}

				b'8	b'7			b'4	b'3
V'пар	0	0	1	0	0	0	0	0	0

Складаємо два числа V'неп і V'пар. Двійковий запис цієї суми позначемо V''

		b''8	b''7	b''6	b''5	b''4	b''3	b''2	b''1
V''	0	0	1	1	0	0	0	0	1

3. Беремо розряди b''4, b''3, b''2, b''1 і приписуємо до цієї четвірки зліва чотири нулі, в результаті чого отримуємо набір V''неп.

						b''4	b''3	b''2	b''1
V''неп	0	0	0	0	0	0	0	0	1

Так само беремо розряди b''8, b''7, b''6, b''5 зрушуємо їх вправо на чотири розряди на місця b''4, b''3, b''2, b''1 відповідно, приписуємо до них зліва чотири нулі і отримуємо набір V''пар

						b''8	b''7	b''6	b''5
V''пар	0	0	0	0	0	0	0	1	1

Нарешті, складаємо два числа, двійковими уявленнями яких є $V^{неп}$ і $V^{пар}$. Послідовність $V=(00000100)$, що являє собою двійковий запис результату підсумовування, одночасно є двійковим записом суми розрядів слова V (в даному випадку чотири) [3].

Якщо n довільне і не є ступенем двійки, то до слова V припишем попередньо ліворуч нули так, щоб сумарна довжина стала ступенем двійки, найближчою зверху до n . Після такого попереднього зауваження легко узагальнити алгоритм на випадок слів будь-якої довжини.

Чи існують швидші алгоритми для обчислення числа одиниць у слові? Чи існує «оптимальний» алгоритм? Питання оптимальності алгоритмів важливе, але він потребує конкретизації. Щоб показати оптимальність алгоритму, необхідно точно визначити клас допустимих алгоритмів і критерій оптимальності. У разі алгоритмів підрахунку числа одиниць у наборі така конкретизація була б складною і досить нечіткою, що включає подробиці про способи роботи обчислювального пристрою.

Однак ми можемо навести правдоподібні докази на користь того, що запропонований далі алгоритм підсумовування числа одиниць (алгоритм 1.4) є найшвидшим з усіх можливих, оскільки він, використовуючи пошукову таблицю, отримує результат у сутності за операцію. Платою за таку швидкість є марнотратне використання об'єму пам'яті (2^n комірок), що для великих значень n і робить алгоритм практично непридатним до вживання. однієї точки зору, тим він гірший з іншого[2].

Алгоритм 1.4 заснований на тій ідеї, що можна заздалегідь вирішити задачу для всіх можливих наборів фіксованої довжини, зберігати в пам'яті отримані результати і потім шукати серед них те, що нам потрібно. У разі $n=3$ ми маємо зберігати інформацію наступного виду:

Слово	Число одиниць
000	0
001	1
010	1
011	2
100	1
101	2
110	2
111	3

Який найшвидший шлях пошуку у цій таблиці? При тих же припущеннях, що й у попередніх алгоритмах, можна вважати на адресу комірки пам'яті, що містить суму одиниць у В. Цей факт і дає нам алгоритм, що вимагає лише одного звернення до пам'ят[2].

Завершуючи розділ, відзначимо велику різноманітність існуючих алгоритмів обчислення числа одиниць у наборі, які ґрунтуються на абсолютно різних принципах. Алгоритми 1.1 і 1.4 вирішують завдання «в лоб»: алгоритм 1.1 переглядає кожен розряд і тому потребує багато часу, алгоритм 1.4 зберігає рішення кожного набору і тому потребує великого обсягу пам'яті. Алгоритм 1.3 є витонченим компромісом між ними.

1.2 Огляд областей застосування способів підрахунку в електронних системах

Двійкові послідовності, які складаються з нулів та одиниць, є основними засобами передачі інформації у більшості сучасних та високотехнологічних пристроях. Проте більшість програмістів рідко працюють з інформацією на рівні двійкових послідовностей. Це є зрозумілим, особливо в контексті чисельних обчислень, де програмісти зазвичай фокусуються на арифметичних операціях та рідко вдаються до вивчення внутрішнього представлення чисел.

З іншого боку, в деяких областях використання чисел у формі двійкових послідовностей не є настільки поширеним. Деякі важливі операції, які наразі складно виконувати на обчислювальних пристроях або високорівневих мовах програмування, використовують комбінаційні обчислювальні класи операцій.

Для ефективного програмування в таких областях необхідно розуміти алгоритми, які працюють на рівні двійкових символів.

Обчислення кількості одиниць у двійкових послідовностях є важливою операцією в різних сферах, таких як стиснення повідомлень, двійкове кодування для різних кодів, автоматизація промислових процесів у дискретних аналізах, захист даних від помилок та генерація комбінаторних конфігурацій. Знання кількості одиниць дозволяє створювати різноманітні коди, визначати стан керованих об'єктів у системах автоматизації та проводити біомне стиснення двійкових комбінацій.

У зв'язку з цим, велика увага приділяється розробці швидкодіючих та ефективних методів для підрахунку кількості одиниць у двійкових послідовностях, що дозволяє здійснювати цю операцію з високою швидкістю та мінімізувати апаратно-програмні витрати. Це особливо актуально в контексті розробки електронних пристроїв, які забезпечують ефективне виконання даного завдання в різних сценаріях застосування.

1.3 Апаратне забезпечення для обчислення кількості одиниць в двійкових комбінаціях

Апаратне забезпечення представляє собою набір елементів, які входять у склад електронних обчислювальних пристроїв. В його склад входять логічні, арифметичні, цифрові та аналогові електронні схеми. Ці пристрої дозволяють забезпечити як ефективне функціонування будь-яких систем, так і виконання основних функцій з можливістю їх розширення. Розглянемо компоненти, які будуть використані при розробці високоефективного пристрою для обчислення кількості одиниць у двійкових послідовностях.

Кон'юнктор

Кон'юнктор є логічним елементом, що втілює операцію \wedge (множення) і, зазвичай, має від двох до восьми входів та один вихід[1]. Вихідний сигнал кон'юнктора рівний 1 лише у випадку, якщо всі входи також мають сигнал 1[1]. На рисунку 1.1 зображено графічне представлення кон'юнктора з двома входами.

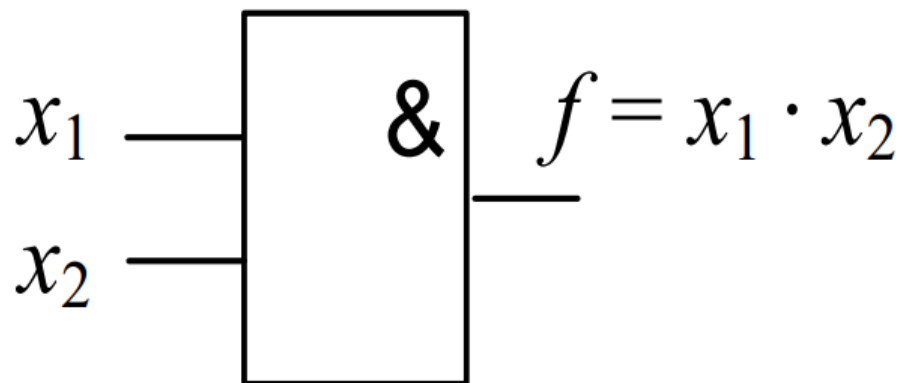


Рисунок 1.1 – Графічне зображення кон'юнктора

В таблиці 1.1 показано таблицю істиності станів кон'юнктора

Таблиця 1.1 – Таблиця істиності станів кон'юнктора

Входи		Вихід
x_1	x_2	f
0	0	0
0	1	0
1	0	0
1	1	1

Диз'юнктор

Диз'юнктор є логічним елементом, що реалізує логічну операцію АБО (додавання), зазвичай має від двох до восьми входів та один вихід. Вихідний сигнал диз'юнктора рівний 0 лише у випадку, якщо на всі входи також подається сигнал 0. На рисунку 1.2 зображено графічне представлення диз'юнктора з двома входами[1].

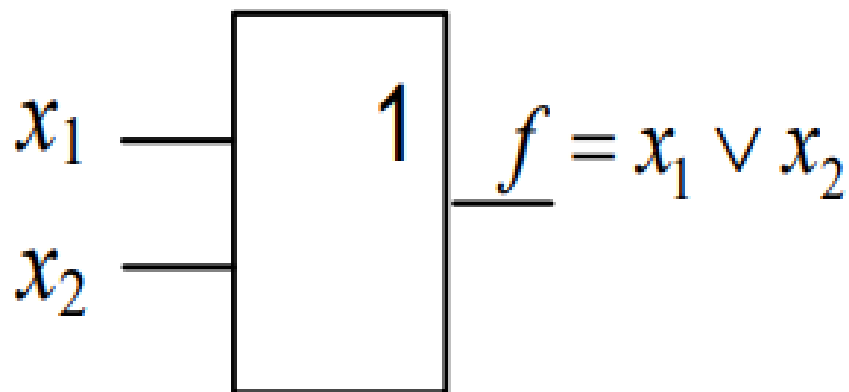


Рисунок 1.2 – Графічне зображення диз'юнктора

В таблиці 1.2 показано таблицю істиності станів диз'юктора

Таблиця 1.2 – Таблиця істиності станів диз'юктора

Входи		Вихід
X_1	X_2	f
0	0	0
0	1	1
1	0	1
1	1	1

Дешифратор

Дешифратор - це цифровий пристрій з кількома входами та виходами, призначений для визначення одного з можливих вихідних станів в залежності від конфігурації вхідних двійкових сигналів. Цей компонент вважається важливим елементом автоматичного керування, обчислювально-електронних систем, систем зв'язку та комунікацій[1]. На рисунку 1.3 показано графічне зображення дешифратора.

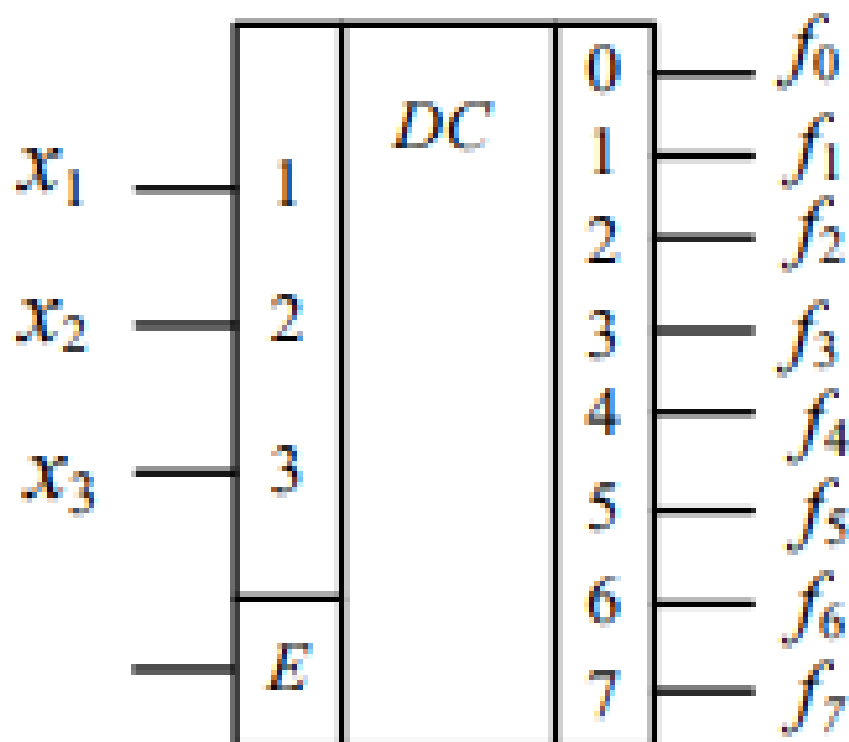


Рисунок 1.3 – Графічне зображення дешифратора

В таблиці 1.3 показано таблицю істиності станів дешифратора

Таблиця 1.3 – Таблиця істиності станів дешифратора

Входи			Вихід
X ₁	X ₂	X ₃	1
0	0	0	f ₀
0	0	1	f ₁
0	1	0	f ₂
0	1	1	f ₃
1	0	0	f ₄
1	0	1	f ₅
1	1	0	f ₆
1	1	1	f ₇

Тригер

Тригер - це цифровий пристрій, що може зберігати сигнали та видає два вихідні сигнали - прямий та інверсний, які працюють одночасно. Відмінність тригера полягає в його здатності залишатися в певному стані після переходу, навіть якщо вхідний сигнал зникає. У відмінну від комбінаційних схем, де сигнал на виході зберігається лише під час наявності вхідного сигналу, тригери відмічаються довільністю стану під час відсутності вхідного сигналу. Тригери є важливими компонентами цифрових схем і є основою для будівництва регістрів, лічильників, оперативної пам'яті, суматорів та інших цифрових пристроїв. Кожен стан тригера відповідає двійковому сигналу на його прямому і інверсному виходах: один стан відповідає 1 на прямому виході і 0 на інверсному виході, нульовий стан - інверсія першого, тобто 0 на прямому виході і 1 на інверсному виході[3].

На рисунку 1.4 наведено графічне представлення тригера з двома входами.

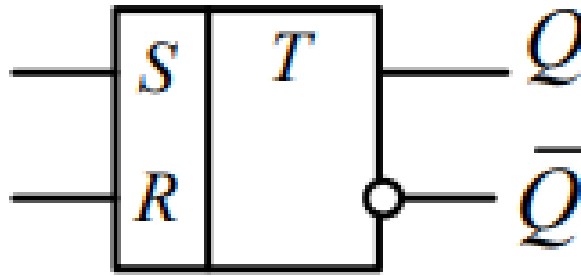


Рисунок 1.4 – Графічне зображення тригера

Регістр

Регістр є пристроєм, призначеним для тимчасового зберігання, трансформації і передачі інформації. Він складається з окремих тригерів, кожен з яких відповідає за запис та зберігання одного бітового розряду у двійковому числі. За своєю функціональністю регістри можуть бути класифіковані як накопичувальні, зсуву та перетворення [1].

1.4 Постановка завдання проектування

З урахуванням технічної літератури та використаних інформаційних джерел стосовно тематики кваліфікаційної роботи магістра «Адаптивна електронна система обчислення двійкових одиниць в інформаційних послідовностях» можна зробити висновок, що актуальною є задача підвищення швидкості обчислення кількості двійкових інформаційних одиниць для різноманітних електронних систем, таких як системи керування, збору даних, кодування та передачі інформації. Проектований пристрій має базуватися на простих операціях, щоб забезпечити ефективність його функціонування:

- порозрядного зрушення;
- віднімання двійкової одиниці;
- логічного множення;
- двійкового підсумовування;
- розбиття оброблюваних двійкових послідовностей.

Метою даної кваліфікаційної роботи магістра є розробка адаптивної електронної системи обчислення двійкових одиниць в інформаційних послідовностях, що характеризується зменшеними часовими витратами при функціонуванні.

Вимоги до технічних характеристик проектного пристрою для обчислення кількості одиниць формулюються наступним чином:

- довжина інформаційних послідовностей, що оброблюються, становить не менше 16 двійкових розрядів;

- максимальна кількість машинних тактів для отримання кількості одиниць не повинна перевищувати 8 тактів;

- тривалість одного машинного такту не повинна бути більше 0,1 мкс;

- застосування швидкодіючої елементної бази для практичної реалізації пристрою;

- Довжина оброблюваних інформаційних послідовностей повинна становити не менше 16 двійкових розрядів.

- Максимальна кількість машинних тактів, необхідних для визначення кількості одиниць, не повинна перевищувати 8 тактів.

- Тривалість кожного машинного такту не повинна перевищувати 0,1 мкс.

- Використання елементної бази високої швидкодії для практичної реалізації пристрою.

Потужність споживання електроенергії проектного пристрою визначення кількості одиниць становить не більше 2 Вт.

РОЗДІЛ 2 НАУКОВО-ДОСЛІДНА ЧАСТИНА

Результати синтезу алгоритмів підрахунку числа логічних одиниць показують, що з них, згідно з постановкою задачі, алгоритми є найбільш придатними для адресно-векторного кодування і декодування **1** і **2**. На відміну від алгоритму **2**, час роботи якого дорівнює постійній величині $\log_2 n$, алгоритм **1** Може виконуватися за різну кількість кроків.

Проаналізуємо часові характеристики алгоритму **1**: мінімальний, максимальний і середній час його спрацьовування.

Твердження 10. Нехай вхідні дані для алгоритму **1** є бінарними n -бітові слова $a_i, i=1, \dots, 2^n$, то мінімум V_{min} і максимум V_{max} кількість тактів його роботи без урахування етапів установки дорівнює: $V_{min}=0$ и $V_{max}=\alpha-1$.

Доказ. Дійсно, коли $Val a_i=0$ и $Val a_i=1$, Це визначається відразу кроками 3 і 4 установки відповідно. Потім, виходячи з результату їх дії, алгоритм **1** зупиняється. Отже $V_{min}=0$.

Максимальний час виконання розглянутого алгоритму визначається межею, встановленої кроком 7, а саме числом $k=\alpha-1$. Таким чином, $V_{max}=\alpha-1$, що і потрібно було довести

Робота алгоритму **1** Він не залежить від розташування логічних одиниць або нулів у двійковому слові, а залежить від їх кількості. У той же час число v_1 циклів роботи цього алгоритму дорівнюють:

$$v_1 = \begin{cases} k, & 0 \leq k < \alpha - 1 \\ \alpha - 1, & \alpha - 1 \leq k \leq n - \alpha + 1 \\ n - k, & n - \alpha + 1 < k \leq n \end{cases} \quad (2.1)$$

Очевидно, що в порівнянні з алгоритмом **2** алгоритм **1** слід використовувати, коли числа одиниць в двійкових словах не перевищують значення $\log_2 n$.

На рисунку 2.1 за допомогою графіків функцій $v_1=f(k)$, $v_2=\leq \log_2 n \leq$ та стрілок показані області значень k раціонального використання алгоритму **1** для

256- розрядних слів, де $0 \leq k \leq 16$. Аналогічний вигляд мають графіки та для $0 \leq n-k \leq 16$ нулів. При $16 < k < 240$ час v_1 значно більше часу v_2 . Очевидно, що для $k \leq \log_2 256 = 8$ або $n-k \leq \log_2 256 = 8$ швидше працює алгоритм **1**, в інших випадках ефективний алгоритм **2**.

У загальному випадку для того, щоб визначити, який алгоритм підрахунку застосувати при адресно-векторному кодуванні, необхідно знайти середнє значення V_1 числа тактів алгоритму **1** і порівняти його з числом $\log_2 n$. При цьому, якщо виконуватиметься нерівність $0 \leq V_1 \leq \log_2 n$,

то слід використовувати при стисканні алгоритм **1**, в іншому випадку – алгоритм **2**.

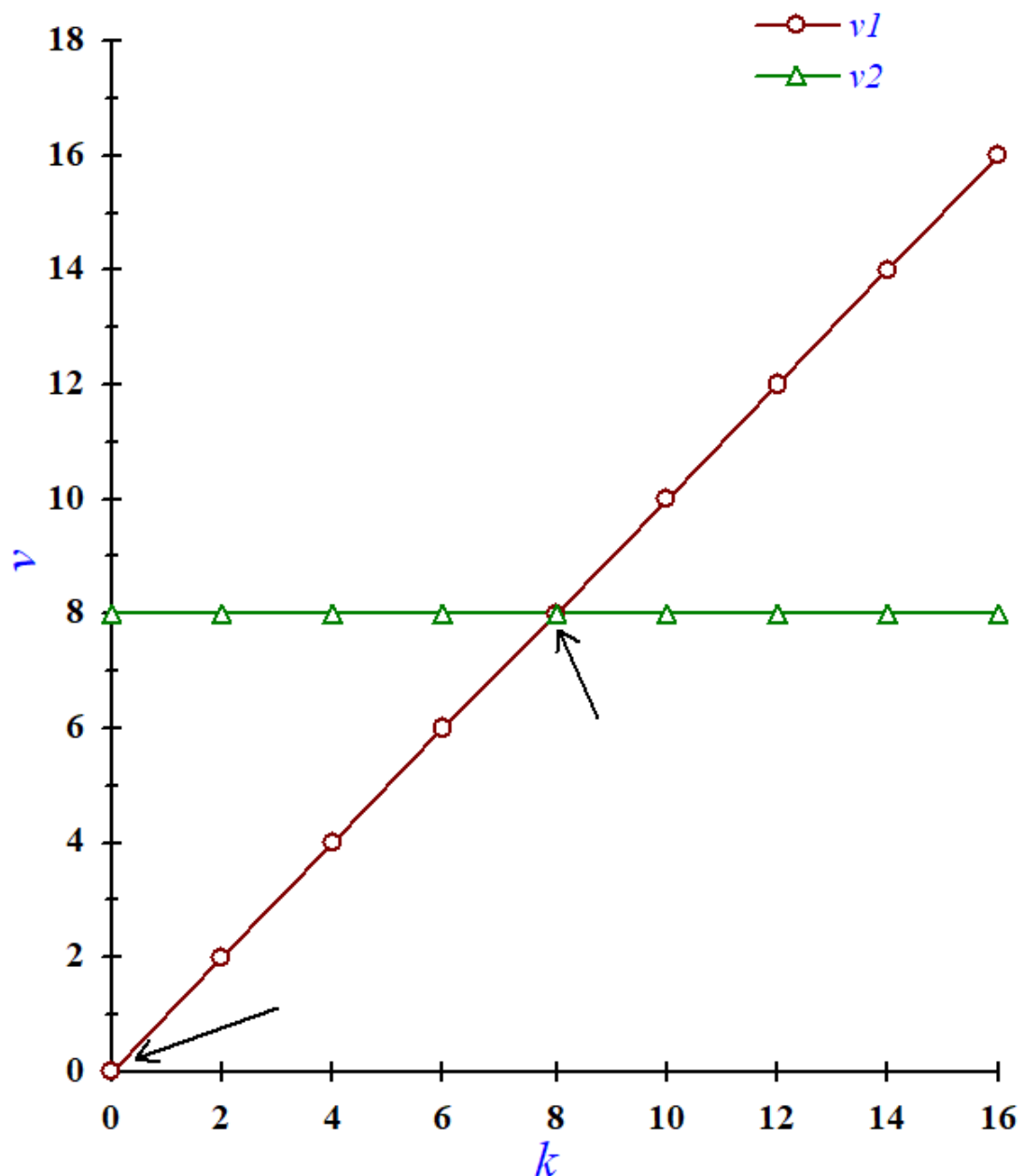


Рисунок 2.1 – Графіки числа v_1 кроків алгоритму **1** та числа v_2 кроків алгоритму **2** в залежності від k одиниць

З урахуванням незалежності появи слів один від одного, середнє значення V_1 визначається як математичне очікування випадкової величини v_1 , змінюється від 0 до $\alpha-1$ включно, з ймовірностями P_k :

$$V_1 = M[t] = \sum_{k=0}^n P_k t = \sum_{k=0}^{\alpha-2} P_k k + \sum_{k=\alpha-1}^{n-\alpha+1} P_k (\alpha-1) + \sum_{k=n-\alpha+2}^n P_k (n-k) \quad (2.2)$$

Використовуючи те, що $P_k = C_n^k p^k (1-p)^{n-k}$, на рисунку 2.1 в одній системі координат наведено графіки функцій $V_1=f(p)$ и $V_2=v_2=\log_2 n$ при $n=256$ залежно від p . Значення V_1 для вибірових 10 точок наведено у таблиці 2.1. Стрілками вказана область раціонального використання алгоритму 2. Из рисунку 2.1, таблиці 2.1 и формул (2.1, 2.2) следует, что в численном выражении данная область при $n=256$ має вигляд:

$$0,0298 < p < 0,969.$$

Таблиця 2.1 – Залежність часу V_1 від ймовірності p появи одиниць при $n=256$.

P	V_1
0,0099	2,53
0,0297	7,6
0,0584	15,21
0,099	25,04
0,188	31
0,505	31
0,901	25,04
0,941	15,21
0,970	7,6
0,990	2,53

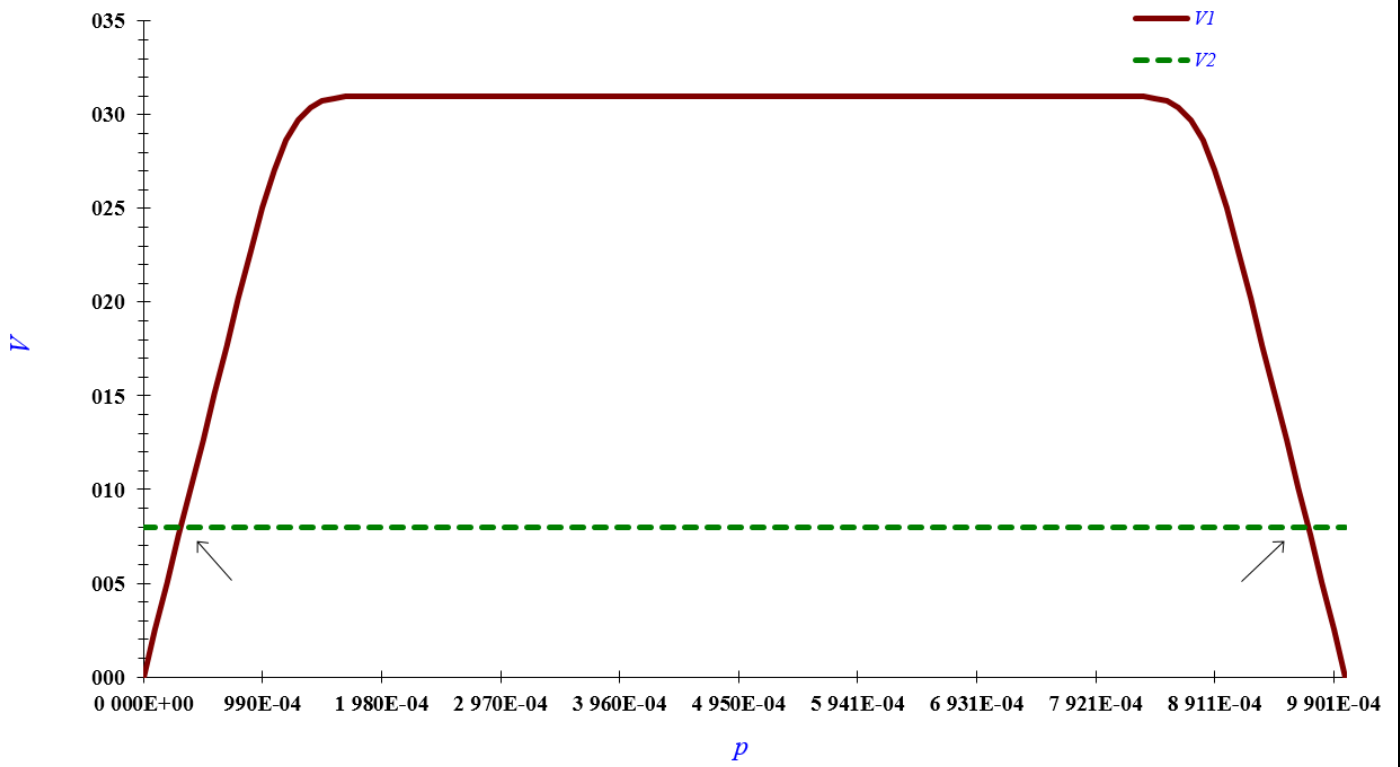


Рисунок 2.2 – Графіки середнього числа V_1 и V_2 кроків підрахунку числа одиниць для алгоритмів **1** та **2** відповідно, залежно від ймовірності p .

Зм.	Лист	№ документа	Підпис	Дата

РОЗДІЛ 3 РОЗРОБКА ЕЛЕКТРОННОЇ СИСТЕМИ З ВИКОРИСТАННЯМ ОТРИМАНИХ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

3.1 Розробка та обґрунтування алгоритму функціонування

Операцію підрахунку числа логічних одиниць необхідно розглядати з погляду адресно-векторного кодування, котрим важлива інформація такого роду: яким методом - адресним чи векторним - закодувати двійкове слово і якщо адресним, те яке одиниць є у ньому.

Тривіальний алгоритм підрахунку полягає в наступному: послідовно переглядати кожен розряд двійкового слова, і якщо він одиничний, змінної одиниць задавати відповідне збільшення. Для адресно-векторного кодування даний алгоритм можна модифікувати таким чином: ввести ще одну змінну – змінну нулів та одночасно з підрахунком одиниць проводити підрахунок нулів. Очевидно, роботу алгоритму можна перервати у разі перевищення або рівності обома змінними значення $\alpha-1$ нижньої умови стиснення. Перевагою такого алгоритму є те, що він дозволяє одночасно з обчисленням чисел одиниць та нулів визначати їхнє розташування. Але існує значний недолік, який применшує наведену гідність. Він полягає в необхідності витратити великий час (рівне n) для адресно-векторного стиску двійкових слів і особливо тих, чиї числа k одиниць задовольняють умов стиснення.

Розглянемо швидше обчислення числа k одиниць у припущенні, що пристрій, що реалізує алгоритм, здатний проводити арифметичні та логічні операції над розрядами двійкових слів.

Алгоритм 1, виконує обчислення числа k одиниць у двійковому слові $a_i \in A$ для адресно-векторного стиску з використанням арифметичних та логічних операцій:

1. [Початкове встановлення параметрів вихідного вектора].

Встановити $k \leftarrow 0$ (k містять поточне значення, одночасно рівне числу одиниць і нулів).

2. [Підготовка слова до одночасного підрахунку одиниць та нулів].

Встановити $s_k \leftarrow a_i, s_m \leftarrow \bar{a}_j$ (s_k и s_m - проміжні змінні).

3. Якщо $\text{Val } s_k = 0$, то перейти до кроку 8 ($\text{Val } s_k$ - число, двійковим записом якого є s_k).

4. Якщо $\text{Val } s_m = 0$, то перейти до наступного кроку. В іншому випадку відбувається перехід до кроку 6 ($\text{Val } s_m$ - число, двійковим записом якого є s_m).

5. Виконати $k \leftarrow n - k$ і перейти до кроку 8.

6. [Підрахунок логічних одиниць].

Виконати $s_k \leftarrow s_k \wedge (s_k - 1)$, $k \leftarrow k + 1$.

7. [Підрахунок логічних нулів].

Виконати $s_m \leftarrow s_m \wedge (s_m - 1)$.

8. [Перевірка на перевищення чи рівність k значення $\alpha - 1$].

Якщо $k < \alpha - 1$, то перейти до кроку 3. Інакше перейти до наступного кроку.

9. Зупинення алгоритму.

Блок-схема алгоритма **1** приведена на рисунку 3.1.

Операція $s_k \leftarrow s_k \wedge (s_k - 1)$ та $s_m \leftarrow s_m \wedge (s_m - 1)$ заміняють у двоичних слів s_k та s_m крайні справа одиниці банкрутом. Цикл алгоритму **1** триває доти, доки s_k або s_m не стануть рівними нулю або поки що змінна k не стане більше $\alpha - 1$ (одночасно з цим і число логічних нулей буде більше $\alpha - 1$).

Хоча алгоритм **1** працює, не знаючи а пріогі розташування двійкових символів, але це не можна назвати недоліком, оскільки цим стає можливим одночасний, в один такт, підрахунок одиниць і нулів, що значно підвищує його швидкодію в порівнянні з попереднім алгоритмом та алгоритмом, який вважав би лише одиниці. Очевидно, алгоритм **1**, використовує операції $s_k \leftarrow s_k \wedge (s_k - 1)$ та $s_m \leftarrow s_m \wedge (s_m - 1)$, ефективний для "розріджених" слів, тобто, двійкових слів з малим числом одиниць, і максимальний час його роботи буде меншим або рівним $\alpha - 1$.

Наступним алгоритмом обчислення числа одиниць є алгоритм, заснований на тому, що кілька операцій складання розрядів у коротких частинах проводяться шляхом однієї операції в довгих частинах двійкового слова. При цьому послідовно розбивають вихідне слово на рівні частини, представляючи їх як двійкові записи чисел одиниць, далі доповнюють отримані частини нулями до довжини вихідного слова і складають. Очевидно, що розбиття n -розрядного двійкового слова на рівні частини до довжини відрізка рівному l біту буде проводитися $\lceil \log_2 n \rceil$ разів. До апаратного забезпечення алгоритму висуваються такі самі вимоги, як і алгоритму **1**, але з додаванням того, що має існувати можливість зсуву слова на $1, 2, 4, 8, \dots$ розрядів.

Зм.	Лист	№ документа	Підпис	Дата

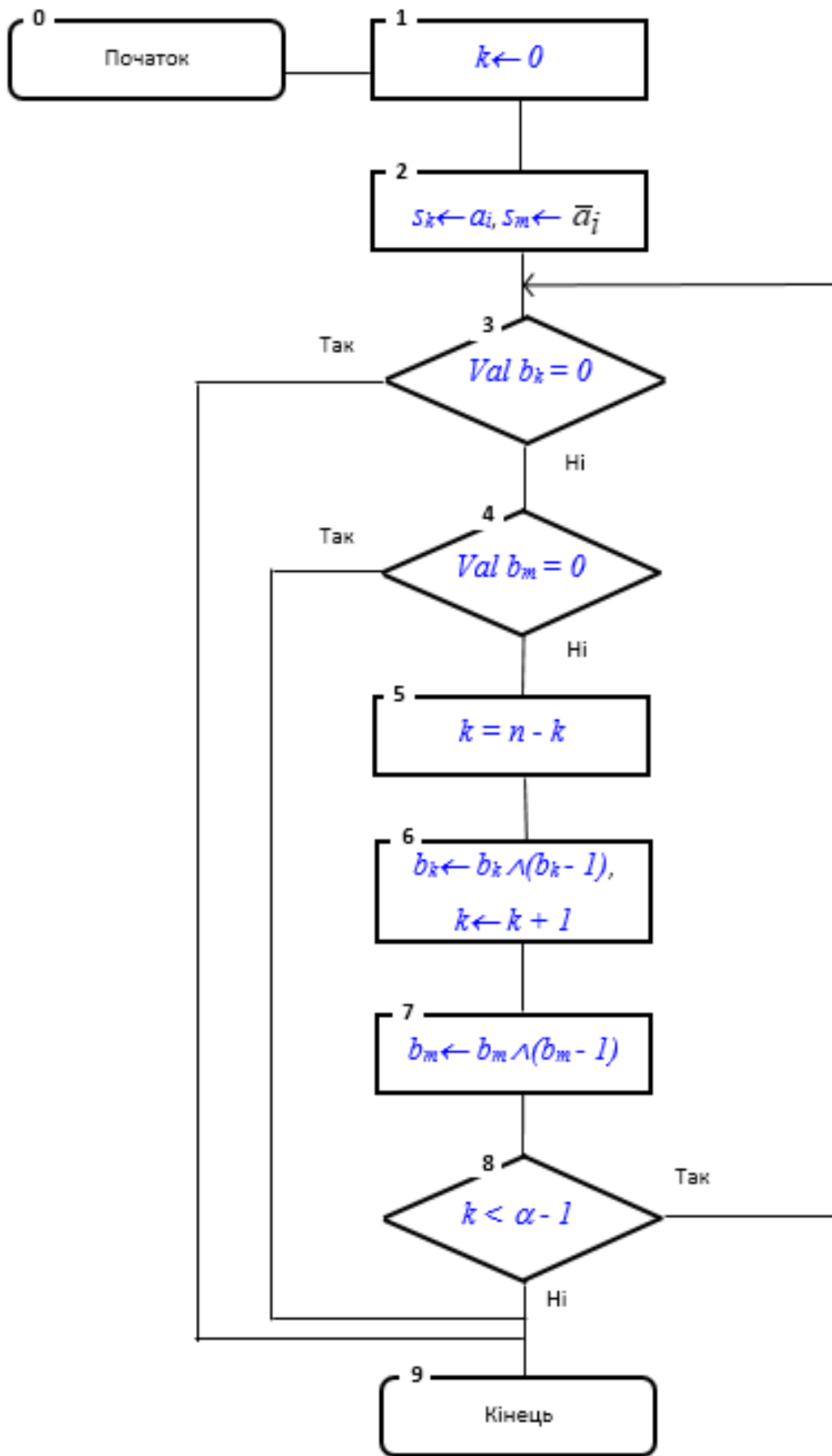


Рисунок 3.1 – Блок-схема алгоритму 1 підрахунку логічних одиниць

Алгоритм 2, здійснює обчислення числа k одиниць шляхом послідовного розбиття вихідного слова a_i на рівні частини з їх подальшим додаванням:

1. [Початкове встановлення параметрів двійкового слова a_i].

Встановити $s \leftarrow a_i$, $d \leftarrow 0$, $v \leftarrow 0$ ($s = s_{n-1} \dots s_h \dots s_0$ – змінна, що містить двійкові складові числа логічних одиниць, d – змінна, що вказує довжину зсуву та відстань вибірки розрядів двійкового слова s , v – змінна числа груп послідовно розташованих розрядів, що заповнюються нулями).

2. [Доповнення a_i нулями до довжини n , рівного цілого ступеня двійки, на випадок довільної розрядності слова].

Виконати $n \leftarrow \lceil a_i \rceil + q$ (q – число нульових розрядів, що дорівнює 0, Якщо довжина a_i не ділиться на 2, і не дорівнює 0 в іншому випадку).

3. [Цикл обчислення числа логічних одиниць: кроки 3-10].

Якщо $d > \log_2 n - 1$, то перейти до кроку 11. Інакше перейти до наступного кроку.

4. Встановити $r \leftarrow 2^d$ (r – довжина зсуву та міжрозрядна відстань вибірки двійкових біт слова s).

5. Встановити $x \leftarrow y \leftarrow s$.

6. [Визначення для слова s числа груп послідовно розташованих розрядів, що заповнюються нулями].

Встановити $v \leftarrow n/r$.

7. [Визначення для s адрес початку і кінця груп послідовно розташованих розрядів, що заповнюються нулями].

Встановити $h_0 \leftarrow 2^d, \dots, h_i \leftarrow 2^d \cdot (i+1), \dots, h_v \leftarrow 2^d \cdot (v+1)$ и $g_0 \leftarrow 2^d \cdot 2, \dots, g_i \leftarrow 2^d \cdot (i+2), \dots, g_v \leftarrow 2^d \cdot (v+2)$ (h_i и g_i – початкова та кінцева адреси розрядів S , заповнюваних логічними нулями).

8. [Заповнення нулями груп послідовно розташованих розрядів у слові X].

Встановити $l_x[h_0] \leftarrow l_x[h_0+1] \leftarrow \dots \leftarrow l_x[g_0] \leftarrow 0, \dots, l_x[h_i] \leftarrow l_x[h_i+1] \leftarrow \dots \leftarrow l_x[g_i] \leftarrow 0, \dots, l_x[h_v] \leftarrow l_x[h_v+1] \leftarrow \dots \leftarrow l_x[g_v] \leftarrow 0$ ($l_x[h_i]$ – вміст розряду з адресою h_i слова x).

9. [Заповнення нулями груп послідовно розташованих розрядів у слові Y].

Встановити $l_y[h_0-1] \leftarrow l_y[h_0] \leftarrow \dots \leftarrow l_y[g_0-1] \leftarrow 0, \dots, l_y[h_i-1] \leftarrow l_y[h_i] \leftarrow \dots \leftarrow l_y[g_i-1] \leftarrow 0, \dots, l_y[h_v-1] \leftarrow l_y[h_v] \leftarrow \dots \leftarrow l_y[g_v-1] \leftarrow 0$ и $Y \leftarrow \text{Shift}(r, y)$ ($l_y[h_i]$ – вміст розряду з адресою h_i слова y , $\text{Shift}(r, y)$ – функція зсуву вправо на r розрядів слова y).

10. Встановити $s \leftarrow x+y$, $d \leftarrow d+1$ і перейти до кроку 3.

11. Зупинення алгоритму.

Блок-схема алгоритму **2** наведена на рисунку 3.2.

Роботу цього алгоритму найкраще показати на прикладі.

Нехай вихідне двійкове 8- розрядне слово $a_i = 11001011$. Оскільки число $n=8$ є цілою ступенем двійки, то до нього дією кроку 2 додається число $q=0$. При початковому $d=0$ з слова $s=a_i$ кроком 7 виділяються непарні (і нульовий) розряди із заповненням інших нулями: $x=01000001$ а кроком 8 виділяються аналогічним чином парні розряди і зсуваються на 1 розряд вправо: $y=01000101$. Складування x та y по кроку 10: $s=x+y=10000110$, тим самим отримуємо: $s_1s_0=a_{i1}+a_{i0}=10$, $s_3s_2=a_{i3}+a_{i2}=01$, $s_5s_4=a_{i5}+a_{i4}=00$, $s_7s_6=a_{i7}+a_{i6}=10$, тобто одна сума $x+y$ здійснює чотири операції складання над розрядами. Далі, після виконання кроку 10 і вже при $d=1$ по дії кроків 7-9 виділяються пари розрядів s_1s_0 , s_5s_4 для x і пари s_3s_2 , s_7s_6 для y із заповненням інших нулями та наступним зрушенням y на 2 розряди вправо: $x=00000010$, $y=00100001$. Складування x та y : $s=x+y=00100011$, тим самим отримуємо: $s_3s_2s_1s_0=a_{i3}+a_{i2}+a_{i1}+a_{i0}$, а $s_7s_6s_5s_4=a_{i7}+a_{i6}+a_{i5}+a_{i4}$, то єсть, на втором цикле алгоритма **2** сума $x+y$ здійснює дві операції складання. При $d=2$ виділяються четвірки розрядів $s_3s_2s_1s_0$ для X і $s_7s_6s_5s_4$ для y із заповненням інших нулями та наступним зрушенням y уже на 4 розряди: $x=00000011$, $y=00000010$. Складування x та y : $s=x+y=00000101$, тим самим отримуємо: $s_7s_6s_5s_4s_3s_2s_1s_0=a_{i7}+a_{i6}+a_{i5}+a_{i4}+a_{i3}+a_{i2}+a_{i1}+a_{i0}$, що і є кінцевим результатом. Тут $x+y$ виконує одну операцію додавання над розрядами, в результаті якої виявляються врахованими всі розряди двійкового слова a_i . Таким чином, для заданої комбінації 11001011 виконані всі 8 складання вмісту розрядів і результуюче число $k=5$.

Алгоритм **2** має дуже важливу перевагу: повторенням циклу основних операцій фіксована кількість разів - $\lceil \log_2 n \rceil$, яке не залежить від виду a_i . До його нестачі

Зм.	Лист	№ документа	Підпис	Дата

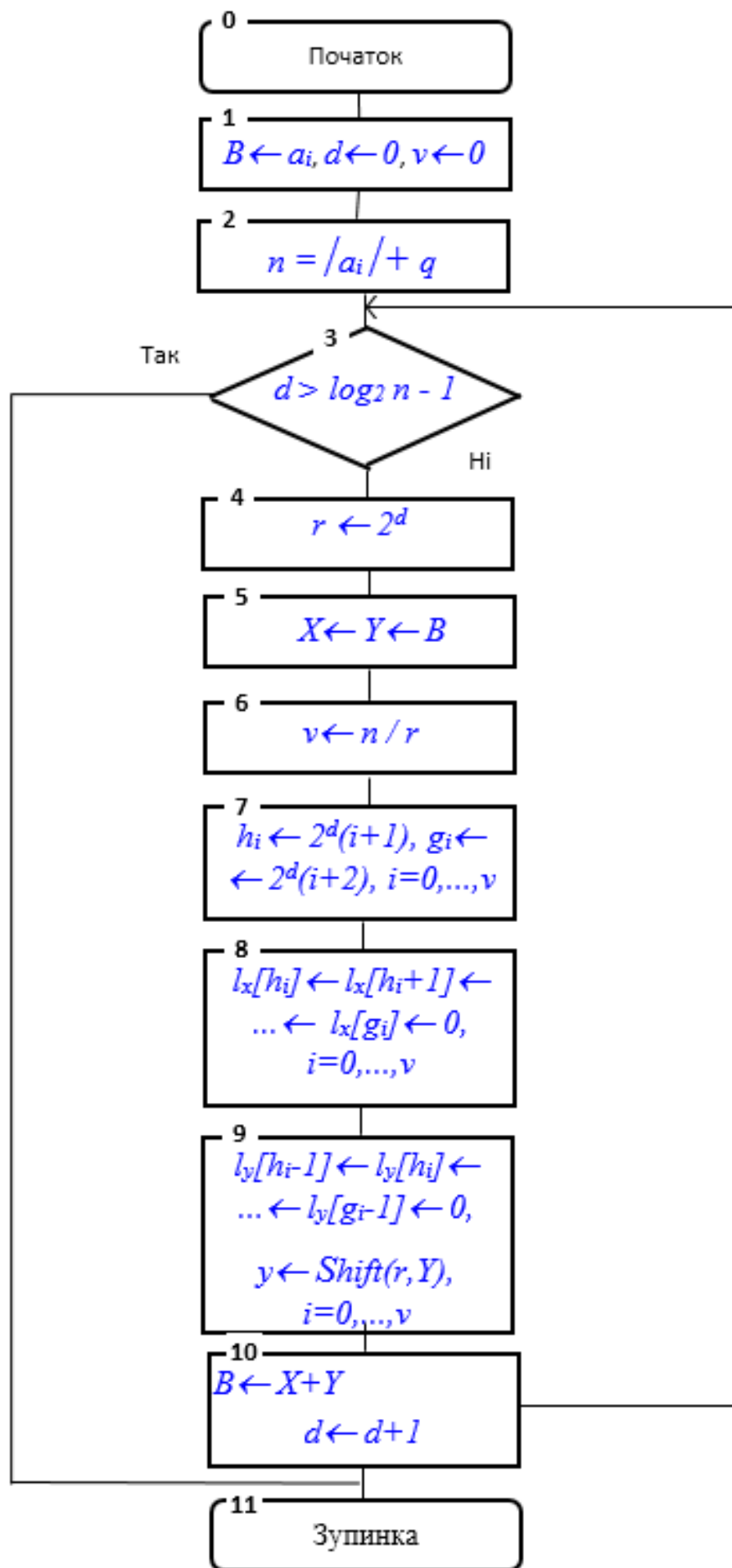


Рисунок 3.1 – Блок-схема алгоритму 2 підрахунку логічних одиниць

можна віднести підвищені в порівнянні з алгоритмом 1 апаратурні вимоги: забезпечення зсуву двійкових слів на 1, 2, 4, 8, ... розрядів.

1. Таким чином, відповідно до постановки задачі та враховуючи, що підрахунок числа логічних одиниць є однією з основних операцій для адресно-векторного кодування, щодо представлених алгоритмів можна укласти наступне.

2. Для забезпечення швидкодії адресно-векторного кодування слід використовувати алгоритми **1** та **2**, які для двійкових n - розрядних слів обчислюють число одиниць за час набагато менше, ніж n .

3. Технічна (або програмна реалізація) алгоритмів **1** та **2** має передбачати виконання арифметичних та логічних порозрядних операцій та додатково для алгоритму **2** виконання зсувних операцій.

4. Щоб додатково зменшити час підрахунку двійкових одиниць при адресно-векторному кодуванні, можливо, використовувати обидва алгоритми **1** та **2**, які б підключалися залежно від розподілу одиниць у двійковому слові.

Згідно постановки завдання проектування пристрій повинен перетворювати вхідну двійкову послідовність в 16-ти розрядну паралельне двійкове об'єднання, тому на початковому етапі маємо виконувати цей процес

Для об'єднання знадобиться блок переходу між режимами роботи пристрою, тому на наступному кроці двійкова послідовність буде перевірятись на насиченість одиницями.

Зм.	Лист	№ документа	Підпис	Дата

ЕліТ 8.171.00.10.445

Лист

33

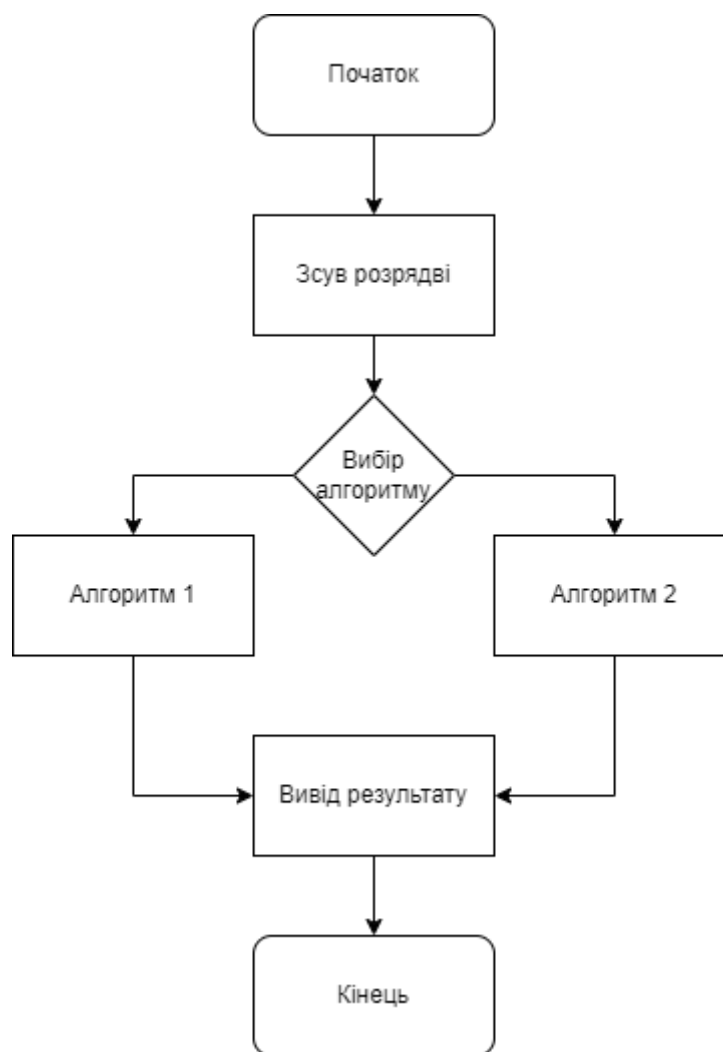


Рисунок 3.3 – Схема алгоритму функціону електронної системи.

3.2 Розробка структурної схеми проектованої системи

Згідно з алгоритмом функціонування, давайте створимо структурну схему, починаючи з обраного методу обчислення кількості одиниць в двійкових послідовностях. В розгляді будемо використовувати цікавий метод, що базується на алгоритмі $V = V \wedge (V - 1)$, де V – двійкова комбінація, а \wedge позначає логічну операцію І, яка операцію проводить паралельно з кожним розрядом між відповідними бітами V та $(V - 1)$. Цей метод широко використовується у пристроях, які здатні виконувати арифметичні та логічні операції над розрядами двійкових послідовностей.

Опишемо та побудуємо структурну схему пристрою, який використовує цей метод для обчислення кількості одиниць у двійковій послідовності. Це важливий аспект у пристроях, які опрацьовують дані у двійковій формі, і надає можливість ефективного використання арифметичних та логічних операцій.

Враховуючи, що пристрій приймає двійкові послідовності і потрібно обробляти їхні розряди, першим кроком буде створення блока для розподілу розрядів. Цей блок буде відповідати за порозрядне зрушення вхідного сигналу на певну кількість розрядів, що не менше 16.

Після розподілу двійкового коду на розряди, наступний етап - блок виявлення одиниць. Відповідно до вказаного алгоритму, для підрахунку одиниць в розподіленому двійковому наборі потрібно виконати певні обчислення. Спочатку віднімається одиниця ($B - 1$), а потім отриманий результат складається з вихідним числом B ($B \& (B - 1)$).

Ця послідовність кроків дозволяє ефективно обробляти вхідні двійкові дані, розподіляти їх на розряди та виявляти наявність одиниць, використовуючи вказані обчислення.

Далі в процесі розробки будуть введені блоки віднімання та складання інформаційних двійкових наборів. Ці блоки необхідні для коректного розрахунку кількості одиниць у заданому наборі. Важливо врахувати, що ефективність підрахунку визначатиметься кількістю обчислень.

На наступному етапі слід створити блок підрахунку одиниць. Цей блок визначатиме кількість одиниць у заданому двійковому наборі. Для цього буде використано вже введені блоки віднімання та складання.

Однак створені блоки не зможуть працювати коректно самостійно один від одного. Тому необхідно введення блоку керування, який буде відправляти сигнали дозволу іншим блокам для синхронізації їхньої роботи. Зазначте, що блок керування не може діяти самостійно, тому вводиться блок генератора імпульсів, який буде генерувати імпульсні сигнали, необхідні для активації блоку керування.

Об'єднавши всі ці блоки, отримаємо структурну схему пристрою, яка буде забезпечувати правильну обробку та аналіз вхідних двійкових сигналів. На рисунку 3.4 зображено структурну схему за алгоритмом 1.

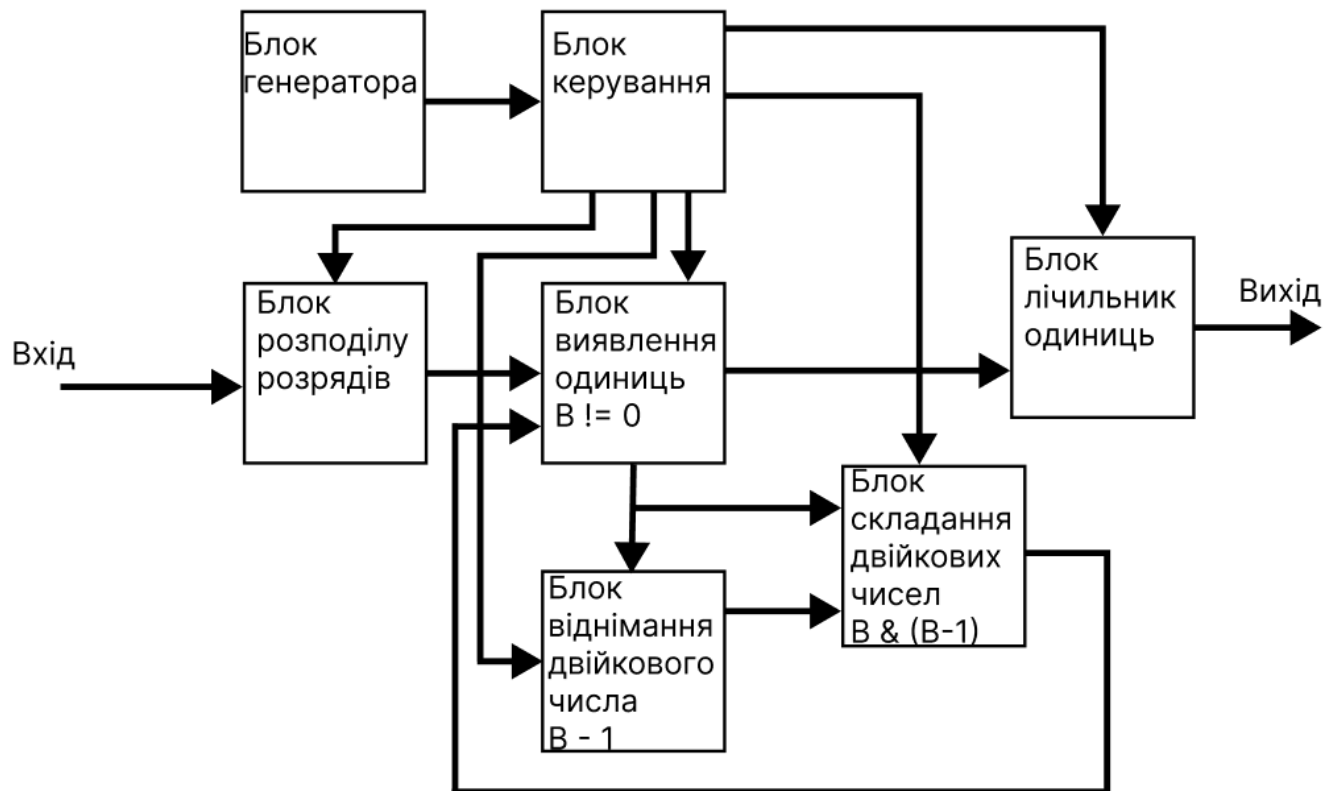


Рисунок 3.4 – Структурна схема алгоритму 1

Побудуємо структурну схему за алгоритмом 2. Блок генератора, блок керування та блок розподілу розрядів залишаються незмінними компонентами схеми. Згідно алгоритму необхідно створити блок зсуву розрядів, який буде розподіляти в необхідному порядку вхідні інформаційні двійкові комбінації необхідним чином. Також згідно алгоритму необхідно створити блок додавання двійкових чисел.

На рисунку 3.5 зображено структурну схему за алгоритмом 2.

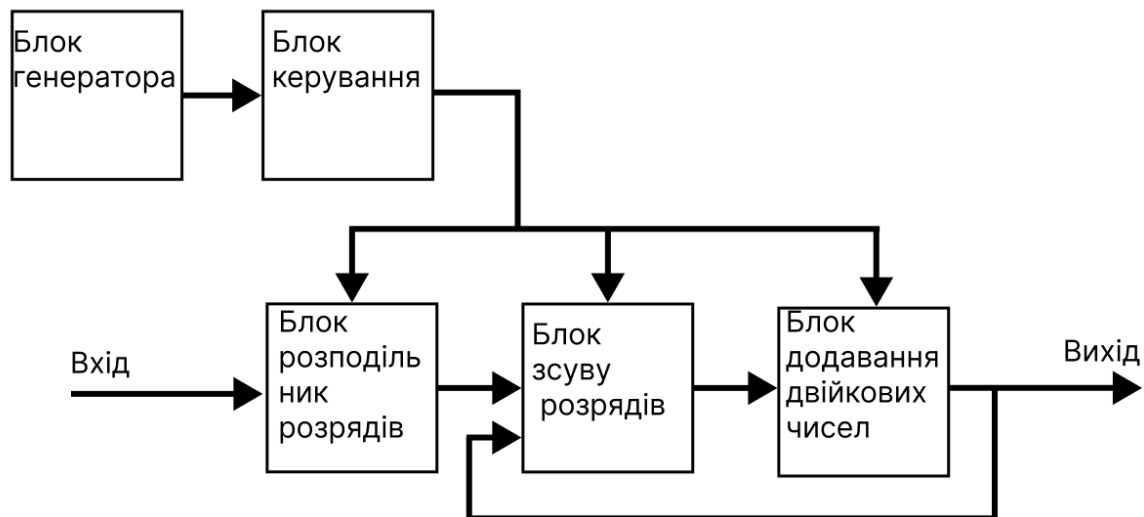


Рисунок 3.5 – Структурна схема алгоритму 2

Розглянувши вищезгаданий спосіб та зіставивши його параметри з постановкою завдання проектування можемо зробити висновок що буде доцільним об'єднати два методи та від кожного взяти їх переваги. Об'єднання не вплине на апаратні затрати пристрою так як між ними є багато спільних блоків які немає необхідності дублювати. Додамо блок пам'яті для тимчасового зберігання двійкової послідовності поки проходять обчислення.

Для перемикання між режимами роботи пристрою додамо блок вибору режиму, який буде визначати більшу або меншу кількість одиниць певного заданого значення. На рисунку 3.6 зображено структурну схему системи обчислення одиниць.

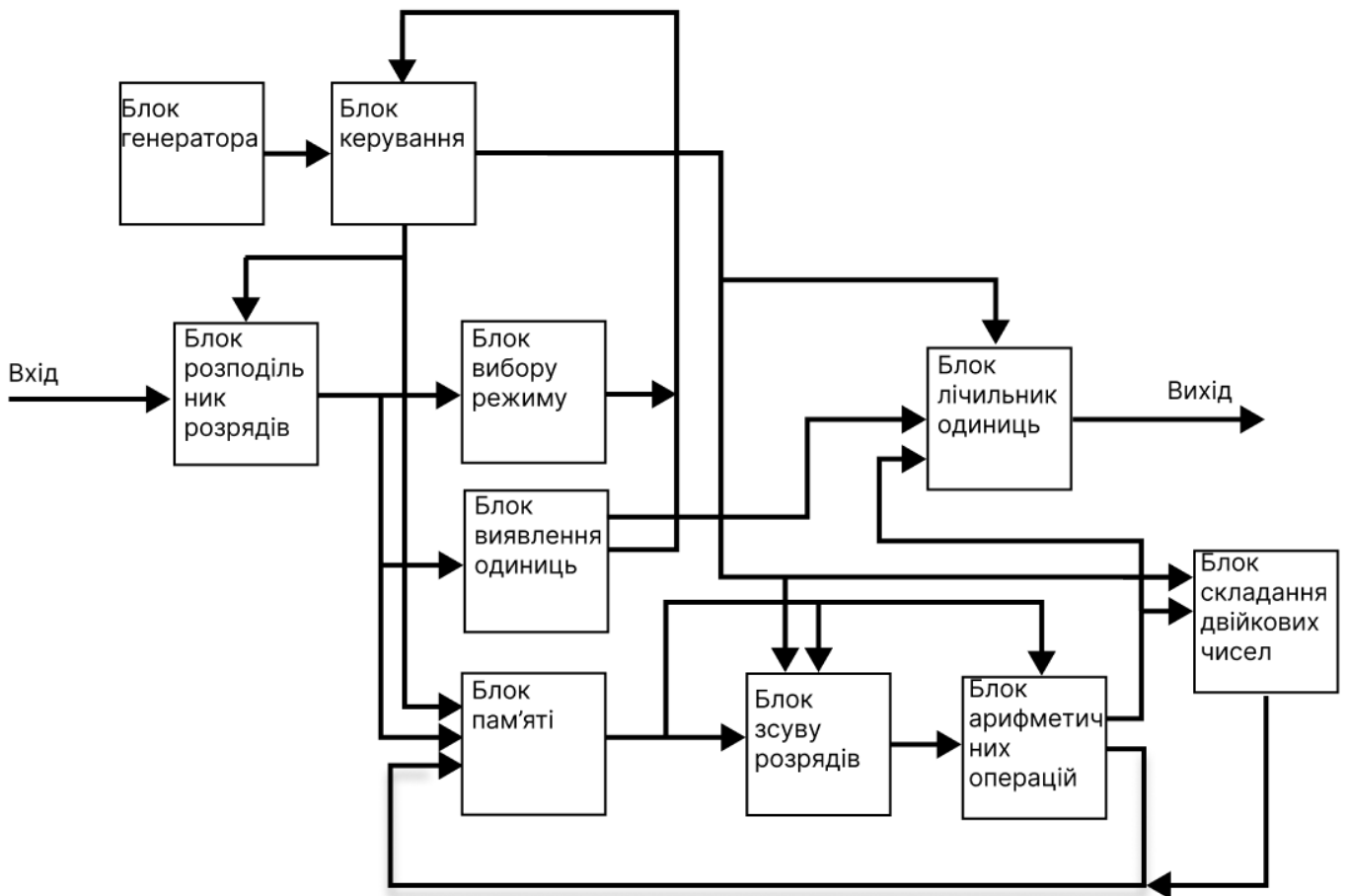


Рисунок 3.6 – Структурна схема системи обчислення одиниць

3.3 Розробка електричної функціональної схеми

- Блок керування

Блок керування — це функціональний компонент системи, який відповідає за координацію та управління роботою інших компонентів чи підсистем. Цей блок може включати в себе алгоритми, логіку та інші елементи, які забезпечують взаємодію між різними частинами системи для досягнення загальних цілей. Блок керування відіграє важливу роль у забезпеченні координації і управлінні компонентами системи для досягнення встановлених цілей.

- Блок розподілу розрядів

Блок розподілу розрядів – це електронний пристрій або логічний блок, який виконує конвертацію інформації, поданої у послідовному форматі, в паралельний формат. Блок отримує послідовний потік бітів, який представляє інформацію, яку

Зм.	Лист	№ документа	Підпис	Дата

необхідно перетворити. Для правильного розпізнавання та розбору вхідних даних блок може використовувати сигнали синхронізації для визначення моменту зміни біту. Після обробки вхідних даних блок генерує паралельний вихід, що представляє ту саму інформацію, але в розподіленому по паралельних каналах форматі.

- Блок вибору режиму

Блок вибору режиму — це компонент в системі, який відповідає за вибір та управління різними режимами роботи пристрою. Цей блок може бути використаний для переключення між різними алгоритмами роботи пристрою залежно вхідних сигналів які оброблюються. Після вибору режиму блок генерує сигнали, які визначають, як пристрій повинен функціонувати в обраному режимі.

- Блок виявлення одиниць

Блок виявлення одиниць призначений для виявлення одиниць у вхідній двійковій послідовності для подальшого роботи пристрою який без них не знатин працювативизначення кількості двійкових одиниць. Процес виявлення одиниць може бути реалізований на апаратному рівні, наприклад, за допомогою логічних елементів.

- Блок пам'яті

Блок пам'яті призначений для короткочасного зберігання вхідної двійкової послідовності поки проходять необхідні обчислення. Для зберігання може бути використаний регістр пам'яті.

- Блок зсуву розрядів

Блок зсуву розрядів призначений для зсуву вхідних двійкових послідовностей в необхідному порядку згідно алгоритму 2 роботи пристрою.

- Блок арифметичних операцій

Блок арифметичних операцій виконує функцію додавання та віднімання вхідних двійкових послідовностей згідно алгоритму 1 та 2 роботи пристрою

- Блок складання двійкових чисел

Блок арифметичних операцій виконує функцію складання вхідних двійкових послідовностей згідно алгоритму 1 та 2 рооти пристрою

- Блок генератора

Блок генератора імпульсів — це компонент електронної системи, який відповідає за створення імпульсів або сигналів з певною частотою, формою та амплітудою. Генератори імпульсів широко використовуються в електроніці для таких завдань, як синхронізація, генерація тактових сигналів. В системі що розробляється генератор імпульсів необхідний перш за все для роботи блоку керування.

На рисунку 3.7 зображено функціональну схему проектованої системи.

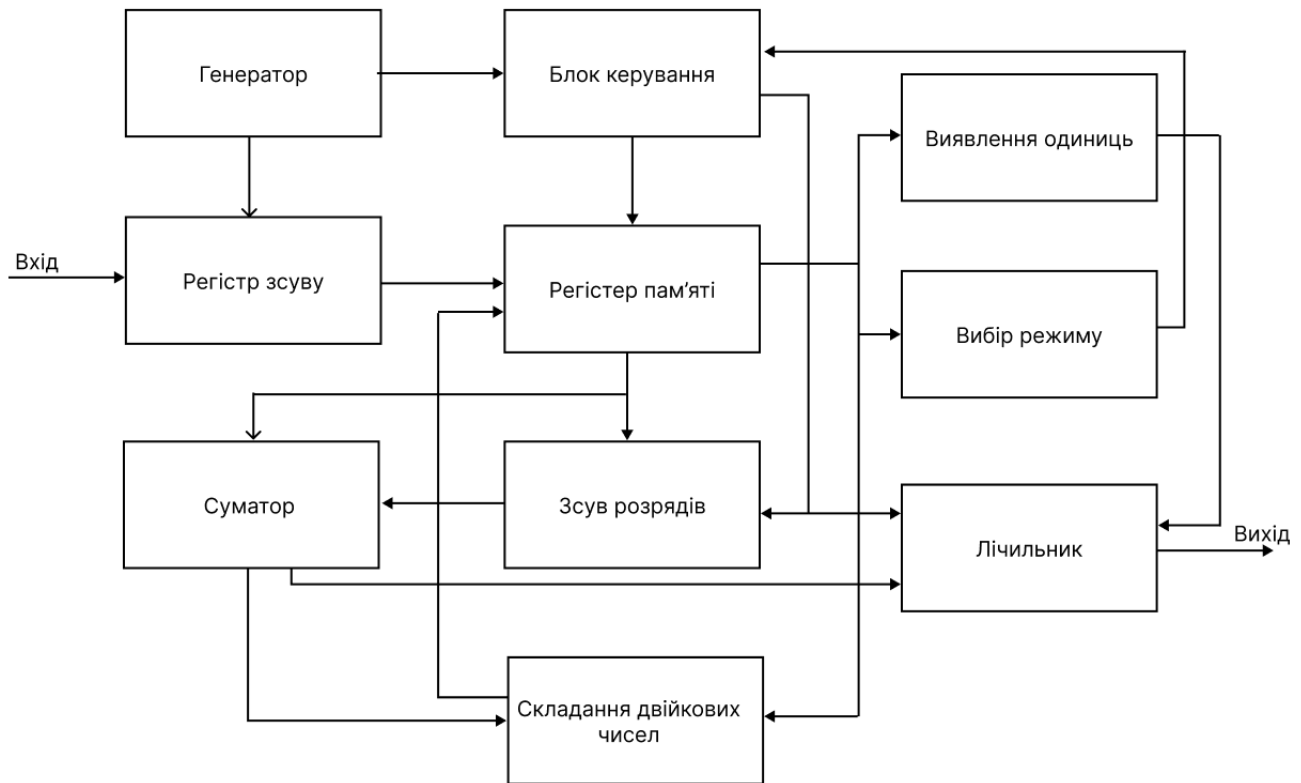


Рисунок 3.7 – Функціональна схема проектованої системи

3.4 Вибір елементної бази

Основним завданням проекту є розробка високошвидкісного пристрою для ефективного підрахунку кількості одиниць в двійкових послідовностях. Це вимагає вибору оптимізованого та продуктивного апаратного забезпечення, яке забезпечить швидку обробку обсягів даних.

Блоки електронних пристроїв включають елементи у формі інтегральних мікросхем, таких як дешифратори, лічильники, розподільники імпульсів і інші. Запобіганням підвищенню захищеності всієї електронної системи слід розглядати підвищення захищеності та надійності використовуваних мікросхем як першочерговий крок. На щастя, вартість цих мікросхем за сучасних технологій виробництва практично не збільшується значно, а швидкість роботи залишається на високому рівні, тоді як надійність збільшується в кілька разів. Для досягнення цієї мети слід розглядати оптимізації на різних рівнях, таких як апаратні засоби.

В якості суматора використаємо мікросхему К155ИМ3 (рис. 3.7), яка містить чотирьохрозрядний повний суматор. Мікросхема містить 781 інтегральний елемент та виконана в корпусі 201.16-6 (рис. 3.8) масою не більше 2,5 г

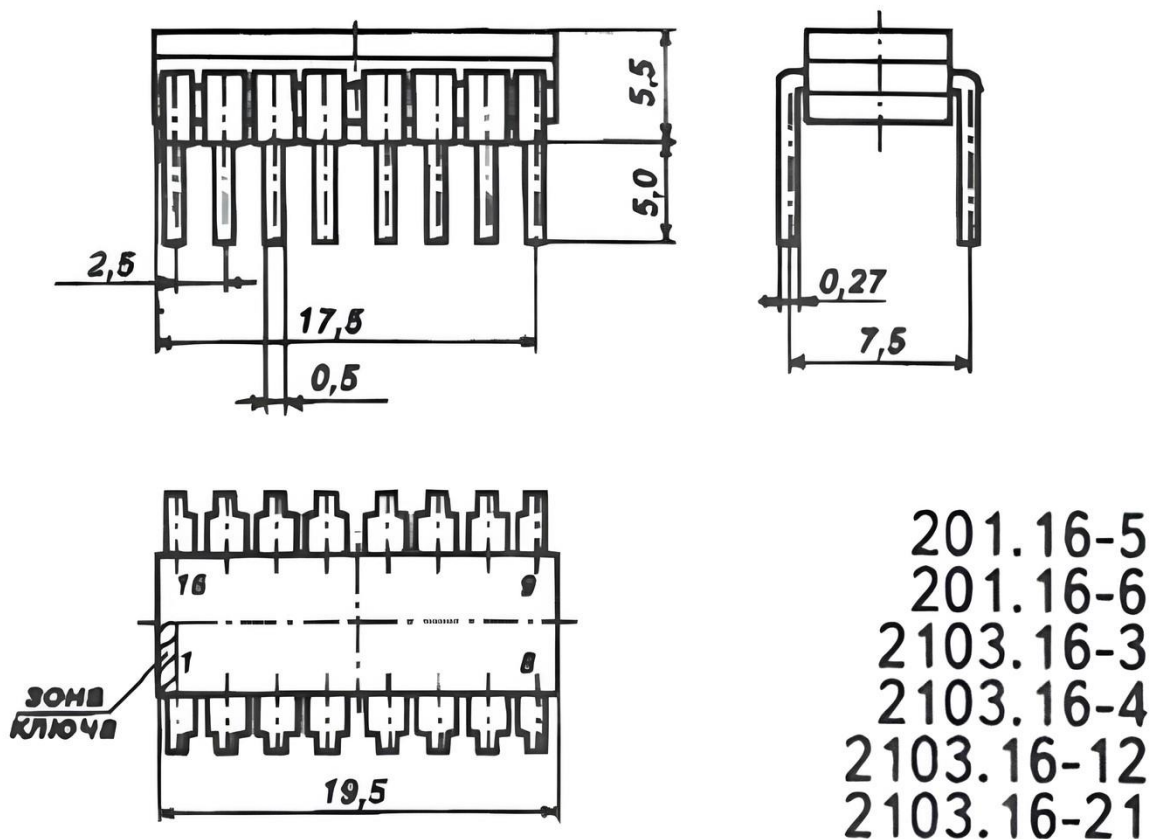


Рисунок 3.8 – Корпус мікросхеми К155ИМ3

Зм.	Лист	№ документа	Підпис	Дата

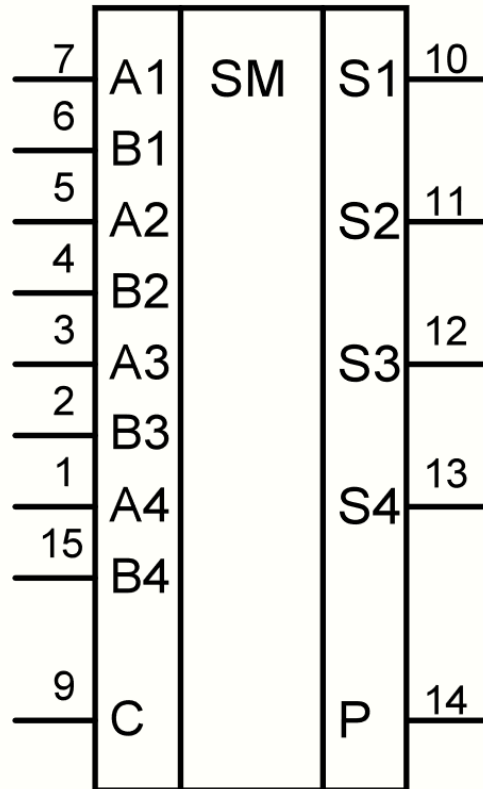


Рисунок 3.9 – Умовне графічне позначення мікросхеми К155ИМ3

Призначення виводів мікросхеми наведено в таблиці 3.1.

Таблиця 3.1 – Призначення виводів мікросхеми К155ИМ3

Вивід №	Призначення	Вивід №	Призначення
1	Вхід доданка А4	8	Загальний
2	Вхід доданка В3	9	Вхід переносу С
3	Вхід доданка А3	10	Вихід суми S1
4	Вхід доданка В2	11	Вихід суми S2
5	Вхід доданка А2	12	Вихід суми S3
6	Вхід доданка В1	13	Вихід суми S4
7	Вхід доданка А1	14	Вихід переносу четвертого розряду Р
15	Вхід доданка А4	16	Живлення

Електричні параметри мікросхеми К155ИМ3 наведені в таблиці 3.2

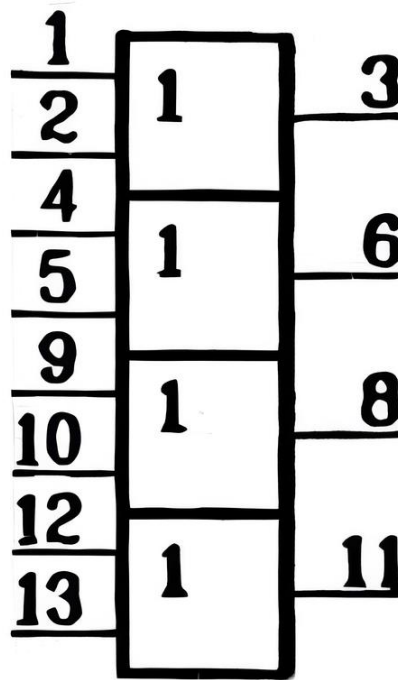


Рисунок 3.10 – Умовне графічне позначення КР1533ЛЛ1

Призначення виводів мікросхеми: 1, 2, 4, 5, 9, 10, 12, 13 – входи; 3, 6, 8, 11 – виходи; 7 – загальний; 14 – живлення.

Таблиця 3.3 – Електричні параметри мікросхеми КР1533ЛЛ1

Номінальна напруга живлення	$5 \text{ В} \pm 5\%$
Вихідна напруга низького рівня	$\leq 0,4 \text{ В}$
Вихідна напруга високого рівня	$\geq 2,4 \text{ В}$
Вхідний струм низького рівня	$\leq 0,1 \text{ мА}$
Вхідний струм високого рівня	$\leq 0,20 \text{ мкА}$
Струм витікання на виході	$\leq 0,2 \text{ мА}$
Струм споживання при низькому рівні вихідної напруги	$\leq 4,9 \text{ мА}$
Струм споживання при високому рівні вихідної напруги	$\leq 4 \text{ мА}$
Час затримки поширення сигналу при вмиканні	$\leq 12 \text{ нс}$
Час затримки поширення сигналу при вимиканні	$\leq 14 \text{ нс}$
Ємність виходу	$\leq 5 \text{ пФ}$

Зм.	Лист	№ документа	Підпис	Дата

ЕліТ 8.171.00.10.445

Лист

44

Для зменшення об'ємних характеристик системи та раціонального використання простору використаємо мікросхему К155ЛІІ5 (зображена на рис. 3.11). Ця мікросхема володіє двома логічними елементами "2І", об'єднаними в одному корпусі.

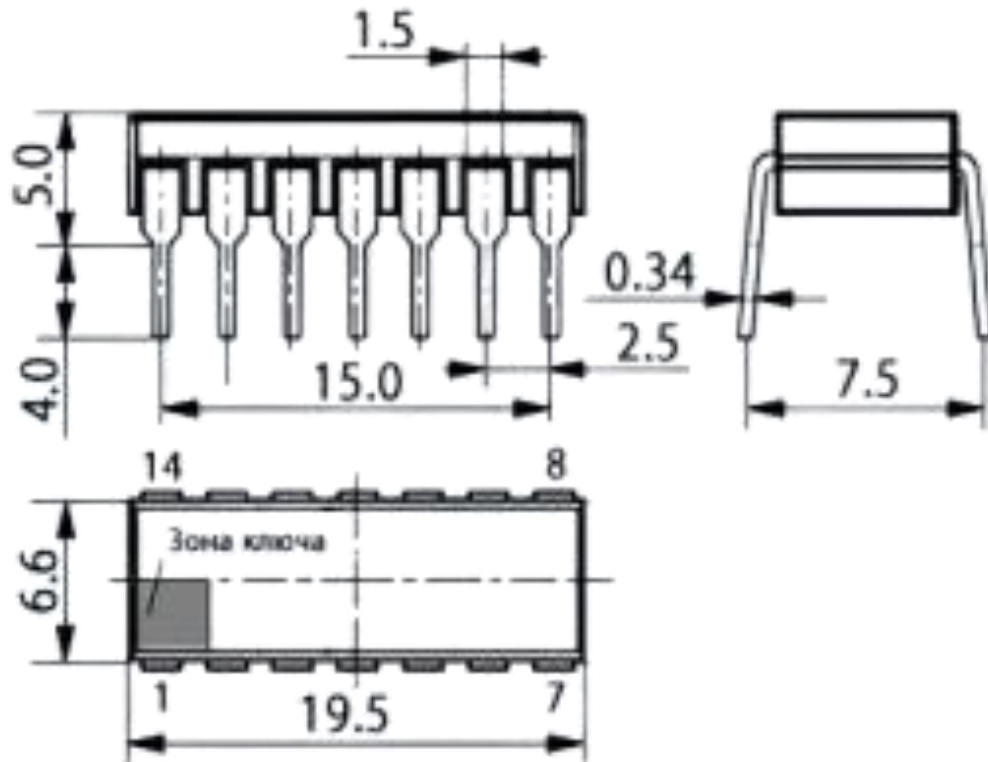


Рисунок 3.10 – Корпус мікросхеми К155ЛІІ5

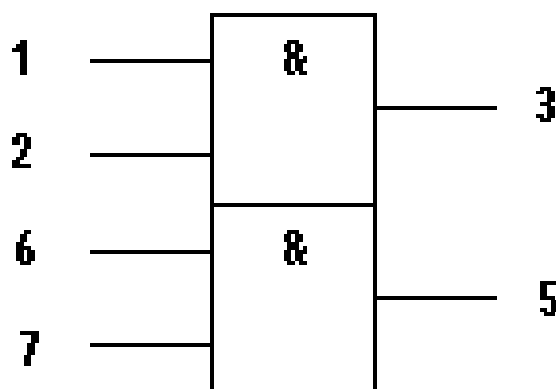


Рисунок 3.11 – Умовне графічне позначення К155ЛІІ5

Призначення виходів: 1, 2, 6, 7 – входи, 3, 5 – виходи.

Зм.	Лист	№ документа	Підпис	Дата

ЕліТ 8.171.00.10.445

Лист

45

Таблиця 3.4 – Параметри мікросхеми К155ЛІІ5

Номінальна напруга живлення	$5 \text{ В} \pm 5\%$
Вихідна напруга низького рівня	$\leq 0,5 \text{ В}$
Вихідна напруга високого рівня	$\geq 2,5 \text{ В}$
Вхідний струм низького рівня	$\leq -1,6 \text{ мкА}$
Вхідний струм високого рівня	$\leq 0,04 \text{ мА}$
Струм витікання на виході	$\leq 1 \text{ мА}$
Струм споживання при низькому рівні вихідної напруги	$\leq 65 \text{ мА}$
Струм споживання при високому рівні вихідної напруги	$\leq 11 \text{ мА}$
Час затримки поширення сигналу при вмиканні	$\leq 25 \text{ нс}$
Час затримки поширення сигналу при вимиканні	$\leq 25 \text{ нс}$
Ємність виходу	$\leq 4 \text{ пФ}$

Для зменшення об'ємних характеристик системи та раціонального використання простору використаємо мікросхему КР1533ЛЕ1 (зображена на рис. 3.12). Ця мікросхема володіє чотирма логічними елементами "2АБО -НІ", об'єднаними в одному корпусі.

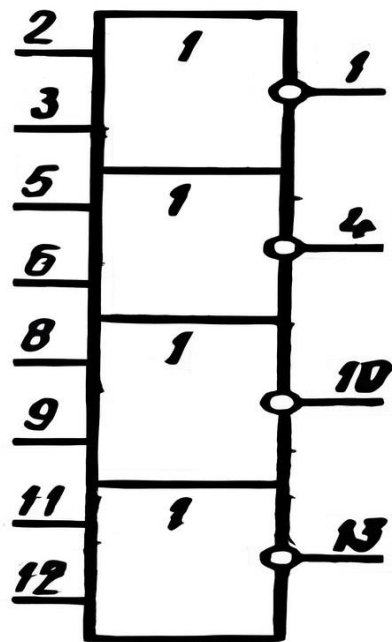


Рисунок 3.12 – Умовне графічне позначення КР1533ЛЕ1

Зм.	Лист	№ документа	Підпис	Дата

ЕліТ 8.171.00.10.445

Лист

46

Призначення виходів: 1, 4, 10, 13 – виходи, 2, 3, 5, 8, 9, 11, 12 – входи; 7 – загальний; 14 – живлення.

В таблиці 3.5 наведено параметри мікросхеми КР1533ЛЕ1.

Таблиця 3.5 – Параметри мікросхеми КР1533ЛЕ1

Номинальна напруга живлення	$5 \text{ В} \pm 5\%$
Вихідна напруга низького рівня	$\leq 0,5 \text{ В}$
Вихідна напруга високого рівня	$\geq 2,5 \text{ В}$
Вхідний струм низького рівня	$\leq 0,1 \text{ мА}$
Вхідний струм високого рівня	$\leq 0,20 \text{ мкА}$
Струм витікання на виході	$\leq 0,2 \text{ мА}$
Струм споживання при низькому рівні вихідної напруги	$\leq 4 \text{ мА}$
Струм споживання при високому рівні вихідної напруги	$\leq 2,2 \text{ мА}$
Час затримки поширення сигналу при вмиканні	$\leq 10 \text{ нс}$
Час затримки поширення сигналу при вимиканні	$\leq 12 \text{ нс}$
Ємність виходу	$\leq 4 \text{ пФ}$

В якості дешифратора використовуємо мікросхему КР1561ИД6 (рис. 3.12), яка містить два двовходових дешифратора. Мікросхема виконана в DIP-корпусі, містить 180 інтегральних елементів. (рис. 3.14) масою не більше 2,5 г

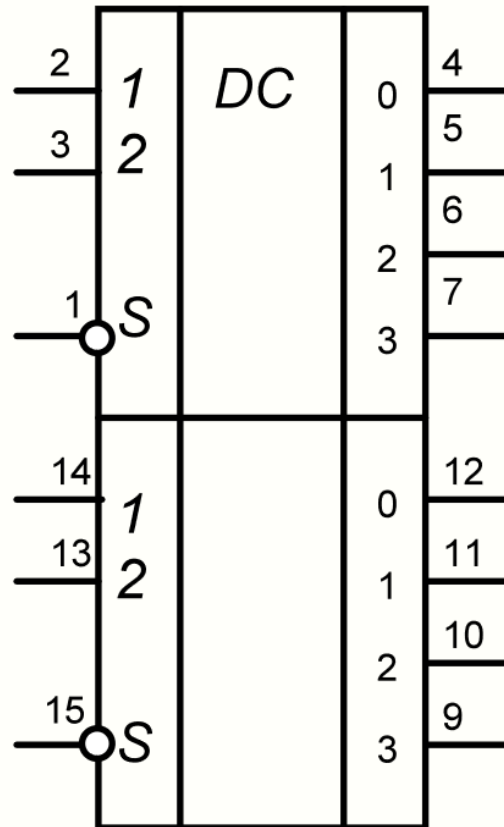


Рисунок 3.13 – Умовне графічне позначення КР1561ИД6

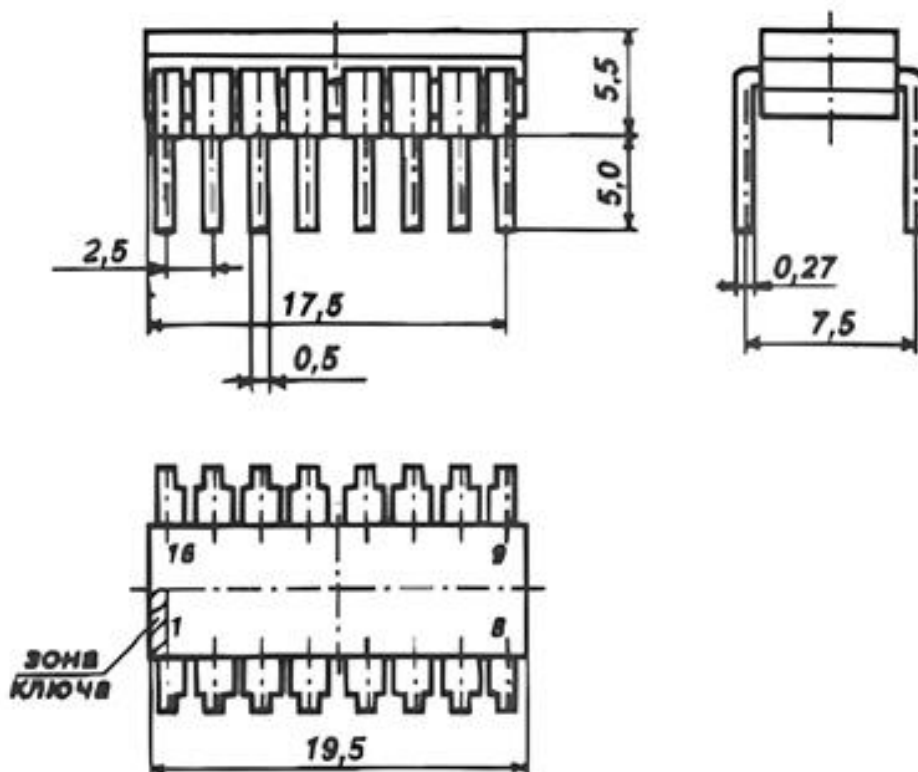


Рисунок 3.14 – Корпус мікросхеми КР1561ИД6

Зм.	Лист	№ документа	Підпис	Дата

ЕліТ 8.171.00.10.445

Лист

48

В таблиці 3.6 наведено призначення виводів мікросхеми КР1561ИД6.

Таблиця 3.6 – Призначення виводів мікросхеми КР1561ИД6

Вивід №	Призначення	Вивід №	Призначення
1	Дозвіл 1	8	Загальний
2	Вхід 1	9	Вихід 5
3	Вхід 2	10	Вихід 6
4	Вихід 1	11	Вихід 7
5	Вихід 2	12	Вихід 8
6	Вихід 3	13	Вхід 4
7	Вихід 4	14	Вхід 4
15	Дозвіл 2	16	Живлення

Електричні параметри мікросхеми КР1561ИД6 наведені в таблиці 3.14

Таблиця 3.7 – Електричні параметри мікросхеми КР1561ИД6

Номінальна напруга живлення	3 ... 18 В
Вихідна напруга низького рівня	≤ 18 В
Вихідна напруга високого рівня	≥ 3 В
Вхідний струм низького рівня	$\leq 1,6$ мА
Вхідний струм високого рівня	$\leq 0,5$ мА
Струм витікання на виході	$\leq 0,88$ мА
Струм споживання при низькому рівні вихідної напруги	≤ 4 мА
Струм споживання при високому рівні вихідної напруги	≤ 2 мА
Час затримки поширення сигналу при вмиканні	≤ 110 нс
Час затримки поширення сигналу при вимиканні	≤ 100 нс

Мікросхема КР1554ИР22 являє собою восьмирозрядний керований по рівню регістр (на D-тригерах) з паралейним вводом-виводом даних. Корпус 2140.20-8, маса 2,6г.

Призначення виходів: 1 – дозвіл зчитування даних; 3, 4, 7, 8, 13, 14, 17, 18 – входи даних D1...D8; 2, 5, 6, 9, 12, 15, 16, 19 – виходи даних; 11 – вхід тактових імпульсів по рівню.

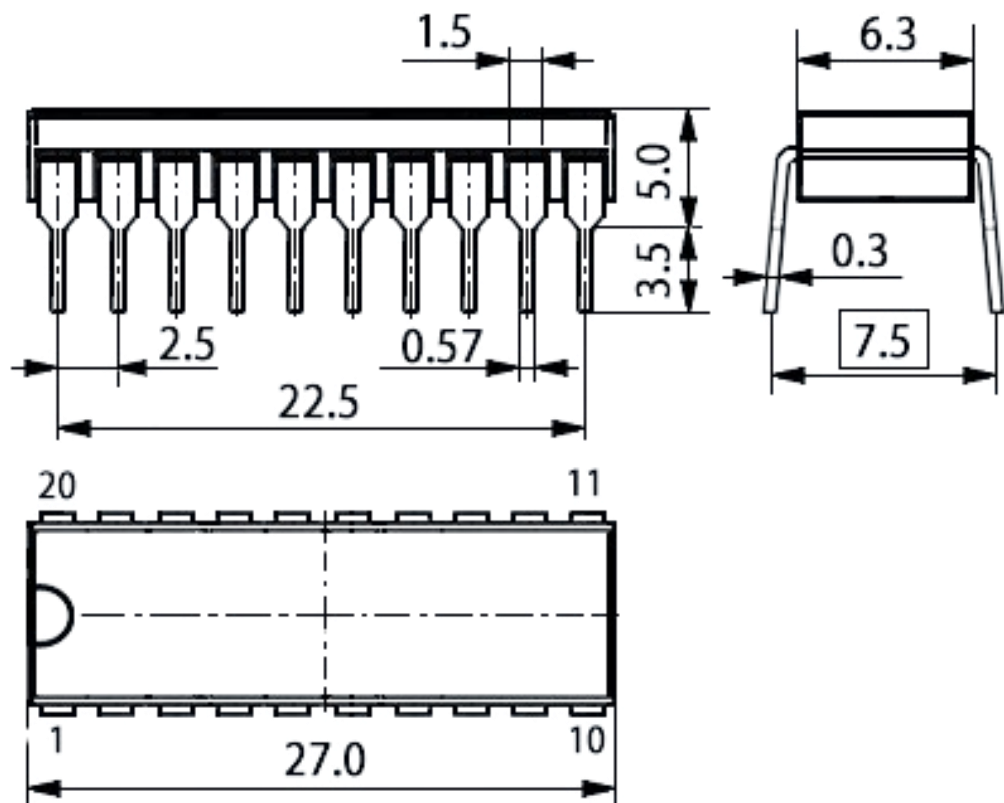


Рисунок 3.15 – Корпус мікросхеми КР1554ИР22

Зм.	Лист	№ документа	Підпис	Дата

ЕліТ 8.171.00.10.445

Лист

50

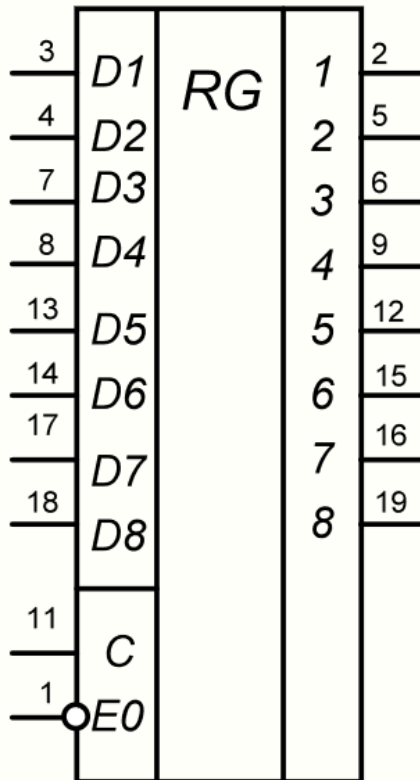


Рисунок 3.16 – Умовне графічне позначення КР1554ІР22

Мікросхема КР1533ТР2 представляє собою два RS тригера. Складається з 130 інтегральних елементів. Корпус типу 238.16-1. Маса 1,2г

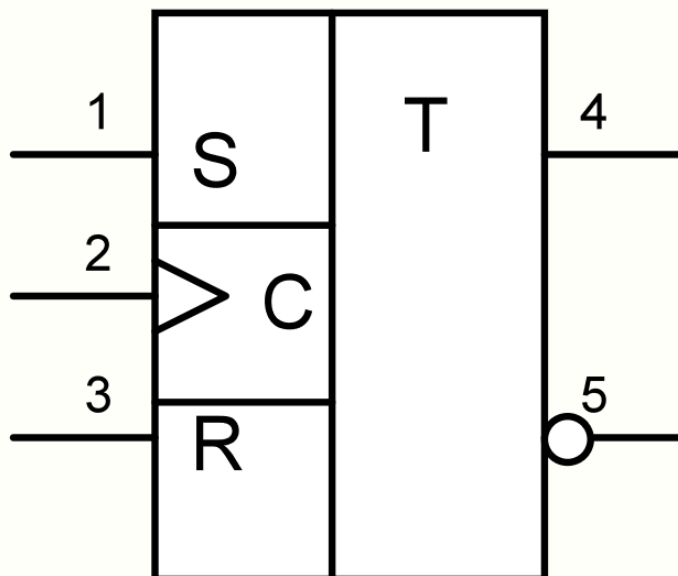


Рисунок 3.17 – Умовне графічне позначення КР1533ТР2

Зм.	Лист	№ документа	Підпис	Дата

ЕліТ 8.171.00.10.445

Лист

51

Електричні параметри мікросхеми КР1533ТР2 наведені в таблиці 3.8

Таблиця 3.8 – Електричні параметри мікросхеми КР1533ТР2

Номинальна напруга живлення	$5 \text{ В} \pm 10\%$
Вихідна напруга низького рівня	$\leq 0,4 \text{ В}$
Вихідна напруга високого рівня	$\geq 2,5 \text{ В}$
Вхідний струм низького рівня	$\leq 1,5 \text{ мА}$
Вхідний струм високого рівня	$\leq 5,5 \text{ мА}$
Струм витікання на виході	$\leq 0,2 \text{ мА}$
Струм споживання при низькому рівні вихідної напруги	$\leq 4 \text{ мА}$
Струм споживання при високому рівні вихідної напруги	$\leq 20 \text{ мкА}$
Час затримки поширення сигналу при вмиканні	$\leq 24 \text{ нс}$
Час затримки поширення сигналу при вимиканні	$\leq 22 \text{ нс}$
Ємність виходу	$\leq 5 \text{ пФ}$

3.5 розробка принципів схем блоків

При створенні принципової схеми необхідно враховувати вже розглянуте апаратне забезпечення, що було представлено у першому розділі, і обрану елементну базу. Принципові схеми виконують дві ключові функції:

Надають вказівки для відтворення схеми: Читаючи символи та слідуючи за їхніми взаємними з'єднаннями, можна відтворити всю структуру пристрою за принциповою схемою.

Предоставляють загальну інформацію про принципи функціонування та компонентну структуру схеми: Це допомагає зрозуміти принципи роботи пристрою та є корисним при вирішенні завдань ремонту або модифікації пристрою.

Такий підхід дозволяє нам не лише детально описати структуру схеми, але і надати інформацію, яка допомагає зрозуміти принципові аспекти функціонування пристрою.

Зм.	Лист	№ документа	Підпис	Дата

ЕліТ 8.171.00.10.445

Лист

52

3.5.1 Розробка блоку розподілу розрядів

Для реалізації блоку розподілу розрядів використаємо нагромаджувальні регістр зсуву які призначені для перетворення вхідного послідовного сигналу в паралейний та короточасного зберігання цифрового сигналу. Практичне застосування послідовного, паралельного зсуву регістра полягає в перетворенні даних з послідовного формату на одному дроті в паралельний формат на декількох проводах[10].

На рисунку 3.5.1 показана часова діаграма регістру зсуву двійковій інформаційної послідовності. На рисунку 3.5.1 зображена принципова схема регістру зсуву.

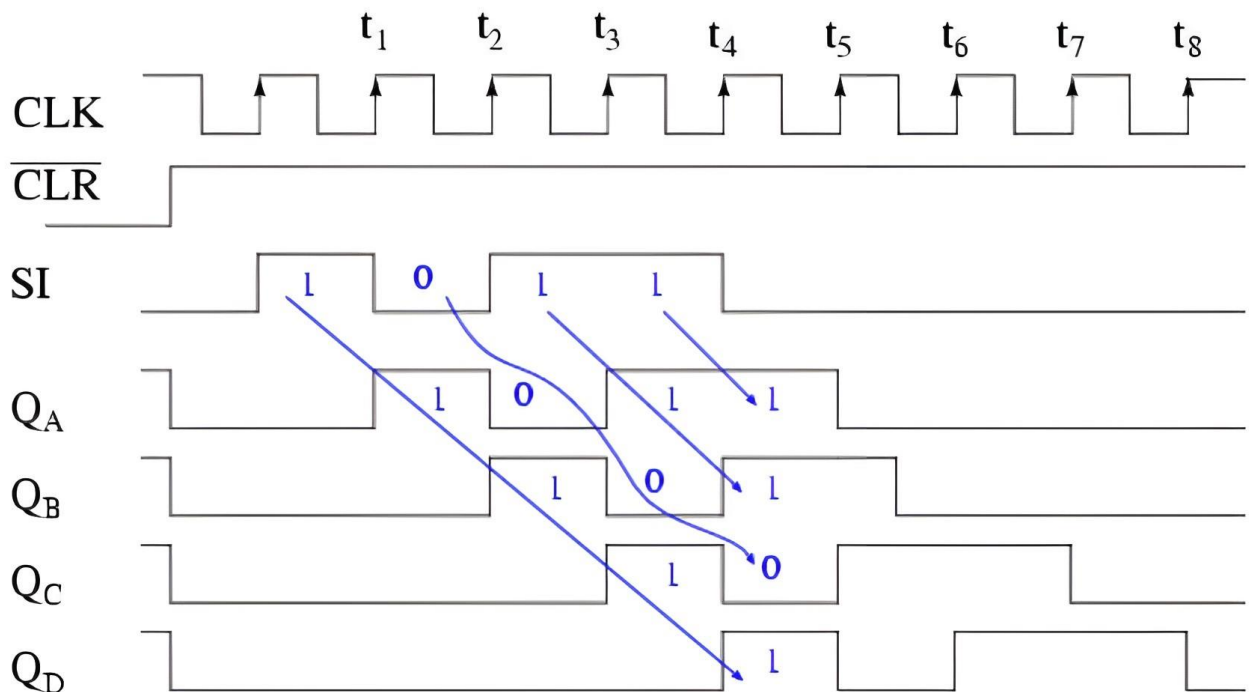


Рисунок 3.5.1 – Часова діаграма регістру зсуву

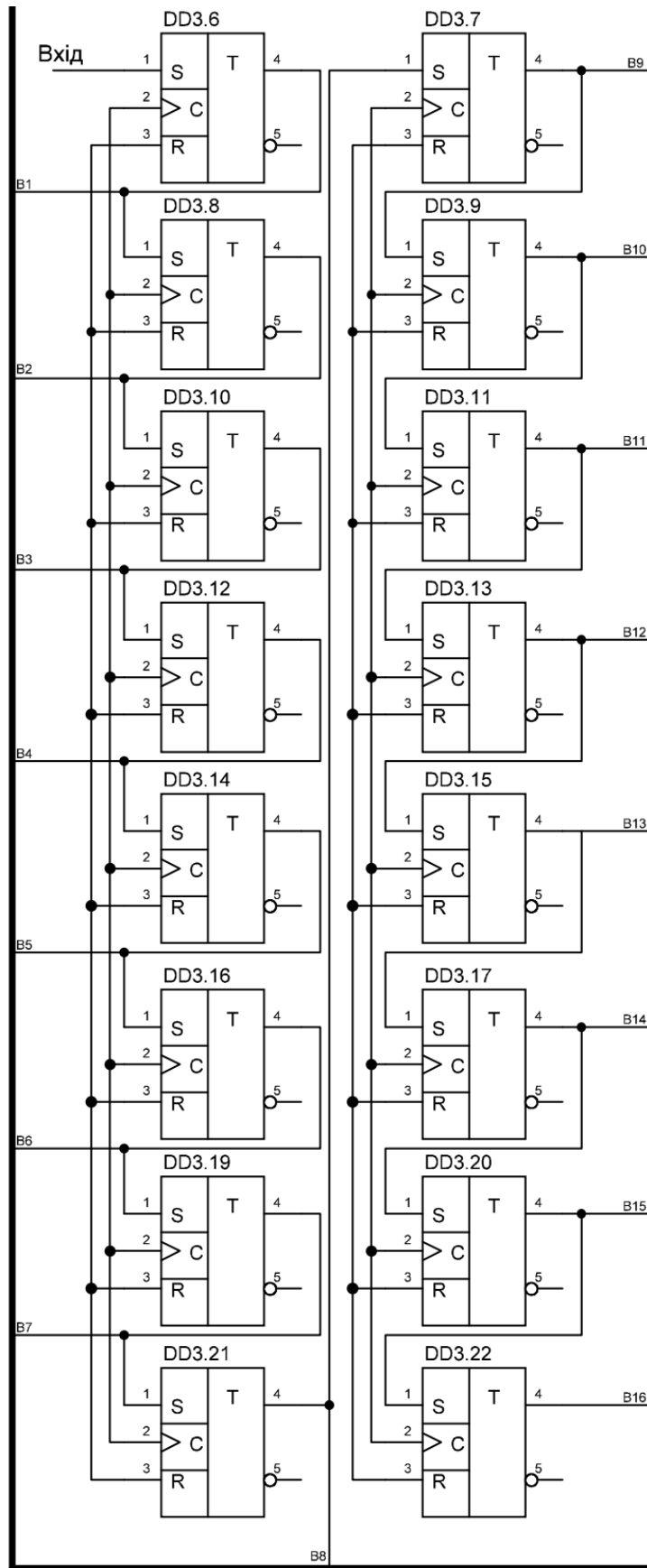


Рисунок 3.5.2 – Принципова схема регістру зсуву

Зм.	Лист	№ документа	Підпис	Дата

ЕліТ 8.171.00.10.445

Лист

54

Для дозволу зчитування даних з регістру додатково встановимо двовходові схеми І. В кожній схемі один з входів використовуватиметься для дозволу зчитування від блоку керування, а другий для сигналу відповідного розряду. На рисунку 3.5.3 зображено принципову схему дозволу зчитування даних з регістру зсуву.

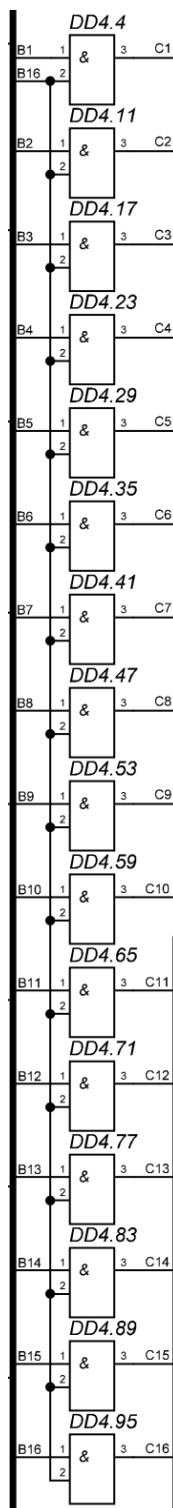


Рисунок 3.5.3 – Принципова схема дозволу зчитування даних з регістру зсуву

Зм.	Лист	№ документа	Підпис	Дата

3.5.2 Розробка блоку виявлення одиниць

При розробити блок виявлення одиниць, використаємо чотиривходові согічні елементи АБО, які будуть об'єднані за формулою логічної операції "АБО": $F = (x_0 + x_1 + x_2 + x_3) + (x_4 + x_5 + x_6 + x_7) + (x_8 + x_9 + x_{10} + x_{11}) + (x_{12} + x_{13} + x_{14} + x_{15})$. На рисунку 3.5.4 зображено принципову схему виявлення одиниць.

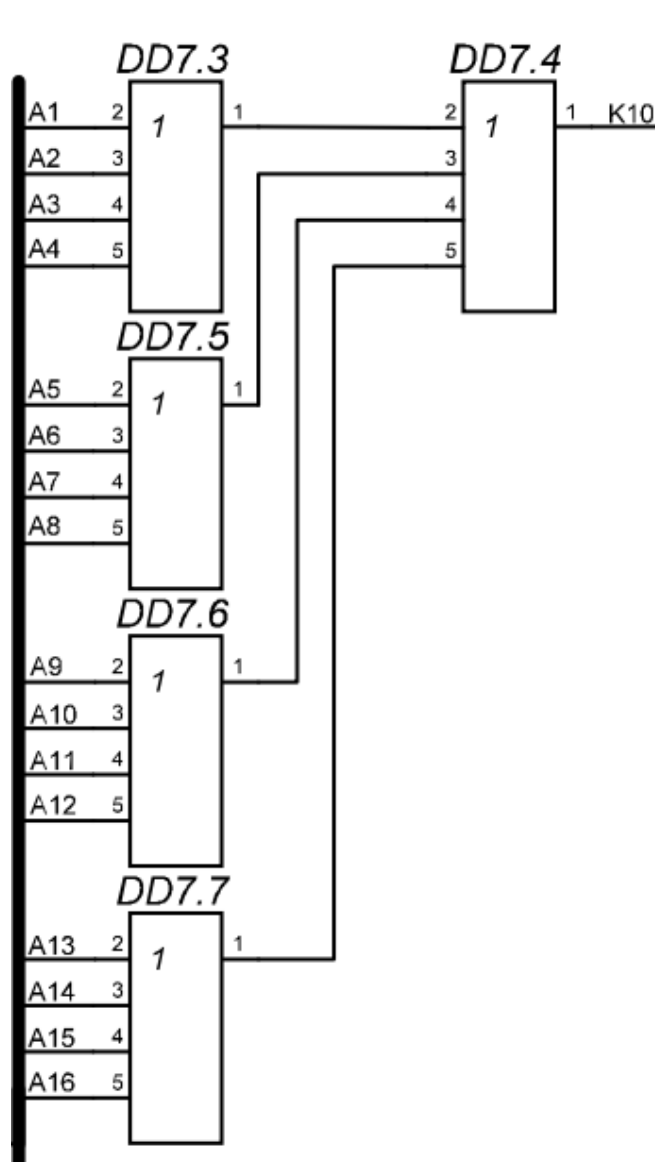


Рисунок 3.5.4 – Принципова схема виявлення одиниць

Якщо хочаб на одному з входів А з'явиться значення 1 то на виході К10 теж буде значення 1.

Зм.	Лист	№ документа	Підпис	Дата

3.5.3 Розробка блоку пам'яті

Для проведення необхідних обчислень потрібно тимчасово зберігати вхідні розряди тому на базі двох обраних ввосьмирозрядних регістрів створемо блок пам'яті. Вхідний сигнал K14 керує записом входів C1-C16, а інверсний вхід E0 керує зчитуванням даних з виходів A1-A16. На рисунку 3.5.5 зображено принципову схему блоку пам'яті.

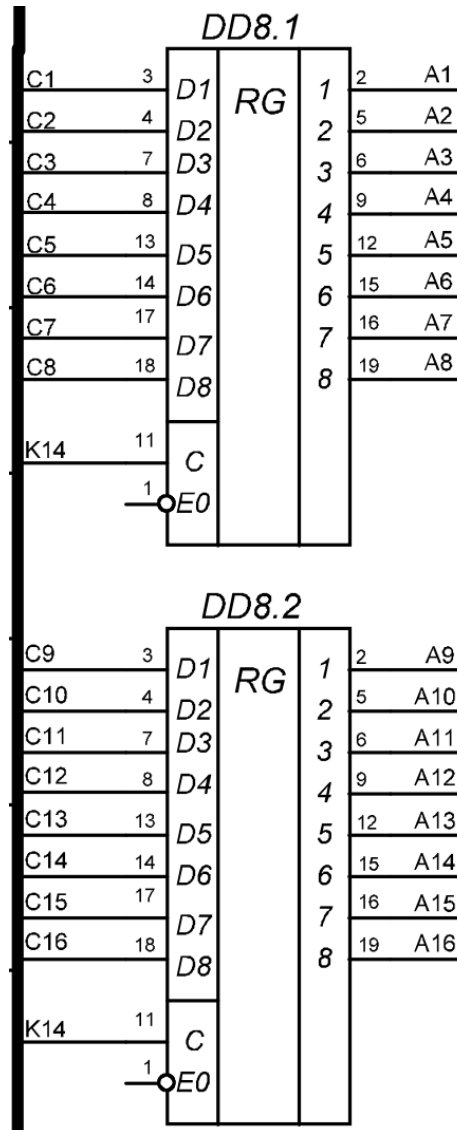


Рисунок 3.5.5 – Принципова схема блоку пам'яті

3.5.4 Розробка блоку арифметичних операцій

Для реалізації блоку арифметичних операцій двійкових чисел використаємо мікросхеми суматори розглянуті в попередньому розділі. З допомогою них зможемо виконувати необхідні арифметичні операції згідно алгоритмів роботи пристрою, а саме додавання та віднімання двійкових чисел.

Суматор працює наступним чином В віднімається з А (А і В - вхідні сигнали), результат (різниця) з'являється на виході S. Якщо більше А (як у рядку 2 таблиці), потрібно зайняти 1 в сусідньому старшому розряді. Сигнал займа вказаний у стовпці Р.

При проведенні віднімання багаторозрядних двійкових чисел необхідно враховувати перенос "одиниць" у старших розрядах. Нижче подано таблицю істинності, що містить різні можливі комбінації при відніманні двійкових чисел. Таблиця істинності станів суматора зображено в таблиці 3.5.1.

Таблиця 3.5.1 – Таблиця істинності станів суматора

Входи			Виходи	
			Різниця	Займ
А	В	С	S	Р
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Розглянемо деталізований алгоритм віднімання на конкретному прикладі, де відбувається віднімання десяткового числа 4 від десяткового числа 10 (у двійковій системі: 1010 – 0100).

Спочатку застосовується метод десяткового віднімання, а потім процес переводиться у двійкову систему. Замість віднімання числа 6 від числа 10, використовується порозрядне доповнення до 1, де всі біти інвертуються (1 стає 0, а 0 стає 1), і використовується циклічне (кругове) перенесення.

Доповненням до 1 двійкового числа 0101 є число 1010. Після цього виконується додавання із циклічним перенесенням, отримуючи проміжний результат 10010. Останній перенос циклічно зміщується вліво і додається до розряду одиниць.

Отже, шляхом використання циклічного перенесення та доповнення до 1, віднімання двійкових чисел 1010 і 0101 призводить до відповіді 0100 (число 2 в десятковому еквіваленті). Додавання чисел проходить аналогічним чином, але без інверсії другого числа. На рисунку 3.5.6 зображено принципову схему блоку арифметичних операцій.

Для побудови блоку арифметичних операцій знадобиться 4 суматори підключених послідовно один після одного для переносу числового залишку вихід Р першого суматору підключається до входу С другого суматора і т. д.

На рисунку 3.5.6 зображено принципову схему блоку арифметичних операцій.

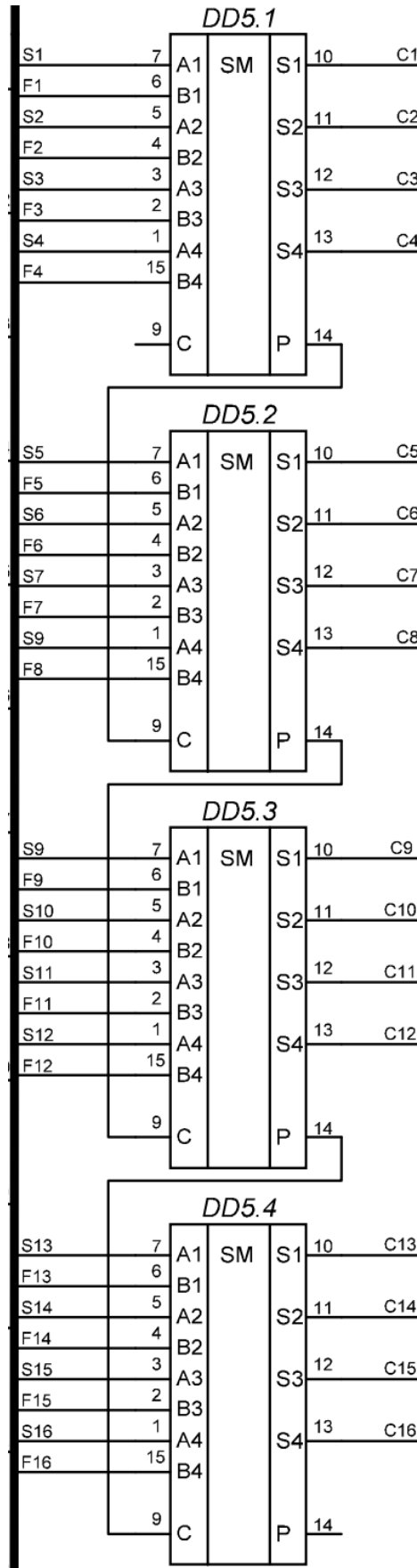


Рисунок 3.5.6 – Принципова схема блоку арифметичних операцій

3.5.5 Розробка блоку складання двійкових чисел

Для створення блоку складання двійкових чисел будемо використовувати тривходові схеми І. Перший вхід слугуватиме сигналом дозволу складання, наданого блоком керування. Другий вхід використовуватиметься для прийому основних розрядів з блоку розподілу розрядів, а третій вхід призначений для прийому розрядів з блоку віднімання двійкових чисел. При з'єднанні слід враховувати конкретні номери розрядів, які будуть складатись між собою. На рисунку 3.5.7 зображено принципову схему блоку складання двійкових чисел.

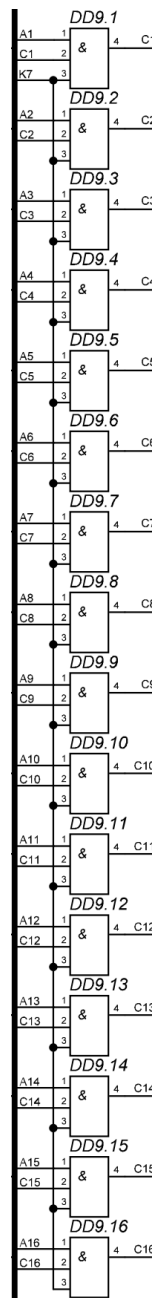


Рисунок 3.5.8 – Принципова схема блоку складання двійкових чисел

Зм.	Лист	№ документа	Підпис	Дата

3.5.6 Розробка блоку лічильника одиниць

Для створення блоку лічильника одиниць будемо використовувати тригери. Для активації режиму лічильника, потрібно подавати сигнали імпульсів з блоку генератора на входи С тригерів. Послідовно з'єднуючи інверсні виходи тригерів, ми можемо отримати вихідний сигнал для трьохрозрядної двійкової комбінації. Щоб визначити необхідну кількість тригерів, можна використати формулу $\text{Log}_2(n)$, де n - максимальна кількість підрахунку. У нашому випадку, якщо максимальна кількість розрядів для лічильника - 7, то $n = 7$, і $\text{Log}_2(7)$ дорівнює 3. Таким чином, нам потрібно використовувати три тригери для ефективного лічильника. На рисунку 3.5.8 зображено принципову схему блоку лічильника.

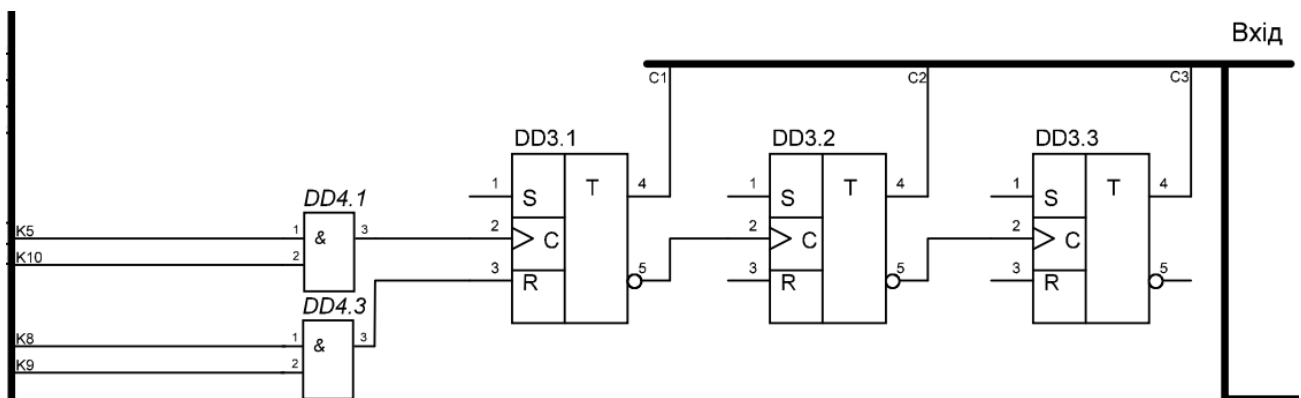


Рисунок 3.5.8 – Принципова схема лічильника

На входах першого розряду тригера додатково встановимо двовходові логічні елементи І для дозволу рахування та збросу лічильника. К5, К8, К9, К10 – керуючі сигнали дозволу які надходять з інших блоків пристрою.

3.5.7 Розробка блоку керування

Блок керування виконує функцію керування іншими блоками пристрою, забезпечуючи їх увімкнення та вимкнення у потрібний момент. Вхідний сигнал для дешифратора буде створено за допомогою лічильника, аналогічного тому, що використовується у розділі 3.5.8. Цей лічильник отримуватиме вхідний сигнал від блоку генератора.

Зм.	Лист	№ документа	Підпис	Дата

Для управління іншими блоками створимо дешифратор з двома входами та чотирма виходами. Перший вихід призначимо для входу С лічильника блоку лічильника одиниць, другий - для увімкнення блоку арифметичних операцій, третій - для дозволу складання в блоку складання двійкових чисел. Для обґрунтування роботи дешифратора складемо таблицю, яка відобразить стани виходів дешифратора в залежності від входів. Перша колонка буде позначати номер двійкової комбінації, друга - вихідну двійкову комбінацію, а третя - номер сигналу на виході. В таблиці 3.5.2 зображено таблицю істинності дешифратора

Таблиця 3.5.2 – Таблиця істинності дешифратора

	Вхідна двійкова комбінація	Номер виходу дешифратора
№	a1a2a	X
1	00	1
2	01	2
3	10	3
4	11	4

Для створення вхідного сигналу для дешифратора використаємо лічильник, згідно з формулою $\text{Log}_2(4) = 2$, визначимо необхідну кількість тригерів для охоплення всіх комбінацій. Тригери будуть працювати в режимі лічильника, отримуючи імпульси на вході С. Послідовно з'єднуючи інверсні виходи тригерів на входи С, ми отримаємо вихідний сигнал у вигляді двохрозрядної двійкової комбінації з прямих виходів тригерів. Для керування іншим алгоритмом роботи пристрою також використаємо дешифратор, виходи якого почерзі вмикатимуть блок зсуву. На рисунку 3.5.9 зображено принципову схему блоку керування.

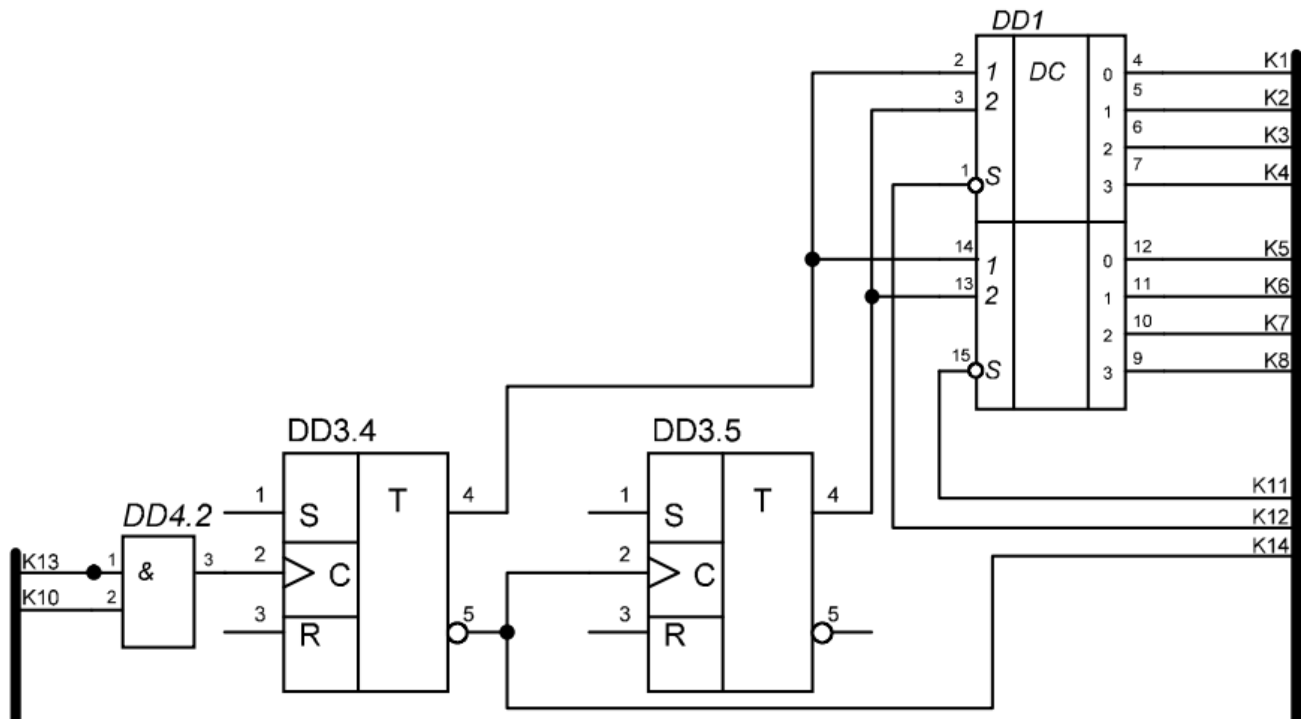


Рисунок 3.5.9 – Принципова схема блоку керування

3.5.8 Розробка блоку вибору режиму

Для розробки блоку виявлення одиниць використаємо чотиривходові схеми АБО двовходові, які будуть об'єднані за формулою: $F = (A1 + A2) + (A3 + A4) + (A5 + A6) + (A7 + A8) \vee (A9 + A10) + (A11 + A12) + (A13 + A14) + (A15 + A16)$. Як вже зазначалось блок слугіє для вибору режиму роботи пристрою в залежності від насиченості одиницями вхідної двійкової послідовності. Блок вибору режиму зображений на рисунку 3.5.10.

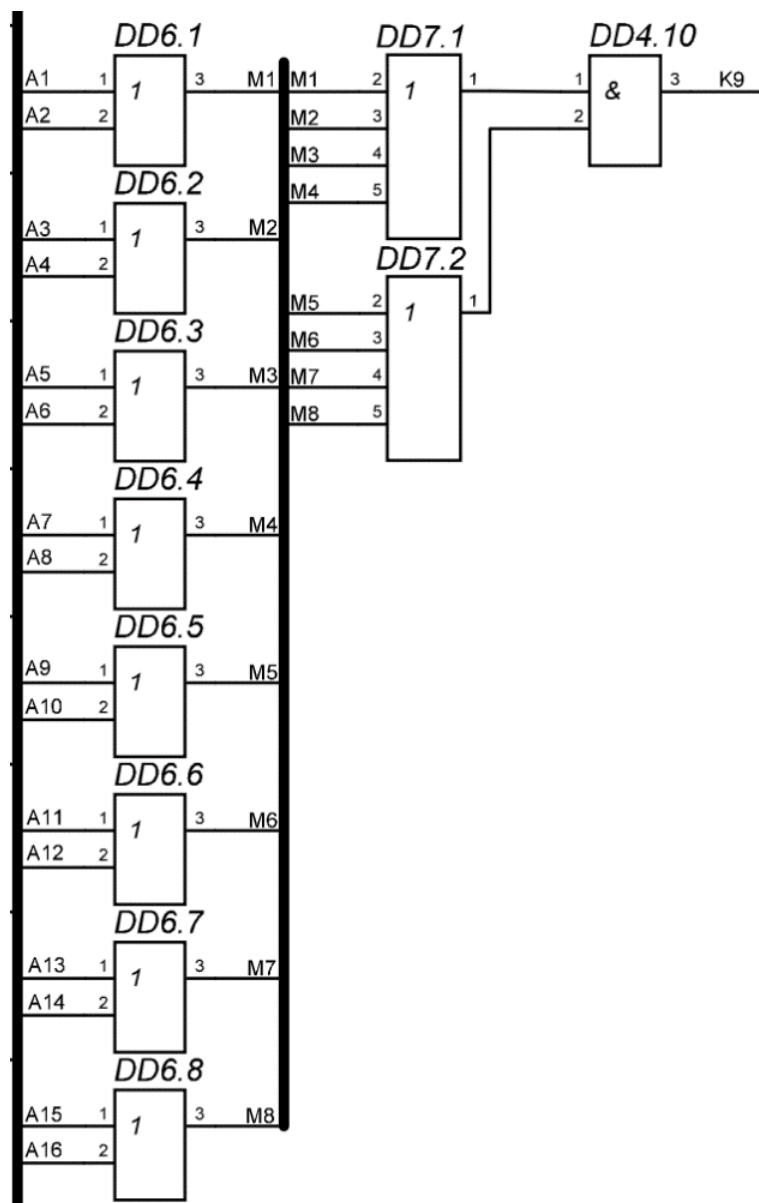


Рисунок 3.5.10 – Принципова схема блоку вибору режиму

3.5.9 Розробка блоку зсуву

Блок зсуву складається з двовходових логічних елементів І, один з входів якого призначений для керування дозволом виходу вхідної послідовності двійкових чисел. Другий вхід призначений для пропускання двійкової комбінації яка обчислюється. Виходи логічних елементів розподілені по різних місцях згідно алгоритму роботи пристрою. Принципові схема блоку зсуву зображена на рисунку 3.5.11.

Зм.	Лист	№ документа	Підпис	Дата

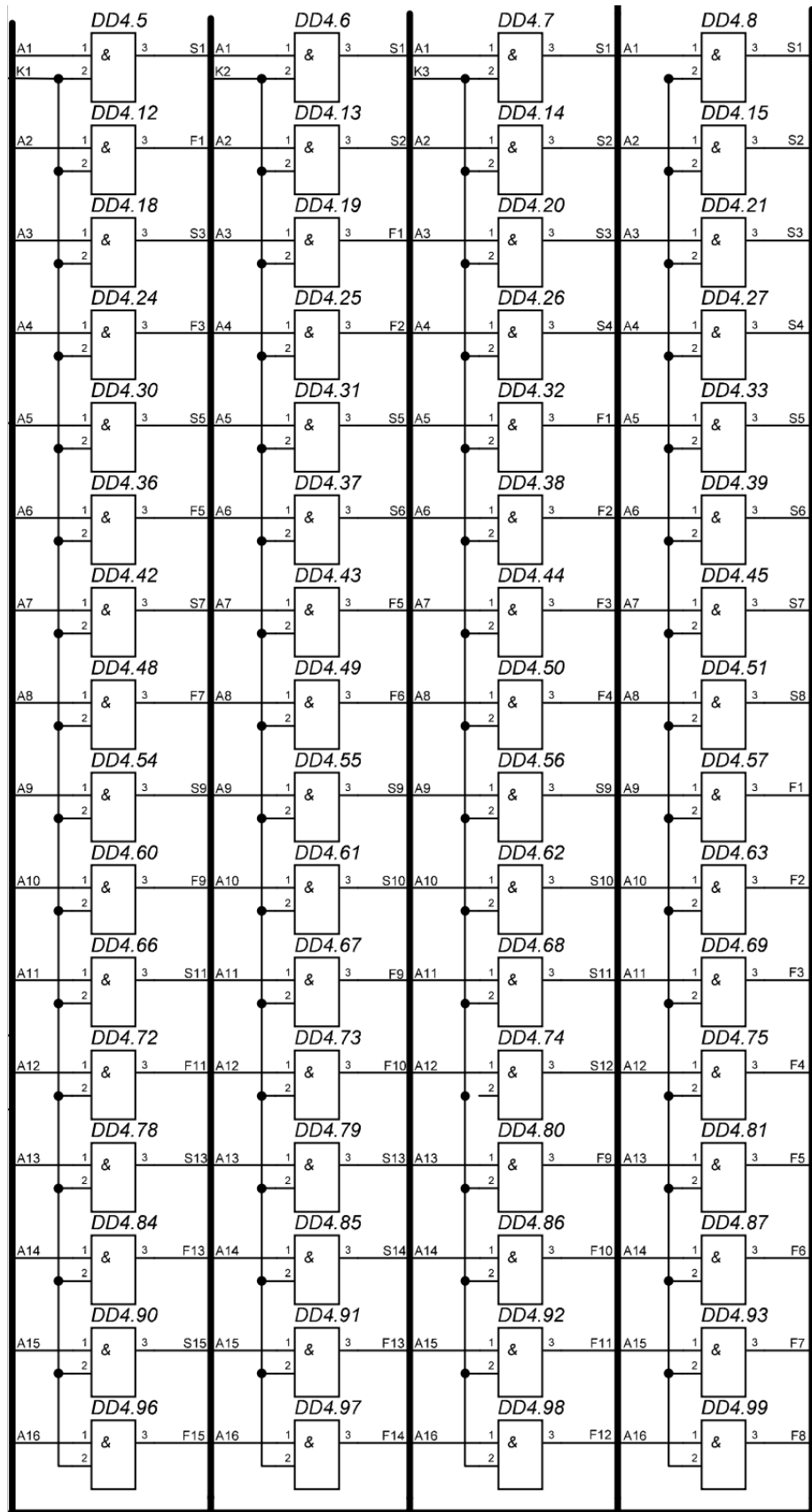


Рисунок 3.5.11 – Принципова схема блоку зсуву

Зм.	Лист	№ документа	Підпис	Дата

ЕліТ 8.171.00.10.445

Лист

66

3.5.10 Розробка блоку генератора

Зазвичай, пристрої для перетворення кодів широко використовуються в каналотворюючих апаратах, де велике значення має стабільність частот схем синхронізації. Ця стабільність полягає в здатності утримувати частоту генерації сталою навіть при змінах температури, напруги живлення та/або інших параметрів схеми. Зазвичай, автоколивальні генератори відзначаються низькою стабільністю. Щоб підвищити цю стабільність, широко використовують кварцові резонатори.

На рисунку 3.5.12 подано принципову схему генератора тактових імпульсів із кварцовим резонатором Cr1, який використовує послідовний резонанс та блок заборони. Кварцовий резонатор включений у зворотний зв'язок інверторів DD2.1 і DD2.2, забезпечуючи поворот фази на π (по ланцюгу зворотного зв'язку). У "кварцових генераторах" цифрові елементи (інвертори DD1.1 і DD1.2) працюють у лінійному режимі, що визначається резистивними зворотними зв'язками $R1 = R2 = 220$ Ом. Конденсатор $C1 = 200$ пФ виступає у ролі фазозсувного елемента. Частота тактових імпульсів, яку генерує блок генератора, залежить від параметрів резонатора та, в меншій мірі, від параметрів інших елементів схеми.

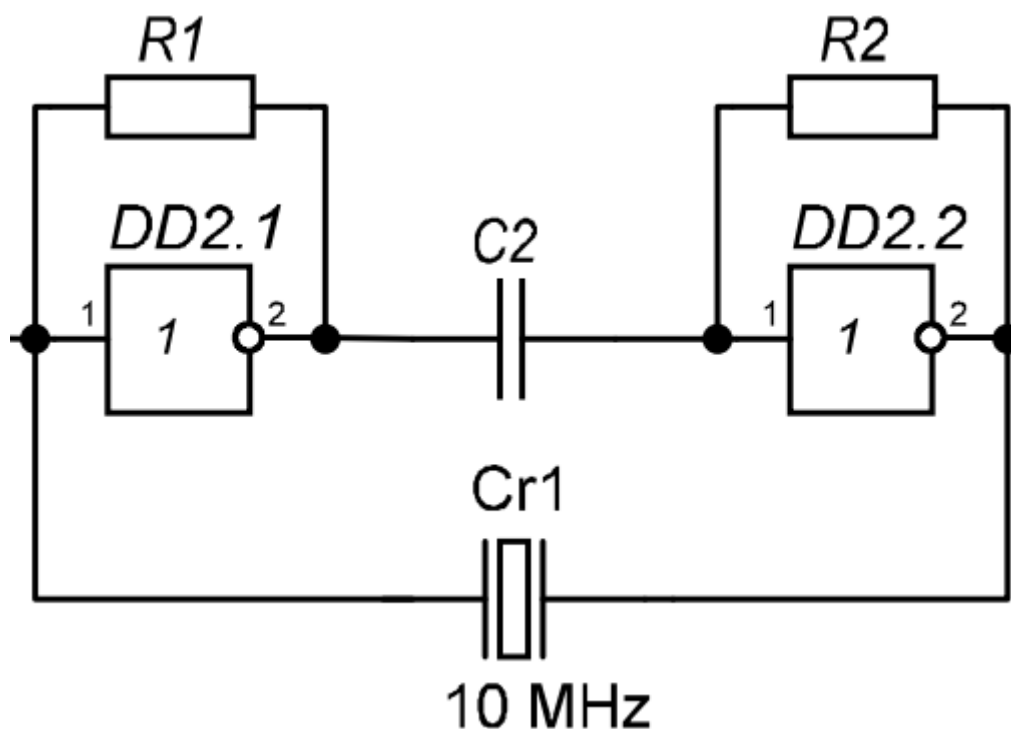


Рисунок 3.5.12 – Принципова схема блоку генератора

Зм.	Лист	№ документа	Підпис	Дата

РОЗДІЛ 4 ТЕХНІКО-ЕКОНОМІЧНА ЧАСТИНА

4.1 Розрахунок повної собівартості системи

Розрахунки витрат на матеріали та комплектуючі вироби базуються на вартості одиниці матеріалу або комплектуючого та потрібній їх кількості. Ціна за одиницю матеріалу або комплектуючого взята з роздрібних прейскурантів постачальників, які визначають вартість матеріалів, сировини, комплектуючих і послуг за одиницю випуску продукції

Результат розрахунків за цією статтею є дані, наведені в таблицях 4.1 та 4.2.

Таблиця 4.1 – Результати розрахунків витрат на комполектуючі

№ з/п	Назва комплектуючого	Кількість, шт.	Ціна за одиницю, грн	Сумарна вартість, грн
Мікросхеми				
1	SN7483	4	8,4	33,6
2	KP1533ЛЛ1	2	10	20
3	K155ЛИ5	50	8	400
4	KP1533ЛЕ1	1	9,5	9,5
5	KP1561ИД6	1	18	18
6	KP1554ИР22	2	20	40
7	KP1533ТР2	11	8,25	90,75
8	K155ЛЕ3	4	10,8	43,2
Резистори				
1	C2-23-0,125-270Ом ±5%	2	0,4	0,8
Інше				
1	HC49/S	1	5	5
2	Y5V-200nf-25v	1	4	4
Всього				664,85

Таблиця 4.2 – Результати розрахунку витрат на сировину та матеріали

Матеріал/сировина	Одиниця виміру	Норма витрат	Ціна за одиницю, грн	Сумарна вартість, грн
Провід монтажний 0,5 мм	кг	0,18	110	19,8
Склотекстоліт	м	0,07	630	44,1
Флюс	кг	0,05	160	8
Припій	кг	0,08	420	33,6
Лак	кг	0,1	55	5,5
Матеріал для корпусу	кг	0,3	315	94,5
Всього:				205,5

Враховуючи транспортно-заготівельні витрати в розмірі 5% до 15% (K_{T-3}), які будуть застосовані при проектуванні одиниці продукції, вартість комплектуючих та матеріалів збільшиться і становитиме наступну суму:

$$KM = \frac{(K+M) \cdot (100+k_{T-3})}{100} = \frac{(664,85 + 205,5) \cdot (100+10)}{100} = 957,38 \text{ (грн)}$$

Розрахунок витрат на основну заробітну плату здійснюється за наступною формулою:

$$ЗП = \sum_{i=1}^n T_{\Gamma_i} * H_{\text{ч}_i}$$

де T_{Γ_i} – це ставка оплати праці за годину для конкретного спеціаліста (наприклад, інженера електронної техніки або лаборанта), який бере участь у виробництві конкретного пристрою, системи або установки., грн/год;

$H_{\text{ч}_i}$ – період, протягом якого працівник витрачає свій час на виготовлення та налаштування конкретного пристрою. (системи/установки), грн/год;

Зм.	Лист	№ документа	Підпис	Дата
-----	------	-------------	--------	------

ЕліТ 8.171.00.10.445

Лист

69

n – це кількість осіб, які беруть участь у виготовленні та виробництві пристрою. (системи/установки).

Тарифна ставка за одну годину роботи розраховується на основі окладу спеціаліста за місяць за допомогою наступної формули:

$$T_{Г_i} = \frac{T_{M_i}}{B_{Ф_i} * 8}$$

де T_{M_i} – це оклад (ставка) спеціаліста (робітника) за місяць роботи, грн;
 $B_{Ф_i}$ – це відпрацьований фактично час за окремий розрахунковий період (місяць/квартал/рік), днів (змін).

Станом на сьогодні описані вище параметри стандартизовані та мають наступні значення:

$$T_{M_i} = 13000 \text{ грн}$$

$$B_{Ф_i} = 20 \text{ днів}$$

Враховуючи стандартизовані дані, які були наведені вище, розрахуємо тарифну ставку окремого спеціаліста за одну годину роботи:

$$T_{Г_i} = \frac{13000}{20 * 8} = 81,25 \text{ грн}$$

Використовуючи отримане значення годинної тарифної ставки, розрахуємо витрати на основну заробітну плату за наступною формулою, враховуючи, що $N_{ч_i}$ (нормована тривалість робочого дня) дорівнює 3 годинам, а n (кількість робочих днів) дорівнює 1:

$$ЗП = 3 * 81,25 = 243,75 \text{ грн}$$

За наступною формулою розраховується значення додаткової заробітної плати:

$$З_{д} = ЗП * \frac{K_{д}}{100} = 243,75$$

					<i>ЕліТ 8.171.00.10.445</i>	Лист
						70
Зм.	Лист	№ документа	Підпис	Дата		

де K_d – Це відсоток, який регулює значення додаткової заробітної плати в межах від 10% до 30% від загальної заробітної плати (ЗП). Припустимо, що значення K_d (коефіцієнт додаткової заробітної плати) дорівнює 10%, і використовуючи наведену вище формулу, розрахуємо значення додаткової заробітної плати.

$$Z_d = 243,75 * \frac{10}{100} = 24,37 \text{ грн}$$

До відрахувань на соціальні заходи відносяться знімання визначених відсотків від розміру основної та додаткової заробітної плати відповідно до встановлених стандартів. Ці відрахування включають:

- внески на загальнообов'язкове державне пенсійне страхування;
- внески на державне страхування від нещасних випадків;
- внески на загальнообов'язкове державне соціальне страхування на випадок безробіття;
- відрахування у зв'язку з тимчасовою втратою працездатності;
- витрати, пов'язані з народженням дитини та похованням.

З урахуванням всіх наведених ставок, можна визначити коефіцієнт для розрахунку відрахувань на соціальні заходи:

$$V_{CЗ} = (Z_d + ЗП) * \frac{36,3}{100} = (24,37 + 243,75) * \frac{36,3}{100} = 97,32 \text{ грн}$$

Витрати на експлуатацію та обслуговування обладнання визначаються як вартість утримання та експлуатації обладнання (ВУЕУ), де ВУЕУ розраховується як добуток базової заробітної плати на обслуговуючих працівників та відсотка відносної вартості обладнання (прийmemo %ВУЕУ рівним 135%).

$$ВУЕУ = ЗП * 1,35 = 243,75 * 1,35 = 329,1$$

Наступною статтею витрат є загальновиробничі витрати, які охоплюють витрати, пов'язані з керуванням підрозділом, відрядженням працівників цеху, амортизаційними відрахуваннями від вартості основних засобів загальноцехового

										Лист
										71
Зм.	Лист	№ документа	Підпис	Дата	<i>ЕліТ 8.171.00.10.445</i>					

призначення і т.д. Ці витрати визначаються у розмірі від 130% до 250% посадового окладу. Для прикладу, припустимо, що загальновиробничі витрати становлять 130% від основної заробітної плати.

$$V_{ЗВ} = ЗП * 1,5 = 243,75 * 1,3 = 316,87 \text{ грн}$$

Перелік статей:

1. Матеріали та комплектуючі.
2. Основна заробітна плата.
3. Додаткова заробітна плата.
4. Відрахування на соціальні заходи.
5. Витрати на утримання та експлуатацію обладнання.
6. Загальновиробничі витрати

Враховуючи всі зазначені вище види витрат, вартість виробництва системи розраховується за такою формулою:

$$C_B = КМ + ЗП + ВУЕУ + V_{ЗВ} + V_{СЗ} + З_д = 957,38 + 243,75 + 329,1 + 316,87 + 97,32 + 24,37 = 1968,8 \text{ грн}$$

Витрати на адміністрування визначаються на основі аналізу повної вартості виробу і, в середньому, можуть становити від 140% до 200% від основної заробітної плати.

$$V_A = ЗП * 1,4 = 243,75 * 1,4 = 341,25 \text{ грн}$$

Слідуючою категорією витрат є витрати на збут, які включають в себе витрати на рекламу та підготовку системи перед її впровадженням. Приблизний розмір цих витрат визначається у діапазоні від 5% до 10% від виробничої собівартості такої системи.

$$V_3 = C_B * 0,05 = 1968,8 * 0,05 = 98,4 \text{ грн}$$

$$ПС = C_B + V_3 + V_A = 1968,8 + 98,4 + 341,25 = 2408,45 \text{ грн}$$

					<i>ЕліТ 8.171.00.10.445</i>	Лист
						72
Зм.	Лист	№ документа	Підпис	Дата		

На підставі отриманих результатів створимо таблицю калькуляції собівартості проекрованої системи:

Таблиця 4.3 – Калькуляція собівартості проекрованої системи

№	Найменування статей калькуляції	Проектний варіант, грн
1	Матеріали та комплектуючі	957,38
2	Основна заробітна плата	243,75
3	Відрахування на соціальні заходи	97,32
4	Додаткова заробітна плата	24,37
5	ВУЕУ	329,1
6	Загальновиробничі витрати	316,87
Виробнича собівартість		1968,8
7	Витрати на збут	98,4
8	Адміністративні витрати	341,25
Повна собівартість		2408,45

4.2 Розрахунок ціни системи

У ринковій економіці існують різні методи ціноутворення: прибуток плюс прибуток, надання фіксованої суми прибутку в залежності від рівня попиту і т.д. Оптову ціну системи що проектується розрахуємо за принципом «прибуток плюс собівартість».

$$C_{\text{опт}} = \text{ПС} + \frac{\text{ПС} \cdot R}{100} = 2408,45 + \frac{2408,45 \cdot 0,35}{100} = 2416,88 \text{ грн}$$

де R – рентабельність проекрованої системи, що становить 35% від її собівартості, враховуватиметься при розрахунку відпускної ціни. Додамо до цієї суми податок на додану вартість (ПДВ), що становить 20%.

$$C_{\text{відп}} = C_{\text{опт}} \cdot 1,2 = 2416,88 \cdot 1,2 = 2900,25 \text{ грн}$$

Позитивні аспекти даної стратегії включають його простоту та легку зрозумілість функції ціноутворення, зокрема відшкодування витрат на виробництво та забезпечення рентабельності системи. Недолік полягає у тому, що цей підхід не враховує ринкові фактори, зокрема попит. Проте в умовах ринкової економіки існують сценарії, коли підприємства можуть виявити його доцільним: у відсутності конкуренції (монополія), обмеженої державою рентабельності, при виконанні одноразових замовлень, або при виробництві унікальної продукції. Важливо відзначити, що для встановлення реальної ціни, яка відповідає умовам ринку, необхідно провести відповідні маркетингові дослідження.

4.3 Визначення річних витрат споживача

Витрати експлуатації споживача за рік ($V_{срв}$) охоплюють різні складові, такі як витрати на електроенергію (живлення системи), амортизаційні відрахування, заробітну плату персоналу, який обслуговує систему, витрати на технічне обслуговування, поточний ремонт та інші витрати.

Визначення витрат на електроенергію здійснюється за допомогою наступної формули:

$$V_e = W * C_e * t,$$

де W – потужність проектованої системи, кВт;
 C_e – вартість одного кіловата електроенергії за годину роботи (кВт*год), грн/кВт*год (за тарифом для промислових користувачів);
 t – кількість годин роботи системи що розробляється за рік, год. Для вищенаведених параметрів приймемо наступні значення:

$$W = 2,4 * 10^{-3} \text{ кВт}, C_e = 2,64 \text{ грн/кВт}, t = 1000 \text{ год.}$$

$$V_e = 2,4 * 10^{-3} * 2,64 * 1000 = 6,33 \text{ грн}$$

Сума річних амортизаційних витрат визначається відповідно до норм амортизації, враховуючи вартість системи. Вартість системи включає ціну системи, а також витрати на транспортування та монтаж, що становлять від 5% до 15% від

загальної ціни. Розрахунок амортизаційних витрат для електронних пристроїв (систем) здійснюється за допомогою прямолінійного методу.

$$A = F * a$$

F – первісна ціна системи; a – норма амортизаційних витрат за рік.

Розрахуємо первісну ціну системи:

$$F = C_{\text{відп}} + B_{\text{тм}} = 2900,25 + 290,02 = 3190,27 \text{ грн}$$

де $C_{\text{відп}}$ – це відпускна (роздрібна) ціна системи; $B_{\text{тм}}$ – це сума витрат на монтаж та транспортування планованої системи, яка встановлюється як 10% від вартості системи. Норма витрат на амортизацію розраховується методом прямолінійного методу, відповідно до терміну корисного використання системи користувачем, за наступною формулою:

$$a = \frac{1}{T_{\text{вик}}} = 0,2$$

де $T_{\text{вик}}$ – термін корисного використання споживачем системи визначається відповідно до Податкового кодексу України і повинен бути не менше 2 років ($T_{\text{вик}} \geq 2$ років). Проте, для розробленої системи термін корисного використання споживачем становить $T_{\text{вик}} = 5$ років. Тому, величина амортизації розраховується за прямолінійним методом і дорівнює:

$$A = 3190,27 * 0,2 = 638,05 \text{ грн}$$

Розходи на поточний ремонт та інші витрати охоплюють вартість заміни несправних електронних компонентів, що виникли впродовж експлуатаційного терміну, і витрати на проведення демонтажно-монтажних робіт. Розмір витрат на ремонт визначається як 5% від капітальних витрат, з урахуванням прогнозу ймовірності виникнення несправності.

$$B_{\text{пр}} = F * k_{\text{пр}} = 3190,27 * 0,05 = 159,5$$

					<i>ЕліТ 8.171.00.10.445</i>	Лист
						75
Зм.	Лист	№ документа	Підпис	Дата		

У рамках цього розділу наведено результати розрахунків за різними статтями витрат, такими як вартість електроенергії, амортизаційні відрахування та витрати на поточний ремонт. Підсумкові показники представлені у вигляді Таблиці 4.4.

Таблиця 4.4 – Експлуатаційні витрати споживача за рік

№	Статті аксплуатаційних витрат	Сума, грн
1	Витрати на електроенергію	6,33
2	Витрати на амортизацію	638,05
3	Витрати на ремонт	159,5
Всього(В _{СРВ})		803,88

4.4 Розрахунок повної ціни користувача

Ціна користувача $C_{кор}$ включає в себе капітальні вкладення F та загальні витрати на експлуатацію пристроїв за рік (В_{СРВ}) протягом терміну служби пристрою.:

$$C_{кор} = F + V_{срв} * T_{вик} = 3190,27 + 803,88 * 5 = 7209,67 \text{ грн}$$

4.5 Висновки до техніко-економічної частини

Під час аналізу техніко-економічного аспекту дослідження розглянуті різні стратегії формування цін та методи розрахунку витрат на створення імпульсної системи зі збільшеною стійкістю до перешкод на основі біноміальної системи числення. Основні фактори, що впливають на остаточну вартість системи, проаналізовані та розраховані, зокрема:

- потужність системи;
- середня кількість годин наробітку на відмову;
- ймовірність безвідмовної роботи системи;
- маса пристрою;

- виробнича собівартість;
- відпускна ціна;
- експлуатаційні витрати користувача щорічно;
- повна ціна користувача.

Порівняльний аналіз техніко-економічних показників розробленого пристрою рентоспроможність на ринку. Також зазначено можливість зниження загальної вартості пристрою шляхом автоматизації виробничого процесу.

Таблиця 4.5 – Порівняння техніко-економічних характеристик системи-аналога та проектованої системи

№	Показник	Значення	Одиниці вимірювання
1	Потужність системи	$2,4 \cdot 10^{-3}$	кВт
2	Середня кількість годин наробітку на відмову	6500	год
3	Ймовірність безвідмовної роботи системи	99	%
4	Маса пристрою	0,5	кг
5	Виробнича собівартість	1968,8	грн
6	Повна собівартість	2408,45	грн
7	Відпускна ціна	2925,25	грн
8	Експлуатаційні витрати користувача за рік	803,88	грн
9	Повна ціна користувача	7209,67	грн

ВИСНОВОК

У кваліфікаційній роботі магістра було проведено проектування адаптивної електронної системи обчислення одиниць в інформаційних послідовностях пристрою для розрахунку кількості одиниць у двійкових послідовностях, використовуючи компоненти жорсткої логіки, такі як комбінаційні логічні схеми, тригери, регістри та дешифратори.

У першому розділі було проведено огляд основних методів розрахунку кількості одиниць у двійкових послідовностях та розглянуті області їх використання в електронних системах. Другий розділ присвячено розгляду обраного методу розрахунку, а також розробці структурної схеми. У третьому розділі представлені принципові схеми блоків розробленого пристрою.

Під час виконання даної роботи були закріплені знання в галузі електронних та цифрових систем, набуті нові навички в проектуванні цифрових пристроїв та розробці структурних та принципових схем. Ці навички є важливим активом для подальшої роботи та вивчення.

					<i>ЕліТ 8.171.00.10.445</i>	Лист
						78
<i>Зм.</i>	<i>Лист</i>	<i>№ документа</i>	<i>Підпис</i>	<i>Дата</i>		

СПИСОК ЛІТЕРАТУРИ

1. Цифрова схемотехніка : підручник / О. А. Борисенко. – Суми : Сумський державний університет, 2016. – 200 с
2. Рейнгольд Э., Нивергельт Ю., Део Н. Комбінаторні алгоритмі. Теорія та практика. 1980. 480 с.
3. Оцінка традиційних способів підрахунку одиниць та електронних пристроїв на їх основі /Кулик І.А., Греков Т.В., Манько Д.С., // Фізика, електроніка, електротехніка (ФЕЕ-2022). Матеріали та програма науково-технічної конференції. – Суми: СумДУ, 2022. – С.70
4. Синтез швидкодіючих алгоритмів стиску на основі адресно-векторного кодування для інформаційних задач АСУ. [Текст] : дис... канд. техн. наук: 05.13.06 / Кулик Игор Анатолійович; СумДУ. - Сумы, 1998. - 140 л. - л. 136-140
5. <https://studref.com/671305/prochie/summatory> (оновлена інформація 2022 р.)
6. <http://electricalschool.info/electronica/1918-logicheskie-jelementy-i-ili-ne-i-ne-ili.html>(оновлена інформація 2022 р.)
7. https://etk.lntu.edu.ua/pluginfile.php/7361/mod_resource/content/2/%D1%80%D0%B5%D0%B3%D1%96%D1%81%D1%82%D1%80%D0%B8.pdf (оновлена інформація 2021 р.)
8. https://etk.lntu.edu.ua/pluginfile.php/7361/mod_resource/content/2/%D1%80%D0%B5%D0%B3%D1%96%D1%81%D1%82%D1%80%D0%B8.pdf (оновлена інформація 2021 р.)
9. https://stud.com.ua/28285/tovaroznavstvo/deshifratori_shifratori (оновлена інформація 2020 р.)
10. <http://surl.li/ohsjv> (оновлена інформація 2023 р.)
11. <http://vozom.ho.ua/R4/page72.html>(оновлена інформація 2022 р.)
12. https://imrad.com.ua/userdata/modules/productFiles/yTqo7OFg_1533%D0%A2%D0%A02.pdf(оновлена інформація 2023 р.)
13. <https://www.microshemca.ru/4555/>(оновлена інформація 2022 р.)
14. <https://car-led.org/ua/p1618369925-kr1533tr2-sn74als279n-dip16.html>(оновлена інформація 2023 р.)

<i>Позначення</i>	<i>Назва</i>	<i>Кількість</i>	<i>Примітка</i>
	<i>Мікросхеми</i>		
<i>DD1</i>	<i>KP1561ИД6</i>	<i>1</i>	
<i>DD2.1- DD2.2</i>	<i>KP1533ЛЕ1</i>	<i>1</i>	
<i>DD3.1- DD3.22</i>	<i>KP1533ТР2</i>	<i>11</i>	
<i>DD4.1- DD4.84</i>	<i>K155ЛИ5</i>	<i>42</i>	
<i>DD5.1- DD5.4</i>	<i>K155ИМ3</i>	<i>4</i>	
<i>DD6.1- DD6.8</i>	<i>KP1533ЛЛ1</i>	<i>2</i>	
<i>DD7.1- DD7.7</i>	<i>K155ЛЕ3</i>	<i>4</i>	
<i>DD8.1- DD8.2</i>	<i>KP1554ИР22</i>	<i>2</i>	
<i>DD9.1- DD9.16</i>	<i>K511ЛА2</i>	<i>8</i>	
	<i>Резистори</i>		
<i>R1-R2</i>	<i>C2-23-0,125-270Ом ±5%</i>	<i>2</i>	
	<i>Конденсатори</i>		
<i>C1</i>	<i>Y5V-200nf-25v</i>	<i>1</i>	
	<i>Інше</i>		
<i>Cr1</i>	<i>HC49/S</i>	<i>1</i>	

					ЕліТ 8.171.00.10.445 ПЕЗ		
<i>Зм.</i>	<i>Лист</i>	<i>№ документа</i>	<i>Підпис</i>	<i>Дата</i>			
<i>Розробник</i>	<i>Греков Т.В.</i>				<i>Лист</i>	<i>Лист</i>	<i>Листів</i>
<i>Перевірив</i>	<i>Кулик І.А.</i>					80	80
<i>Реценз.</i>					СумДУ, гр. ЕС.м-21		
<i>Н. Контр</i>	<i>Гапич В.М.</i>						
<i>Затверд.</i>	<i>Опанасюк А.С.</i>						
					Адаптивна електронна система обчислення двійкових одиниць в інформаційних послідовностях Перелік елементів		

