

681.32(07)

M54

МЕТОДИЧНІ ВКАЗІВКИ
до виконання курсової роботи
з дисципліни «Комп'ютерна логіка»
для студентів напряму підготовки
6.050102 «Комп'ютерна інженерія»

Міністерство освіти і науки України
Вінницький національний технічний університет

МЕТОДИЧНІ ВКАЗІВКИ
до виконання курсової роботи
з дисципліни «Комп'ютерна логіка»
для студентів напрямку підготовки
6.050102 «Комп'ютерна інженерія»

Вінниця
ВНТУ
2013

Рекомендовано до друку Методичною радою Вінницького національного технічного університету Міністерства освіти і науки, молоді та спорту України (протокол №10 від 14.10.2012 р.).

Рецензенти:

Т. О. Савчук, кандидат технічних наук, професор

А. В. Дудатьєв, кандидат технічних наук, доцент

Методичні вказівки до виконання курсової роботи з дисципліни «Комп'ютерна логіка» для студентів напряму підготовки 6.050102 «Комп'ютерна інженерія» / Уклад. Н. О. Біліченко, Д. В. Кисюк - Вінниця : ВНТУ, 2013. – 52 с.

Дані методичні вказівки покликані допомогти студентам у написанні курсової роботи з дисципліни «Комп'ютерна логіка», а також у закріпленні теоретичного матеріалу з даної дисципліни та набутті практичних навичок синтезу операційних та керуючих автоматів, зокрема у розробці словесних алгоритмів та граф-схем виконання арифметичних операцій на суматорах заданого типу для чисел заданої розрядності та форми подання, а також у описах методики контролю заданої операції.

ЗМІСТ

ВСТУП.....	4
1 ОСНОВНІ ВИМОГИ ДО ЗМІСТУ РОЗДІЛІВ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ.....	5
1.1 Особливості виконання основних арифметичних операцій в ЕОМ.....	5
1.1.1 Операція алгебраїчного додавання.....	5
1.1.2 Операція множення.....	6
1.1.3 Операція ділення.....	8
1.2 Поняття граф-схеми алгоритму та правила її складання.....	10
1.3 Основні поняття теорії цифрових автоматів.....	11
1.4 Синтез керуючого автомата.....	13
1.5 Контроль виконання арифметичних операцій.....	15
1.6 Завдання на курсову роботу.....	17
1.7 Приклад виконання завдання.....	22
2 ОСНОВНІ ВИМОГИ ДО ВМІСТУ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ.....	34
2.1 Вміст пояснювальної записки.....	34
2.2 Титульний лист.....	34
2.3 Індивідуальне завдання.....	35
2.4 Анотація.....	35
2.5 Зміст.....	35
2.6 Вступ.....	36
2.7 Висновки.....	36
2.8 Перелік посилань.....	37
2.9 Додатки.....	38
3 ОСНОВНІ ВИМОГИ ДО ОФОРМЛЕННЯ ТЕКСТУ ТА РИСУНКІВ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ.....	39
3.1 Загальні правила.....	39
3.2 Вимоги до оформлення розділів та підрозділів.....	39
3.3 Правила написання тексту.....	40
3.4 Оформлення формул.....	41
3.5 Оформлення ілюстрацій.....	43
3.6 Оформлення таблиць.....	44
ДОДАТКИ.....	47
Додаток А Зразок титульного аркуша курсової роботи.....	48
Додаток Б Зразок індивідуального завдання.....	49
Додаток В Зразок штампа для схеми керуючого автомата.....	50
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	51

ВСТУП

Мета курсової роботи – закріплення теоретичного матеріалу з дисципліни та набуття практичних навичок синтезу операційних та керуючих автоматів.

Для досягнення цієї мети студент повинен згідно з отриманим варіантом самостійно розробити словесний алгоритм та граф-схему виконання операції множення на суматорі заданого типу для чисел заданої розрядності та форми подання; синтезувати операційний та керуючий автомат для виконання заданої операції; описати методику контролю заданої операції та навести приклад на контрольних числах.

Виконання курсової роботи базується на знанні студентом таких дисциплін як «Вступ до фаху», «Організація функціонування ЕОМ» та «Комп'ютерна логіка».

Тема курсової роботи – синтез операційного та керуючого автоматів.

Варіанти курсової роботи задаються викладачем згідно з номером групи і номером студента в списку групи.

Пояснювальна записка має приблизний обсяг 25÷30 сторінок, синтезована схема оформлюється в додатку на аркуші формату А4 (А3). При оформленні пояснювальної записки необхідно дотримуватися вимог ДСТУ 3008-95.

1 ОСНОВНІ ВИМОГИ ДО ЗМІСТУ РОЗДІЛІВ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ

Індивідуальне завдання на розробку студент отримує у вигляді технічного завдання (додаток Б), яке містить в собі необхідні вимоги до оформлення курсової роботи і структури пояснювальної записки. Варіант завдання на виконання курсової роботи студент вибирає з таблиць (відповідно до номера групи і номера студента в списку групи або за методикою, запропонованою викладачем).

У **вступі** повинно бути відображено загальний рівень розвитку галузі, а також стислий опис методу виконання заданої арифметичної операції та реалізації технічного завдання.

В **розділі 1** повинен бути розроблений машинний алгоритм виконання арифметичної операції: складено словесний алгоритм та граф-схему її виконання, а також побудовано операційний автомат та наведено приклад виконання отриманого алгоритму для контрольних чисел.

В **розділі 2** необхідно детально описати процес синтезу керуючого автомата: закодувати граф-схему алгоритму, побудувати таблицю переходів автомата, отримати функції збудження тригерів, мінімізувати їх, після чого побудувати відповідну їм комбінаційну схему.

В **розділі 3** необхідно провести контроль арифметичної операції за допомогою заданого методу та продемонструвати його роботу на контрольних числах.

1.1 Особливості виконання основних арифметичних операцій в електронних обчислювальних машинах (ЕОМ)

1.1.1 Операція алгебраїчного додавання

Додавання чисел з фіксованою комою у цифрових обчислювальних машинах може виконуватися в одному з машинних кодів: прямому, оберненому або доповняльному. Суму також отримаємо в одному з цих кодів. При реалізації операції додавання знаковий розряд й інформаційна частина числа розглядаються як єдине ціле, в результаті чого з від'ємними числами машина оперує як і з додатними. Головна перевага такого методу полягає в тому, що правильний знак суми отримується автоматично в процесі додавання знакових цифр операндів і цифри переносу з сусіднього молодшого розряду. У випадку виникнення **одиниці переносу** зі знакового розряду суми її потрібно відкинути при додаванні в доповняльному коді і додати до молодшого інформаційного розряду суми при додаванні в оберненому коді (тобто, виконати циклічний перенос одиниці переповнення).

Для виявлення **переповнення розрядної сітки** при додаванні вводиться допоміжний розряд у знакову частину зображення числа, що

називають розрядом переповнення. Таке подання числа називається **модифікованим кодом**.

Знакова частина додатного числа містить цифри 00, а від'ємного 11. Ознакою переповнення розрядної сітки є наявність у знаковій частині цифр 01 або 10.

Додавання у прямому коді виконується тільки над числами одного знака. Числа з різними знаками підсумовують в оберненому або доповняльному коді.

Машинна операція додавання чисел з плаваючою комою здійснюється в шість етапів:

- вирівнюються порядки доданків: менший порядок збільшується до більшого, а мантиса числа зсувається вправо на відповідну кількість розрядів;

- виконується перетворення мантис доданків в один з модифікованих кодів (обернений чи доповняльний);

- додаються мантиси доданків;

- у разі необхідності мантиса суми переводиться в прямий код;

- виконується нормалізація суми;

- виконується коригування порядку результату.

Число вважається нормалізованим, якщо старший інформаційний розряд дорівнює одиниці. Також слід зауважити, що додавання мантис здійснюється в оберненому або доповняльному кодах. Якщо мантиса результату додавання нормалізована, то до цього результату приписуємо порядок будь-якого з операндів. В протилежному випадку відбувається **нормалізація числа**. В процесі виконання операції додавання можливе порушення нормалізації справа та зліва. Ознакою порушення нормалізації числа справа є наявність різних цифр у знакових розрядах. У цьому випадку необхідно зсунути число вправо на один розряд. А ознака порушення нормалізації числа зліва – це присутність однакових цифр в розряді переповнення і в старшому розряді цифрової частини. Він показує на необхідність зсуву числа вліво на один розряд.

Кожний зсув мантиси вліво при нормалізації веде до зменшення порядку результату додавання на одиницю. А кожний зсув мантиси вправо – до збільшення. Таким чином відбувається коригування порядку.

1.1.2 Операція множення

Найпростіше множення виконується у прямому коді. У разі подання чисел з фіксованою комою воно реалізується у два етапи. На першому етапі визначається знак добутку шляхом додавання за модулем два цифр знакових розрядів співмножників (див. табл. 1.1). На другому етапі здійснюється множення модулів співмножників, потім, у разі потреби,

округлення модуля добутку, після чого до модуля результату дописується його знак, який визначений на першому етапі.

Таблиця 1.1 - Правила визначення знака добутку

Обчислення вручну	Обчислення в машині
$(+) \cdot (+) = (+)$	$0 \oplus 0 = 0$
$(+) \cdot (-) = (-)$	$0 \oplus 1 = 1$
$(-) \cdot (+) = (-)$	$1 \oplus 0 = 1$
$(-) \cdot (-) = (+)$	$1 \oplus 1 = 0$

\oplus - оператор додавання за модулем два

В залежності від способу формування суми часткових добутків розрізняють чотири основних методи виконання множення та відповідно чотири структури арифметично-логічного пристрою (АЛП) для цієї операції.

Метод 1. Множення, починаючи з молодших розрядів множника, з зсувом часткових добутків вправо при нерухомому множеному. Послідовність дій в кожному циклі виконання множення визначається молодшим розрядом регістра множника, куди послідовно одна за одною надходять цифри множника. Оскільки по мірі зсуву множника вправо старші розряди регістра множника звільняються, він може бути використаний для збереження молодших розрядів добутку, що надходять з молодшого розряду суматора часткових добутків по мірі виконання множення.

Метод 2. Множення, починаючи з молодших розрядів множника, при зсуві множеного вліво та нерухомій сумі часткових добутків. Послідовність дій визначається, як і в першому варіанті, молодшим розрядом регістра множника. При цьому методи регістр множеного та суматор часткових добутків повинні мати подвійну довжину.

Метод 3. Множення, починаючи зі старших розрядів множника, при зсуві суми часткових добутків вліво та нерухомому множеному. Послідовність дій виконання множення визначається старшим розрядом регістра множника. Даний метод потребує додаткового в порівнянні з першим методом обладнання. Множення починається з молодших розрядів і множене зсувається вліво на один розряд в кожному такті.

Метод 4. Множення, починаючи зі старших розрядів множника, при зсуві вправо множеного та нерухомій сумі часткових добутків. Послідовність дій на кожному кроці множення визначається старшим розрядом регістра множника.

Разом з тим, застосування обернених та доповняльних кодів дозволяє

істотно спростити операцію алгебричного додавання. Проте використання такої форми подання чисел має певні особливості:

1. Дії, що виконуються під час множення обернених (доповняльних) кодів, залежать від знака множника.

2. Добуток обернених (доповняльних) кодів співмножників дорівнює оберненому (доповняльному) коду результату тільки у випадку додатного множника.

3. Якщо множник є від'ємним числом, то обернений (доповняльний) код добутку одержується додаванням поправок $[A]_{об} \cdot 2^{-n}$ і $[[A]_{об}]_{об}$ ($[[A]_{топ}]_{топ}$) до добутку обернених (доповняльних) кодів співмножників.

4. Оскільки поправки мають різну вагу, то послідовність їх додавання залежить від того, з яких розрядів множника починається множення.

Для чисел $A = m_A 2^{P_A}$ і $B = m_B 2^{P_B}$, що подані в формі з плаваючою комою, добуток обчислюється за формулою 1:

$$AB = m_A 2^{P_A} m_B 2^{P_B} = m_A m_B 2^{P_A + P_B} = m_C 2^{P_C} \quad (1)$$

Відповідно процес множення складається з чотирьох етапів:

- множення мантис;
- додавання порядків;
- нормалізація й округлення мантиси добутку;
- коригування порядку добутку.

Підвищення швидкості виконання множення за чотирма основними методами досягається застосуванням **методів прискорення операцій множення**. За способом реалізації вони поділяються на апаратні та логічні.

Апаратні методи прискорення множення потребують для свого здійснення введення додаткової апаратури в основні арифметичні кола пристрою для множення.

Логічними методами прискорення множення називають такі методи, реалізація яких не потребує змін основної структури арифметичних кіл пристрою для множення, а прискорення досягається тільки за рахунок ускладнення схеми керування цим пристроєм. До логічних методів прискорення операції множення відносять метод *множення з пропуском тактів додавань* у тих випадках, коли чергова цифра множника є нулем; метод множення з перетворенням цифр множника шляхом *групування розрядів* та метод множення з *послідовним перетворенням цифр множника*.

1.1.3 Операція ділення

Ділення чисел у двійковій системі числення класифікується таким чином:

- за формою подання чисел:

- з фіксованою комою;
- з плаваючою комою.
- за механізмом виконання операції:
 - з відновленням остачі;
 - без відновлення остачі;
- за швидкодією:
 - просте;
 - прискорене;
- за точністю результату:
 - з округленням результату;
 - без округлення результату.

Для того, щоб поділити двійкові числа з відновленням остачі, необхідно виконати такі операції.

1. Подвоїти модуль діленого $|A|$.
 2. Відняти від подвоєного модуля діленого модуль дільника. Одержана різниця $R = 2|A| - |B|$ є першою остачею.
 3. Проаналізувати знак остачі R . Якщо $R \geq 0$, то черговому розряду частки присвоїти «1» і перейти до п. 5; якщо ж $R < 0$, то черговому розряду частки присвоїти «0».
 4. Відновити остачу, додавши модуль дільника $|B|$.
 5. Подвоїти остачу.
 6. Визначити чергову остачу, віднявши від попередньої остачі модуль дільника. Перейти до п. 3.
- Вищевказані дії слід виконувати до одержання всіх необхідних цифр частки.

Алгоритм ділення модулів чисел без відновлення остачі зводиться до виконання таких дій.

1. Подвоїти модуль діленого $|A|$.
2. Відняти від подвоєного модуля діленого модуль дільника. Одержана різниця $R = 2|A| - |B|$ є першою остачею.
3. Проаналізувати знак остачі R . Якщо $R \geq 0$, то черговому розряду частки присвоїти 1; якщо ж $R < 0$, то черговому розряду частки присвоїти «0».
4. Подвоїти остачу.
5. Визначити чергову остачу, віднявши від попередньої остачі модуль дільника якщо $R \geq 0$ і додавши до попередньої остачі модуль дільника якщо $R < 0$. Перейти до п. 3.

При прискореному діленні на алгоритм з відновленням чи без відновлення остачі накладається алгоритм виконання операції прискорення.

Для прискорення операції ділення використовують аналіз старших

інформаційних розрядів. Якщо два старші інформаційні розряди дорівнюють одиниці, то у наступний розряд частки записуємо «1», якщо «0», то записуємо «0» і проводимо зсув суматора і регістра результату на два розряди вліво.

Операція ділення належить до розряду неточних операцій, оскільки результат, як правило, отримують з деякою похибкою. Тому ознакою закінчення операції ділення може бути або досягнення заданої точності (кількість розрядів у частці), або отримання чергової остачі, яка рівна нулю.

Після закінчення операції ділення двійкових чисел за вибраним алгоритмом (з відновленням чи без відновлення остачі, з прискоренням чи без прискорення) для забезпечення округлення результату операцію ділення за вибраним алгоритмом продовжують для визначення ще одного розряду результату. Потім аналізують молодший інформаційний розряд результату. Якщо у цьому розряді записана «1», то її додають до попереднього розряду результату, якщо «0», то останній інформаційний розряд результату просто ігнорують. При використанні методу ділення двійкових чисел без округлення результату описані дії не проводять, обмежуючись виконанням алгоритму ділення.

При діленні двійкових чисел з плаваючою комою спочатку визначають знак результату за правилом алгебри логіки “сума за модулем два”, потім проводять коригування форми запису чисел ($|mA| < |mB|$) та визначають порядок результату за формулою 2:

$$P_C = P_A - P_B, \quad (2)$$

де P_C - порядок результату;

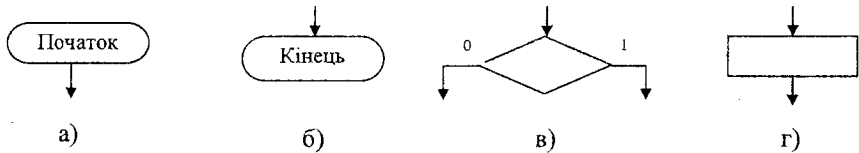
P_A - порядок числа А;

P_B - порядок числа В.

Далі виконують операцію ділення мантиси числа А на мантису числа В за правилами ділення двійкових чисел з фіксованою комою за одним з вибраних алгоритмів (з відновленням чи без відновлення остачі; з округленням чи без округлення; просте чи прискорене). Отриманий результат нормалізують.

1.2 Поняття граф-схеми алгоритму та правила її складання

Граф-схема алгоритму (ГСА) є найбільш наочною формою подання роботи автомата. ГСА – це орієнтований зв’язний граф, що містить вершини чотирьох типів (рис. 1.1). Кожен з існуючих входів і виходів вершин може розгалужуватись потрібну кількість разів.



а) початкова вершина; б) кінцева вершина;
в) умовна вершина; г) операційна вершина.

Рисунок 1.1 – Типи вершин ГСА

Вершина «Початок» входів не має. Вершина «Початок» (рис. 1.1, а) і будь-яка операторна (рис. 1.1, г) вершина мають по одному виходу. Вершина «Кінець» (рис. 1.1, б) виходів не має. Будь-яка умовна вершина (рис. 1.1, в) має два виходи, які позначаються символами «Так» і «Ні». Замість цих символів можуть бути використані цифри «1» і «0», відповідно.

ГСА повинна задовольняти такі вимоги:

- 1) містити скінченне число вершин;
- 2) мати лише одну початкову та одну кінцеву вершини;
- 3) входи і виходи кожної з вершин повинні з'єднуватися дугами, спрямованими від виходу попередньої до входу наступної вершини;
- 4) кожний вихід повинен з'єднуватися тільки з одним входом;
- 5) будь-який вхід повинен з'єднуватися принаймні з одним виходом;
- 6) для будь-якої вершини графа існує хоча б один шлях до кінцевої вершини;
- 7) у кожній умовній вершині записується тільки один з елементів множини логічних умов;
- 8) у кожній операторній вершині записується один або декілька операторів, які можуть одночасно виконуватись, причому допускається, що операторна вершина буде пустою.

1.3 Основні поняття теорії цифрових автоматів

Необхідність формального опису роботи комп'ютера та його окремих частин в процесі проєктування потребує використання спеціального математичного апарату, який необхідний при будь-яких розробках різних методів обробки інформації, при синтезі і аналізі інформаційних процесів, які відбуваються при роботі пристрою. Для цього вводять поняття абстрактного цифрового автомата.

Цифровим автоматом (ЦА) називають пристрій, призначений для обробки та перетворення цифрової інформації. Найбільш розповсюдженим типом цифрових автоматів є комп'ютери.

Цифровим автоматом вважаються пристрої, які характеризуються набором деяких внутрішніх станів $A = \{a_1(t), a_2(t), \dots, a_n(t)\}$, в які потрапляє автомат під впливом вхідних сигналів і відповідних команд розв'язання задачі (рис. 1.2).

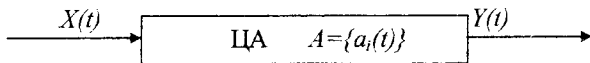


Рисунок 1.2 – Цифровий автомат

Відповідно до рис. 1.2, математичною моделлю ЦА є деякий абстрактний автомат, який задається таким чином, в початковий момент часу $t = t_0$, внутрішній стан автомата $a(t_0) = a_1$ і зберігається таким до моменту часу $t = t_1$, коли змінюється на стан a_2 , ця зміна відбувається під впливом вхідного сигналу $X(t_1)$. При цьому формується вихідний сигнал $Y(t_1) = Y_1$, який визначається як функція від внутрішнього стану a_1 і вхідного сигналу x_1 : $Y = \lambda(a_1, x_1)$. В загальному випадку вважається, що при поданні довільного сигналу x_i автомат переходить від стану $a(t)$ в стан $a(t+1)$, який, в свою чергу, є функцією від попереднього стану і вхідного сигналу. В результаті цього переходу виробляється відповідний сигнал Y .

Абстрактний ЦА описується шістьма основними параметрами:

- a_1 – початковий стан автомата;
- $A = \{a_1, a_2, \dots, a_m\}$ - множина (алфавіт) внутрішніх станів;
- $X = \{x_1, x_2, \dots, x_k\}$ - алфавіт вхідних сигналів;
- $Y = \{y_1, y_2, \dots, y_n\}$ – алфавіт вихідних сигналів;
- $\delta = \{\delta_1, \delta_2, \dots, \delta_p\}$ – сукупність функцій переходу автомата з одного стану в інший;

- $\lambda = \{\lambda_1, \lambda_2, \dots, \lambda_f\}$ – сукупність функцій виходу автомата.

Сукупність правил переходу автомата з одного стану в інший залежно від вхідної інформації і внутрішніх станів називається **алгоритмом перетворення інформації** в цифровому автоматі.

На відміну від абстрактного автомата, реально використовуються скінченні автомати, які мають скінченні множини вхідних сигналів, вихідних сигналів та внутрішніх станів. Усі скінченні автомати поділяються на **повністю визначені**, у яких область визначення D функцій δ та λ збігається з множиною перетину алфавітів вхідного та станів, яка є в свою чергу множиною пар $\{a_n, x_i\}$; та **неповністю визначені** часткові скінченні автомати, для яких функції внутрішніх станів і вихідних сигналів δ та λ визначаються не для всіх пар $\{a_n, x_i\}$, Крім того скінченні автомати підрозділяють за виглядом функцій виходів $y_j = \lambda(a_n, x_i)$ та переходів $a_s = \delta(a_n, x_i)$. За цією ознакою автомати поділяються на автомати Мілі та

Мура. Будь-який автомат можна описати функцією стану і вихідною функцією:

$$\begin{cases} a(t+1) = \delta(a(t), x(t)), \\ y(t) = \lambda(a(t), x(t)). \end{cases} \quad (3)$$

Цьому виразу відповідає автомат, який називається **автоматом Мілі**. На відміну від нього, для **автомата Мура** функція стану не змінюється, а вихідний сигнал залежить тільки від внутрішнього стану автоматів:

$$\begin{cases} a(t+1) = \delta(a(t), x(t)), \\ y(t) = \lambda(a(t)). \end{cases} \quad (4)$$

1.4 Синтез керуючого автомата

Процес синтезу керуючого автомата включає такі етапи.

1. Кодоване подання графа мікропрограми або отримання граф-схеми алгоритму роботи керуючого автомата.

2. Розмічування граф-схеми алгоритму для визначення станів керуючого автомата, який функціонує відповідно до вибраної моделі (Мілі або Мура).

3. Складання структурної таблиці переходів та виходів автомата.

4. Отримання функцій збудження елементів пам'яті (тригерів).

5. Побудова комбінаційної частини автомата.

Кодоване подання граф-схеми алгоритму здійснюється шляхом заміни мікрокоманд, записаних в операторних вершинах, відповідними їм керуючими сигналами y_j , а умов, які перевіряються в умовних вершинах, відповідними їм сигналами X_i .

Для **автомата Мура** вихідний сигнал залежить лише від внутрішнього стану, тобто $y = \lambda(a)$. Тому кожна операторна вершина повинна бути відмічена символом вихідного стану автомата a_i .

Для побудови **автомата Мілі** слід пам'ятати, що вихідний сигнал залежить як від внутрішнього стану, так і від вхідного сигналу (тобто умови X_i). Кодування граф-схеми автомата Мілі відбувається не так, як для автомата Мура. Символом a_0 кодується вхід першої вершини графа, що йде за початковою, і вхід кінцевої вершини. Виходи інших операторних вершин відмічаються символами a_i , причому виходи різних вершин мають різні номери станів.

За кодовою граф-схемою роботи автомата складається таблиця переходів і виходів. Для цього спочатку здійснюють кодування станів автомата двійковими кодами, визначають тип та кількість тригерів. Потім за таблицею переходів визначають значення сигналів на входах тригерів, при яких здійснюються переходи; визначають функції збудження тригерів

та виконують їх мінімізацію. За знайденими виразами будується комбінаційна схема керуючого автомата на вибраних елементах.

Таблиця переходів і виходів має однаковий вигляд для автоматів Мілі та Мура і будується в такій послідовності. В полі a_i графа t таблиці 1.2 записуємо поточний стан автомата, в полі a_i графа $t+1$ – наступний його стан, у полі X_i – умову переходу зі стану $a_i(t)$ в стан $a_i(t+1)$ згідно із граф-схемою алгоритму. В полі Y_i записуються мікрооперації, які виконуються при переході автомата в наступний стан. У полі «Тригери» вказано сигнали, які необхідно подати на входи відповідних запам'ятовувальних елементів. Таким чином описуються всі можливі переходи автомата (табл. 1.2).

Таблиця 1.2– Таблиця переходів і виходів автомата

t		$t+1$	Тригери				
a_i	X_i	a_i	Y_i	T_1	T_2	...	T_n

Кількість тригерів, які необхідні для організації пам'яті керуючого автомата, визначається як найближче більше ціле від двійкового логарифму кількості станів за формулою (5):

$$R = \lceil \log_2 M \rceil, \quad (5)$$

де M – кількість станів автомата,
 R – шукана кількість тригерів.

За отриманою таким чином таблицею записуються та зводяться до мінімальної форми функції збудження тригерів та функції виходів цифрового автомата. Слід пам'ятати, що функції виходів цифрового автомата Мура залежать лише від внутрішніх станів (графа a_i , поля $t+1$) і не залежать від умов переходу X_i . Далі вибирається система елементів, з яких будується схема автомата. У більшості схем як елементи пам'яті використовуються елементарні автомати (тригери), що мають такі особливості:

- 1) вони є автоматами Мура і мають два стійких стани;
- 2) станам елементарного автомата відповідають два різних вихідних сигнали: одиничний (коли на прямому виході тригера одиниця, а на інверсному – нуль) та нульовий;
- 3) у загальному випадку елементарний автомат може мати декілька фізичний входів.

Крім того, схема будь-якого керуючого автомата повинна містити певну кількість логічних елементів, що утворюють функціонально повну систему для синтезу необхідної комбінаційної схеми.

1.5 Контроль виконання арифметичних операцій

Арифметичні операції виконуються на суматорах прямого, оберненого і доповняльного коду. Припустимо, що зображення чисел зберігаються в машині в деякому коді, тобто операція перетворення в заданий код або обернений проводиться на виході чи вході машини. Методика реалізації операцій контролю подається таким чином.

По-перше, розглянемо зображення числа у відповідному коді як єдину кодову комбінацію.

Розглянемо послідовність дій на прикладі суматора прямого коду: якщо додаються тільки цифрові частини зображення чисел, а знак зберігається, то контроль можна здійснити двома способами:

1) розділений контроль знакової і цифрової частин зображень результату;

2) загальний контроль всього зображення .

При розділеному способі для контролю знакових розрядів можна використовувати засіб для визначення переповнення, оскільки у випадку модифікованого коду поява помилок в знакових розрядах приведе до розбіжності інформації в них. При перевірці правильності обробки цифрових частин зображень також не виникає особливих труднощів .

При загальному способі контролю необхідне коригування контрольного коду результату через те, що знак результату при додаванні повторює знак доданків. Загальний спосіб контролю може бути використаний і для суматорів оберненого і доповняльного кодів.

При додаванні чисел на суматорі доповняльного коду можливе коригування контрольного коду в випадку, якщо знакові розряди зображень містять одиницю тому, що при цьому виникає одиниця переносу із знакового розряду. Очевидно, що контрольний код суми буде рівним $r_{(A+B)} = r_{A_d} + r_{B_d} - \alpha$, де α – кореляція ($\alpha=1$, якщо виник перенос із знакового розряду, і $\alpha=0$ – якщо переносу немає).

Контроль за модулем. Різноманітні задачі можна розв'язувати за допомогою методу контролю, оснований на властивостях порівняння. Розвинені на цій основі методи контролю арифметичних і логічних операцій називають контролем за модулем.

Існують два методи отримання контрольного коду: числовий і цифровий.

Числовий метод контролю. При числовому методі контролю код заданого числа знаходиться як найменший додатний залишок від ділення числа на вибраний модуль p :

$$r_A = A - \left\{ \frac{A}{p} \right\} \cdot p, \quad (6)$$

де p – модуль числа,

$\{A/p\}$ – ціла частина від ділення числа,

A – контрольне число.

Величина модуля p значно впливає на якість контролю; якщо $p = q$ (де q – основа системи числення, в якій виражене число) і має місце числовий контроль, то контролюється тільки молодший розряд числа і контроль не має сенсу; для $p = q^m$ справедливі аналогічні роздуми, тому, що якщо $m < n$ (де n розрядність числа), знову не всі розряди числа беруть участь в контролі і помилки в розрядах, які старші m , взагалі не сприймаються.

При числовому методі контролю за модулем p для знаходження залишку використовують операцію ділення, яка потребує значних витрат машинного часу.

Цифровий метод контролю. При цифровому методі контролю контрольний код числа утворюється діленням суми цифр числа на вибраний модуль:

$$r'_A = \sum_i a_i - \left\{ \frac{\sum_i a_i}{p} \right\} p \quad \text{або} \quad r'_A \equiv \sum_i a_i \pmod{p}, \quad (7)$$

де p – модуль числа,

a_i – i -та цифра числа.

Можливі два шляхи отримання контрольного коду:

- 1) безпосереднє ділення суми цифр на модуль p ;
- 2) підсумовування цифр за модулем p .

Другий шлях простіше реалізується, оскільки якщо $a_i < p$, то контрольний код отримується лише операцією додавання.

Проте при цифровому методі властивості порівнянь не завжди справедливі. Це відбувається через наявність переносів (запозичень) при наявності арифметичних дій над числами. Тому знаходження контрольного коду результату операції відбувається обов'язково з коригуванням.

Арифметичний контроль (AN-контроль) потребує утворення контрольного коду у вигляді AN , де A – контрольоване число, N – модуль. Вагою арифметичного коду прийнято вважати кількість ненульових символів у кодовій комбінації, а відстань, що визначається як вага різниці кодових комбінацій, називається арифметичною відстанню.

Якщо відстань між двома числами A_1 та A_2 рівна d , то це означає, що перехід від одного числа до другого досягається додаванням величини d . У цьому випадку всі комбінації чисел, що знаходяться між A_1 та A_2 , є забороненими. Відповідно для виявлення d -кратної помилки необхідно

мати відстань не меншу за $d+1$. Якщо $d=1$, то такий код не зможе виявляти помилки. Величина відстані для кодів AN -вигляду залежить від величин A та N .

Для будь-якого числа A в системі основи $q=2$ існує єдине подання вигляду $A_q = a_n \cdot 2^n + a_{n-1} \cdot 2^{n-1} + \dots + a_0$, де $a_i = \pm 1$ або 0 , в якому немає двох сусідніх коефіцієнтів, відмінних від нуля.

Таке подання містить мінімальне число ненульових коефіцієнтів і називається канонічним. У канонічному поданні вага будь-якого числа, починаючи з 2^{i-1} й до числа $2^i + 2^{i-1}$, на одиницю більша за вагу чисел від 1 до 2^{i-1} . Вага чисел, починаючи з $2^i + 2^{i-1} + 1$ і до 2^{i+1} , збігається з вагою чисел $2^i + 2^{i-1} - 1$, $2^i + 2^{i-1} - 2$ і т. д.

Кількість розрядів для подання числа AN дорівнює $\log_2(AN) = \log_2 A + \log_2 N$, де $\log_2 N$ – надлишковість коду. Таким чином, вибір модуля визначає не тільки надлишковість, але й відстань. Як модуль доцільно вибирати деяке взаємно просте з основою системи q число, що перевищує саму основу.

Можна припустити, що для двійкової системи $N=3$, і тоді будь-який код вигляду $A \cdot 3$ буде виявляти всі поодинокі помилки. Відповідно мінімальна надлишковість при довільній основі визначається як $\log_q(q+1)$, тобто завжди потрібно буде не менше одного, але й не більше двох додаткових розрядів.

Коди з мінімальною відстанню більшою за 2 , характеризуються величиною $M_q(N, d)$. Величина $M_q(N, d)$ – найменше число, яке при множенні його на N дає число, вага якого менша за d у поданні за основою q . Іншими словами, якщо число N мало вагу d у поданні за основою q , то добуток $N \cdot M_q(N, d)$ матиме вагу, меншу від d за цією самою основою q .

Якщо число A змінюється в межах $0 \leq A \leq M_q(N, d)$, то при будь-яких N і q мінімальна відстань $A \cdot N$ -коду буде дорівнювати щонайменше d , що впливає з певної кількості $M_q(N, d)$.

В теорії кодування доведено, що $M_2(N, 3) = (2^{(N-1)/2} + 1)/N$.

Основний спосіб для розрахунку значення M_q – спосіб безпосередніх обчислень (див. [1], С. 163).

1.6 Завдання на курсову роботу

У таблицю 1.3 введено такі умовні позначення та скорочення:

Ф.К. – фіксована кома;

П.К. – плаваюча кома;

КА – керуючий автомат;

r – модуль числа.

Таблиця 1.3 – Варіанти завдань на курсову роботу

Вар.	Операція	Форма подання операндів	Розрядність	Контрольні числа		Метод контролю	р	Тип КА	Тип елементів пам'яті
				А	В				
1	2	3	4	5	6	7	8	9	10
1	Додавання на суматорі оберненого коду	Ф.К.	16	179	91	Числовий	5	Мілі	D-тригер
2	Додавання на суматорі доповняльного коду	Ф.К.	32	-2341	587	AN	3	Мілі	JK-тригер
3	Множення на суматорі прямого коду, починаючи з молодших розрядів множника	Ф.К.	32	835	992	Числовий	7	Мілі	D-тригер
4	Множення на суматорі оберненого коду, починаючи з молодших розрядів множника	Ф.К.	16	78	-137	Цифровий	5	Мілі	RS-тригер
5	Множення на суматорі доповняльного коду, починаючи з молодших розрядів множника	Ф.К.	16	-223	51	AN	5	Мілі	JK-тригер
6	Множення на суматорі прямого коду, починаючи з молодших розрядів множника, з пропуском тактів додавання	Ф.К.	16	33	-434	Числовий	5	Мілі	D-тригер
7	Множення на суматорі прямого коду, починаючи зі старших розрядів множника, з пропуском тактів додавання	П.К.	64	-431	24	Числовий	7	Мура	D-тригер
8	Множення на суматорі оберненого коду, починаючи з молодших розрядів множника, з пропуском тактів додавання	П.К.	32	-934	56	AN	3	Мура	JK-тригер
9	Множення на суматорі доповняльного коду, починаючи зі старших розрядів множника, з пропуском тактів додавання	Ф.К.	32	433	-65	Цифровий	3	Мілі	RS-тригер
10	Множення на суматорі прямого коду, починаючи з молодших розрядів множника, з групуванням розрядів множника по два	П.К.	32	-95	555	Цифровий	9	Мура	RS-тригер

Продовження таблиці 1.3

1	2	3	4	5	6	7	8	9	10
11	Множення на суматорі оберненого коду, почин. з молод. розрядів множника, з групуванням розрядів множника по два	Ф.К.	16	-287	89	Цифровий	5	Мілі	RS-тригер
12	Множення на суматорі доповняльного коду, починаючи з молодших розрядів множника, з групуванням розрядів множника по два	П.К.	32	69	-1873	Числовий	5	Мура	D-тригер
13	Ділення на суматорі оберненого коду без відновлення остачі	Ф.К.	16	580	-58	Числовий	11	Мілі	D-тригер
14	Ділення на суматорі оберненого коду з відновленням остачі	П.К.	32	3933	437	AN	11	Мура	JK-тригер
15	Ділення на суматорі доповняльного коду з відновленням остачі	П.К.	64	-5852	532	Цифровий	5	Мура	RS-тригер
16	Додавання на суматорі доповняльного коду	П.К.	32	2432	794	Цифровий	3	Мура	RS-тригер
17	Додавання на суматорі оберненого коду	Ф.К.	16	67	225	Цифровий	7	Мілі	RS-тригер
18	Множення на суматорі прямого коду, починаючи з молодших розрядів множника	П.К.	64	314	48	Цифровий	9	Мура	RS-тригер
19	Множення на суматорі прямого коду, починаючи зі старших розрядів множника	Ф.К.	32	57	-923	AN	11	Мілі	JK-тригер
20	Множення на суматорі прямого коду, починаючи зі старших розрядів множника	П.К.	32	371	46	Числовий	3	Мура	D-тригер
21	Множення на суматорі оберненого коду, починаючи з молодших розрядів множника	П.К.	32	967	-173	AN	7	Мура	JK-тригер
22	Множення на суматорі оберненого коду, починаючи зі старших розрядів множника	Ф.К.	32	442	-224	Числовий	9	Мілі	D-тригер
23	Множення на суматорі оберненого коду, починаючи зі старших розрядів множника	П.К.	64	37	584	Цифровий	11	Мура	RS-тригер
24	Множення на суматорі доповняльного коду, починаючи з молодших розрядів множника	П.К.	32	-733	-38	Числовий	3	Мура	D-тригер

Продовження таблиці 1.3

1	2	3	4	5	6	7	8	9	10
25	Множення на суматорі доповняльного коду, починаючи зі старших розрядів множника	Ф.К.	32	-55	-569	Цифровий	7	Мілі	RS-тригер
26	Множення на суматорі доповняльного коду, починаючи зі старших розрядів множника	П.К.	64	359	-73	AN	9	Мура	JK-тригер
27	Множення на суматорі прямого коду, починаючи з молодших розрядів множника, з пропуском тактів додавання	П.К.	32	358	-85	Цифровий	3	Мура	RS-тригер
28	Множення на суматорі прямого коду, починаючи зі старших розрядів множника, з пропуском тактів додавання	Ф.К.	32	34	-375	AN	11	Мілі	JK-тригер
29	Множення на суматорі оберненого коду, починаючи зі старших розрядів множника, з пропуском тактів додавання	Ф.К.	32	-65	-432	Числовий	7	Мілі	D-тригер
30	Множення на суматорі оберненого коду, починаючи зі старших розрядів множника, з пропуском тактів додавання	П.К.	64	1118	-31	Цифровий	5	Мура	RS-тригер
31	Множення на суматорі доповняльного коду, починаючи з молодших розрядів множника, з пропуском тактів додавання	П.К.	32	94	-228	Числовий	11	Мура	D-тригер
32	Множення на суматорі прямого коду, починаючи з молодших розрядів множника, з групуванням розрядів множника по два	Ф.К.	16	475	-48	Числовий	7	Мілі	D-тригер
33	Множення на суматорі прямого коду, починаючи зі старших розрядів множника, з пропуском тактів додавання	Ф.К.	32	-666	-64	AN	11	Мілі	JK-тригер
34	Множення на суматорі оберненого коду, починаючи з молодших розрядів множника, з групуванням розрядів множника по два	П.К.	32	36	-321	AN	7	Мура	JK-тригер

Продовження таблиці 1.3

1	2	3	4	5	6	7	8	9	10
35	Множення на суматорі оберненого коду, починаючи зі старших розрядів множника, з групуванням розрядів множника по два	Ф.К.	32	-458	178	Числовий	9	Мілі	D-тригер
36	Множення на суматорі доповняльного коду, починаючи з молодших розрядів множника, з групуванням розрядів множника по два	Ф.К.	16	-375	-55	AN	7	Мілі	JK-тригер
37	Множення на суматорі доповняльного коду, починаючи зі старших розрядів множника, з групуванням розрядів множника по два	Ф.К.	32	-493	289	Цифровий	3	Мілі	RS-тригер
38	Ділення на суматорі оберненого коду без відновлення остачі	П.К.	32	3828	-348	Цифровий	5	Мура	RS-тригер
39	Ділення на суматорі доповняльного коду без відновлення остачі	Ф.К.	32	4734	789	AN	7	Мілі	JK-тригер
40	Ділення на суматорі оберненого коду з відновленням остачі	Ф.К.	16	2352	-168	Цифровий	9	Мілі	RS-тригер
41	Додавання на суматорі доповняльного коду	П.К.	64	278	-4834	Числовий	11	Мура	D-тригер
42	Множення на суматорі доповняльного коду, починаючи зі старших розрядів множника, з пропуском тактів додавання	П.К.	64	-32	-354	AN	5	Мура	JK-тригер
43	Множення на суматорі доповняльного коду, починаючи з молодших розрядів множника, з пропуском тактів додавання	Ф.К.	16	-173	122	AN	9	Мілі	JK-тригер
44	Додавання на суматорі оберненого коду	П.К.	32	793	-1332	AN	5	Мура	JK-тригер
45	Множення на суматорі прямого коду, починаючи зі старших розрядів множника, з групуванням розрядів множника по два	П.К.	64	1132	56	Числовий	3	Мура	D-тригер

Продовження таблиці 1.3

1	2	3	4	5	6	7	8	9	10
46	Множення на суматорі оберненого коду, починаючи зі старших розрядів множника, з групуванням розрядів множника по два	П.К.	64	-91	-321	Цифровий	11	Мура	RS-тригер
47	Множення на суматорі доповняльного коду, починаючи зі старших розрядів множника, з групуванням розрядів множника по два	П.К.	64	-435	757	АН	9	Мура	JK-тригер
48	Ділення на суматорі доповняльного коду без відновлення остачі	П.К.	64	4158	-594	Числовий	3	Мура	D-тригер
49	Ділення на суматорі доповняльного коду з відновленням остачі	Ф.К.	32	4225	325	Числовий	7	Мілі	D-тригер
50	Додавання на суматорі оберненого коду	Ф.К.	16	179	91	Числовий	5	Мілі	JK-тригер
51	Додавання на суматорі доповняльного коду	Ф.К.	32	2341	587	АН	3	Мура	D-тригер
52	Множення на суматорі прямого коду, починаючи з молодших розрядів множника	Ф.К.	32	835	992	Числовий	7	Мілі	RS-тригер
53	Множення на суматорі оберненого коду, починаючи з молодших розрядів множника	Ф.К.	16	78	-137	Цифровий	5	Мура	D-тригер

1.7 Приклад виконання завдання

Завдання: синтезувати операційний та керуючий автомати для виконання операції множення на суматорі оберненого коду, починаючи зі старших розрядів множника, з пропуском тактів додавання. Числа подані у формі з плаваючою комою. Тип керуючого автомата – Мура, розрядність операндів 64. Цифровий метод контролю за модулем 5. Контрольні числа : $A = 1118$, $B = -31$. Тип елементів пам'яті – RS-тригер.

Сформулюємо словесний алгоритм та набір необхідних для розробки пристроїв елементів

Необхідно

1. Два 56-тирозрядних реєстри для зберігання мантис операндів.
2. Два 8-мирозрядних реєстри для зберігання порядків операндів.

3. Суматор для додавання мантис операндів.
4. Суматор для додавання порядків операндів.
5. Схема формування оберненого коду.
6. Двійковий 6-ти розрядний лічильник для підрахунку кількості ітерацій алгоритму.
7. Схема порівняння.

Алгоритм

1. В шину вхідних даних надходять значення мантиси і порядку числа А. Під час цього отримуємо обернений код мантиси множеного.
 - А. Під час цього отримуємо обернений код мантиси множеного.
 2. Заносимо значення мантиси і порядку числа А в регістри RгАм та RгАп, відповідно.
 3. В шину вхідних даних надходять значення мантиси і порядку числа В. Під час цього отримуємо обернений код мантиси множника.
 4. Заносимо значення мантиси і порядку числа В у регістри RгВм та RгВп, відповідно.
 5. Заносимо в СМм та СМп значення 0.
 6. Якщо регістр RгАм = 0, то переходимо в пункт 15.
 7. Якщо регістр RгВм = 0, то переходимо в пункт 15.
 8. Присвоюємо суматору порядків значення суми регістрів порядків RгАп та RгВп. Заносимо в лічильник значення 55.
 9. Якщо множник RгВ від'ємний, то додаємо поправку $HCM := HCM + [RгАм]об$ і робимо зсув на один розряд вліво суматора СМм.
 10. Аналізуємо старший розряд RгВм[1], якщо він дорівнює 1, то до суматора СМм додаємо RгАм.
 11. Виконуємо зсув на один розряд вліво регістра RгВм та суматора СМм, зменшуємо значення лічильника на 1.
 12. Аналізуємо вміст лічильника, якщо він дорівнює 0, то переходимо в пункт 13, інакше переходимо в пункт 10.
 13. Виконуємо зсув на один розряд вправо суматора СМм.
 14. Якщо множник RгВ від'ємний, то додаємо поправку $HCM := HCM + RгАм$ і робимо зсув на один розряд вправо суматора RгАм.
 15. Видаємо результат СМм та СМп на шину даних.
 16. Кінець алгоритму.

Врахуємо особливості елементної бази при реалізації операційного пристрою, який використовує наведений алгоритм

Для реалізації множення чисел у формі з плаваючою комою, починаючи зі старших розрядів у множнику з пропуском тактів додавання, потрібні такі функціональні вузли:

Швх – для вхідних даних (множене і множник), які надходять в пристрій через шини вхідних даних;

Швих – для результату (добуток), який видається з пристрою через шини вихідних даних;

РгАм, РгВм – регістри для зберігання мантис операндів;

РгАп, РгВп – регістри для зберігання порядків операндів;

СМм – накопичувальний суматор оберненого коду для накопичення часткових добутоків мантис;

СМп – накопичувальний суматор для зберігання порядку результату;

СхФОК – схема формування оберненого коду для формування поправки (не)РгА;

ЛІЧ – лічильник для підрахування кількості кроків (ітерацій) множення.

Розробимо граф-схему наведеного алгоритму

Блок-схема алгоритму - орієнтований зв'язаний граф, в якому є одна початкова вершина, довільна кількість умовних та операційних вершин і одна кінцева вершина. Така блок-схема подана на рисунку 1.3.

Побудуємо структурну схему операційного автомата

Операційний автомат (ОА) служить для зберігання слів інформації, виконання набору мікрооперацій і обчислення значень логічних умов, тобто операційний автомат є структурою, організованою для виконання дій над інформацією. Мікрооперації, реалізовані операційним автоматом, ініціюються множиною керуючих сигналів $Y = \{y_1, \dots, y_M\}$, з кожним з яких ототожнюється певна мікрооперація. Значення логічних умов, що обчислюють в операційному автоматі, відображаються множиною інформаційних сигналів $X = \{x_1, \dots, x_L\}$, кожний з яких ототожнюється з певною логічною умовою. Для цього визначимо набір вхідних і вихідних сигналів (таблиця 1.4).

Структурна схема операційного автомата наведена на рисунку 1.4.

Приклад використання розробленого алгоритму

Виконаємо множення за розробленим алгоритмом для чисел $A = 1118_{10}$ та $B = -31_{10}$.

Запишемо машинне зображення мантис та порядків операндів в оберненому коді, відводячи один розряд на знак і 63 на модуль числа:

$$A_{m_{06}} = 0_10\dots010001011110_{55}; \quad B_{m_{06}} = 1_11\dots100000_{55};$$

$$A_{n_{06}} = 0_10\dots01011_{55}; \quad B_{n_{06}} = 0_10\dots0101_{55};$$

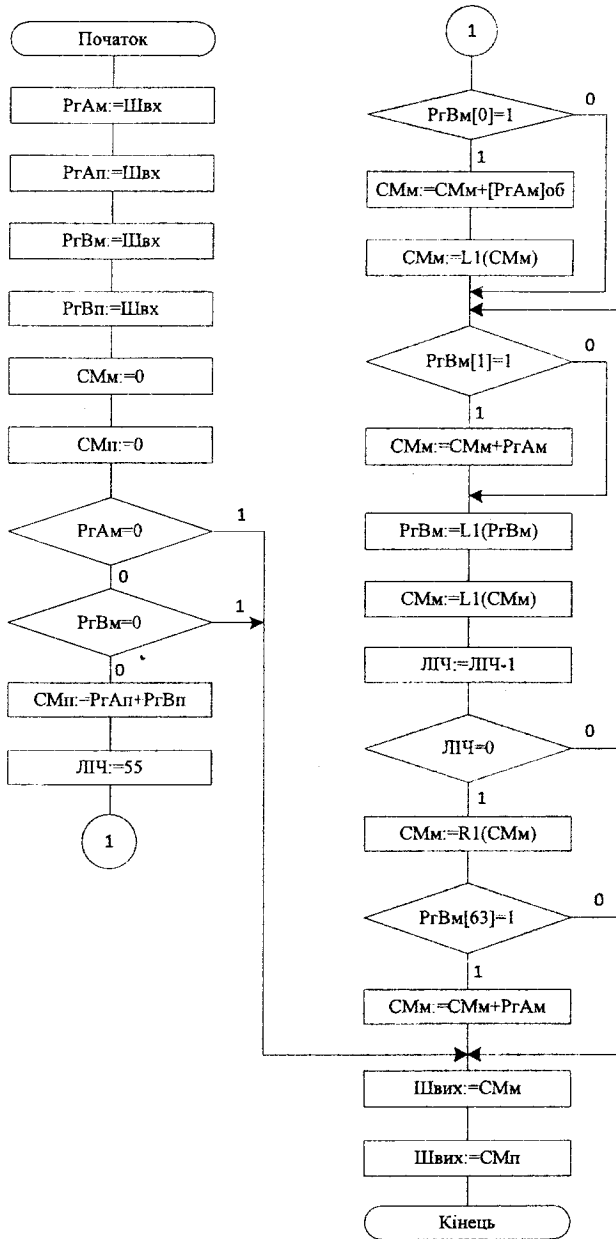


Рисунок 1.3 - Граф-схема алгоритму

Таблиця 1.4 - Набір вхідних і вихідних сигналів для структурної схеми операційного автомата

Вихідні сигнали		Вхідні сигнали	
Позначення	Виконувана дія	Позначення	Умова
Y1	$PrAm := Швх$	X1	$PrAm = 0$
Y2	$PrAn := Швх$	X2	$PrBm = 0$
Y3	$PrBm := Швх$	X3	$PrBm[63] = 1$
Y4	$PrBn := Швх$	X4	$PrBm[62] = 1$
Y5	$CMm := 0$	X5	$ЛПЧ = 0$
Y6	$CMn := 0$		
Y7	$CMn := PrAn + PrBn$		
Y8	$ЛПЧ := 63$		
Y9	$CMm := CMm + (не)PrAm$		
Y10	$CMm := L1(CMm)$		
Y11	$CMm := CMm + PrAm$		
Y12	$PrBm := L1(PrBm)$		
Y13	$ЛПЧ := ЛПЧ - 1$		
Y14	$CMm := R1(CMm)$		
Y15	$Швих := CMm$		
Y16	$Швих := CMn$		

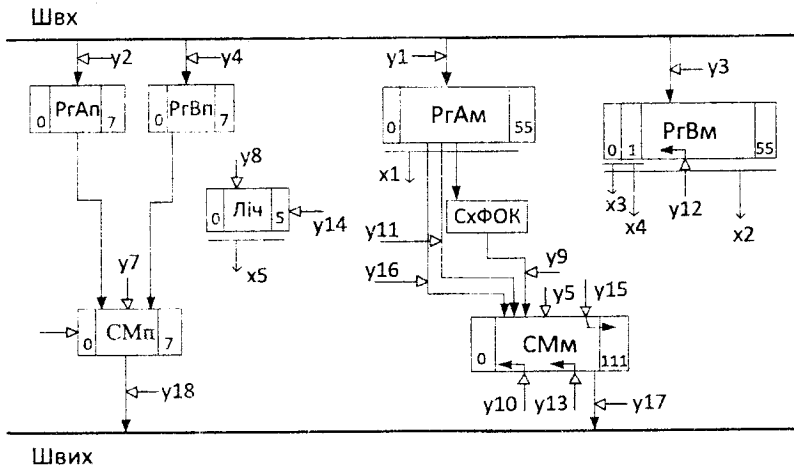


Рисунок 1.4 - Структурна схема операційного автомата

Виконаємо множення мантис (вміст регістрів і операції наведені в таблиці 1.5).

Таблиця 1.5 – Приклад виконання множення мантис

Сумагор СМм	Регістр PrAm	Регістр PrBm	Примітки
$I_1 \dots 111_{111}$	$0_1 \dots 010001011110_{55}$	$1_1 \dots 100000_{55}$	СМм:=0; PrAm:=Швх; PrBm:=Швх; ЛПЧ=1
$I_1 \dots 1011101000011_{111}$	$0_1 \dots 010001011110_{55}$	$1_1 \dots 100000_{55}$	СМм:=СМм+ +[PrA] ₀₆ ; СМм:=L1(СМм); ЛПЧ=1
$I_1 \dots 1011101000011_{111}$	$0_1 \dots 010001011110_{55}$	$1_1 \dots 1000001_{55}$	СМм:=СМм+ +PrA; СМм:=L1(СМм); PrB:=L1(PrB) ЛПЧ=2
Далі, згідно з алгоритмом, відбувається ще 54 кроків, аналогічних попередньому, в результаті яких вміст СМм не зміниться, а вміст регістра PrB зсується на 54 розряди вліво.			
$I_1 \dots 1011101000011_{111}$ І так далі...	$0_1 \dots 010001011110_{55}$	$1_1000001 \dots 1_{55}$	СМм:=СМм+ +PrA; СМм:=L1(СМм); PrB:=L1(PrB) ЛПЧ=55

$$СМп=11+5=16.$$

Отже, у результаті виконання операції на вихідну шину даних потраплять такі дані:

$$СМм=1_0 \dots 1_47 0111100010011101;$$

$$СМп=10000.$$

Якщо ці дані інтерпретувати у десяткову систему числення, то ми отримаємо число -34658.

$$-31 \cdot 1118 = -34658.$$

Отже, ми отримали правильний результат, що підтверджує правильність розробленого алгоритму.

Закодуємо граф-схему алгоритму виконання операції.

Кодовану граф-схему алгоритму множення наведено на рисунку 1.5.

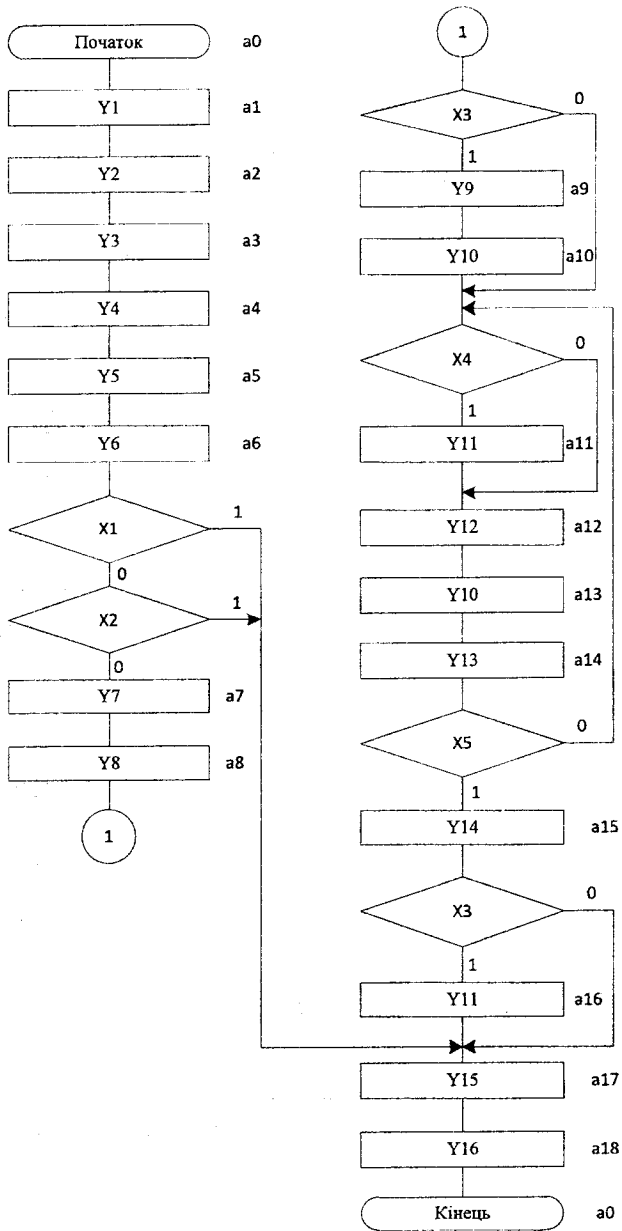


Рисунок 1.5 - Кодована граф-схема алгоритму

Побудуємо таблицю переходів та виходів автомата.

Таблиця переходів і виходів будується в такій послідовності.

В полі a_i графа t таблиці записуємо поточний стан автомата, в полі a_i графа $t+1$ – наступний його стан, у полі X – умову переходу зі стану $a_i(t)$ в стан $a_i(t+1)$ згідно з граф-схемою алгоритму. В полі Y записуються мікрооперації, які виконуються при переході автомата в наступний стан. У полі “Тригери” вказано сигнали, які необхідно подати на входи відповідних запам’ятовувальних елементів. Таким чином описуються всі можливі переходи автомата (таблиця 1.6).

Таблиця 1.6 – Таблиця переходів і виходів

t_i		t_{i+1}		Тригери									
				RS0		RS1		RS2		RS3		RS4	
x_i	a_i	a_{i+1}	y_i	R0	S0	R1	S1	R2	S2	R3	S3	R4	S4
—	a0	a1	y1	1	0	1	0	1	0	1	0	0	1
—	a1	a2	y2	0	0	0	0	0	0	0	1	1	0
—	a2	a3	y3	0	0	0	0	0	0	0	0	0	1
—	a3	a4	y4	0	0	0	0	0	1	1	0	1	0
—	a4	a5	y5	0	0	0	0	0	0	0	0	0	1
—	a5	a6	y6	0	0	0	0	0	0	0	1	1	0
$\overline{X1} * \overline{X2}$	a6	a7	y7	0	0	0	0	0	0	0	0	0	1
$X1 + X2$	a6	a16	y15	0	1	0	0	1	0	1	0	0	0
—	a7	a8	y8	0	0	0	1	1	0	1	0	1	0
$X3$	a8	a9	y9	0	0	0	0	0	0	0	0	0	1
$\overline{X3} * X4$	a8	a11	y11	0	0	0	0	0	0	0	1	0	1
$\overline{X3} * \overline{X4}$	a8	a12	y12	0	0	0	0	0	1	0	0	0	0
—	a9	a10	y10	0	0	0	0	0	0	0	1	1	0
$X4$	a10	a11	y11	0	0	0	0	0	0	0	0	0	1
$\overline{X4}$	a10	a12	y12	0	0	0	0	0	1	1	0	0	0
—	a12	a13	y10	0	0	0	0	0	0	0	0	0	1
—	a13	a14	y13	0	0	0	0	0	0	0	1	1	0
$\overline{X5} * X4$	a14	a11	y11	0	0	0	0	1	0	0	0	0	1
$\overline{X5} * \overline{X4}$	a14	a12	y12	0	0	0	0	1	0	0	1	0	0

Продовження таблиці 1.6

t_i		t_{i+1}		Тригери									
				RS0		RS1		RS2		RS3		RS4	
x_i	a_i	a_{i+1}	y_i	R0	S0	R1	S1	R2	S2	R3	S3	R4	S4
X5	a14	a15	y14	0	0	0	0	0	0	0	0	0	1
X3	a15	a16	y11	0	1	1	0	1	0	1	0	1	0
$\overline{X3}$	a15	a17	y15	0	1	1	0	1	0	1	0	0	0
—	a16	a17	y15	0	0	0	0	0	0	0	0	0	1
—	a17	a18	y16	0	0	0	0	0	0	0	1	1	0
—	a18	a0		1	0	0	0	1	0	0	0	0	0

Кількість тригерів, які необхідні для організації пам'яті керуючого автомата, визначається як найближче більше ціле від двійкового логарифму кількості станів тригера.

За отриманою таким чином таблицею записуються та зводяться до мінімальної форми функції збудження тригерів та функції виходів цифрового автомата.

Отже, для кодування станів автомата необхідно 5 RS-тригерів: RS0, RS1, RS2, RS3, RS4. Проаналізувавши таблицю 1.6, запишемо функції збудження тригерів:

$$R0 = a0 + a18,$$

$$S0 = a6 * (X1 + X2) + X3 * a15 + a15 * \overline{X3} = a6 * (X1 + X2) + a15,$$

$$R1 = a0 + a15 * X3 + a15 * \overline{X3} = a0 + a15,$$

$$S1 = a7,$$

$$R2 = a0 + a6 * (X1 + X2) + a7 + a14 * \overline{X5} * X4 + a15 * X3 + a15 * \overline{X3} + a14 * \overline{X5} * \overline{X4} + a18$$

$$= a0 + a6 * (X1 + X2) + a7 + a14 * X4 + a15 + a18$$

$$S2 = a3 + a8 * \overline{X3} * \overline{X4} + a10 * \overline{X4},$$

$$R3 = a0 + a3 + a6 * (X1 + X2) + a7 + a10 * \overline{X4} + a15 * X3 + a15 * \overline{X3}$$

$$= a0 + a3 + a6 * (X1 + X2) + a7 + a10 * \overline{X4} + a15$$

$$S3 = a1 + a5 + a8 * \overline{X3} * X4 + a9 + a13 + a17,$$

$$R4 = a1 + a3 + a5 + a7 + a9 + a13 + a15 * X3 + a17,$$

$$S4 = a0 + a2 + a4 + a6 * \overline{X1} * \overline{X2} + a8 * X3 + a8 * \overline{X3} * X4 + a10 * X4 + a12 + a14 * \overline{X5} * X4 + a14 * \overline{X5} * \overline{X4} + a14 * X5 + a16$$

$$= a0 + a2 + a4 + a6 * \overline{X1} * \overline{X2} + a8 * X3 + a8 * X4 + a10 * X4 + a12 + a14 + a16$$

Також запишемо функції виходів цифрового автомата, враховуючи тип автомата (Мура):

$$y_1 = a_0,$$

$$y_2 = a_1,$$

$$y_3 = a_2,$$

$$y_4 = a_3,$$

$$y_5 = a_4,$$

$$y_6 = a_5,$$

$$y_7 = a_6,$$

$$y_8 = a_7,$$

$$y_9 = a_8,$$

$$y_{10} = a_9 + a_{12},$$

$$y_{11} = a_8 + a_{10} + a_{14} + a_{15},$$

$$y_{12} = a_8 + a_{10} + a_{14},$$

$$y_{13} = a_{13},$$

$$y_{14} = a_{14},$$

$$y_{15} = a_6 + a_{15} + a_{16},$$

$$y_{16} = a_{17}.$$

Побудова керуючого автомата

Отримавши мінімальні форми функцій збудження тригерів та вихідних сигналів, будемо комбінаційну схему керуючого автомата (рисунок 1.6).

Далі вибирається система елементів, з яких будється схема автомата. У більшості схем як елементи пам'яті використовуються елементарні автомати (тригери), які є автоматами Мура і мають два стійких стани та можуть мати декілька фізичних входів.

Крім того, схема будь-якого керуючого автомата повинна містити певну кількість логічних елементів, що утворюють функціонально повну систему для синтезу необхідної комбінаційної схеми.

Також у нашій схемі буде використаний п'ятивходовий дешифратор, для вибору наступних станів автомата і вихідних сигналів за їх двійковим кодом, який подається з тригерів.

Виконання цифрового методу контролю за модулем 5

При цифровому контролі код числа утворюється діленням суми цифр числа на вибраний модуль:

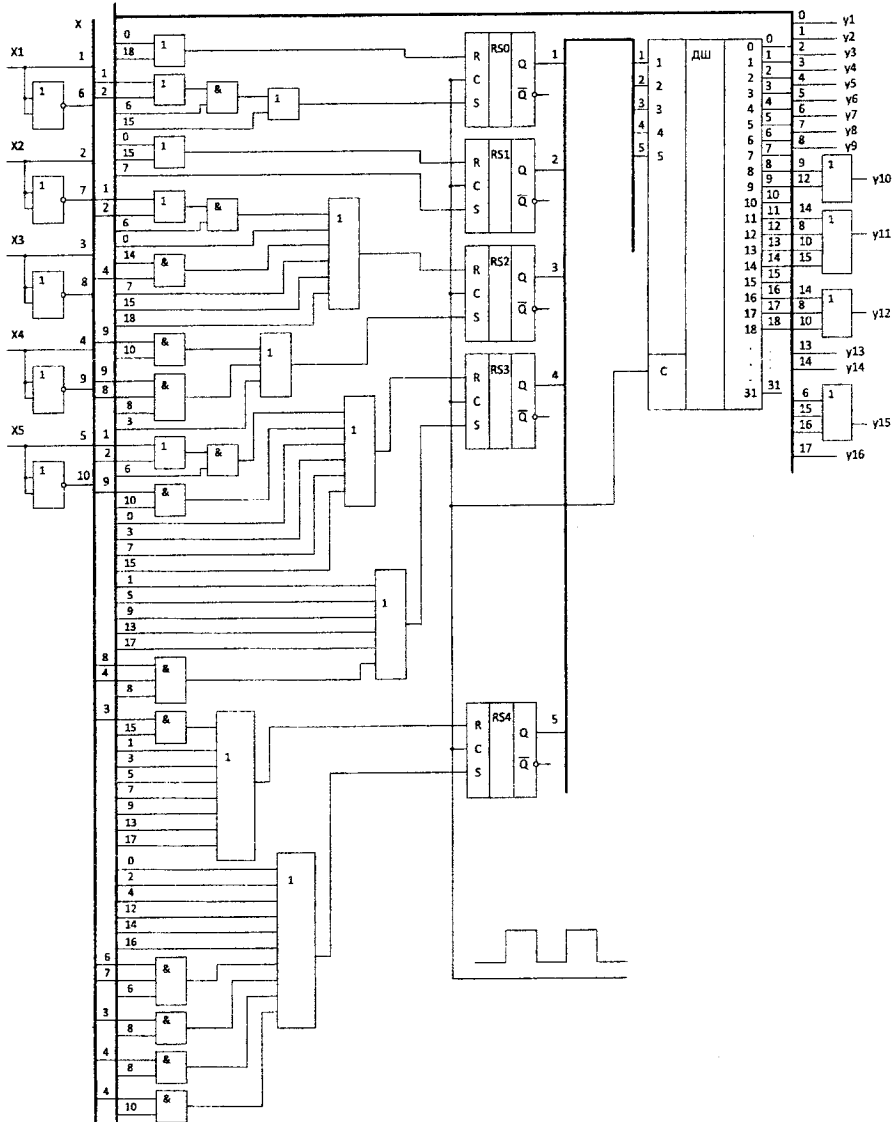


Рисунок 1.6 – Схема керуючого автомата

$$r'_A = \sum_i a_i - \left\{ \frac{\sum_i a_i}{p} \right\} p.$$

При додаванні чисел може виникати перенос у старші розряди, який необхідно враховувати за допомогою поправки, яка залежить від кількості переносів l :

$$r_c = (r_a + r_b - l(q-1)) \bmod p,$$

При виконанні операції віднімання можуть виникати запозичення зі старших розрядів у молодші, тоді також враховується поправка, яка залежить від кількості запозичень S :

$$r_c = (r_a + r_b + S(q-1)) \bmod p,$$

де p – модуль, за яким виконується контроль, у даному випадку він дорівнює 5.

Виконаємо цифровий контроль виконання операції множення за модулем $p = 7$ для заданих чисел ($A = 1118$, $B = -31$, $C = -34658$):

$$r'_a = (1+1+1+8) \bmod 5 = 1 \pmod{5},$$

$$r'_b = (1+3) \bmod 5 = 4 \pmod{5},$$

$$r'_c = r'_a(1*4 - 35*(10-1)) \bmod 5 = 4 \pmod{5},$$

$$r_c = (1*4 - 1(10-1)) \bmod 5 = 1 \pmod{5},$$

Оскільки $r'_c = r_c$, множення виконано правильно.

2 ОСНОВНІ ВИМОГИ ДО ВМІСТУ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ

Курсова робота складається з:

1) пояснювальної записки об'ємом двадцять п'ять - тридцять сторінок формату А4 шрифтом Times New Roman 14 кегль;

2) одного креслення формату А4 (А3).

При виконанні індивідуальних типових завдань КР текст пояснювальної записки не повинен повторюватись (електронний трафарет).

2.1 Вміст пояснювальної записки

Пояснювальна записка вміщує у собі:

- **титульний лист** (зразок оформлення подано в додатку А), на якому вказані: найменування міністерства, ВНЗ, інституту, факультету, кафедри, назва дисципліни, тема роботи, посада та прізвище керівника, прізвище та ініціали студента, номер групи, рік виконання роботи;

- **індивідуальне завдання** (дивись додаток Б);

- **анотація**;

- **зміст**, в якому вказано розділи пояснювальної записки та відповідні їм номери сторінок;

- **вступ**;

- **розділ 1 "Розробка алгоритму та операційного автомата"**;

- **розділ 2 "Синтез керуючого автомата"**;

- **розділ 3 "Виконання контролю арифметичної операції"**;

- **висновки**;

- **перелік посилань**;

- **додатки**.

2.2 Титульний лист

Титульний лист є першою сторінкою КР, яка не нумерується. Згідно з діючим стандартом на текстову конструкторську документацію ДСТУ 3008-95 титульний лист виконується за встановленим зразком (див. додаток А).

Для курсової роботи титульний лист виконується без рамки.

На титульному листі курсових робіт зазначаються:

- тема КР;

- запис „Пояснювальна записка ...” із вказанням спеціальності, цифрового коду кафедри та номера варіанта.

Предметна система умовних позначень для курсових робіт на кафедрах повинна мати таку структуру:

XX-XX.XX.XXX.XX.XXX XX
1 2 3 4 5 6

де

1 /XX-XX/ – числовий шифр кафедри, прийнятий у ВНТУ;

2 /.../ – умовний код дисципліни (скорочена назва дисципліни);

3 /XXX/ – порядковий номер індивідуального завдання на курсову роботу;

4 /XX/ – порядковий номер варіанта технічного завдання;

5 /XXX/ – три символи для позначень простих складових одиниць, що входять в основні складові одиниці курсової роботи;

6 /XX/ – код документа (BC, СК, ПЗ, КЗ, ЕЗ, Е5 тощо).

Приклад коду пояснювальної записки: 08-23.КЛ.013.00.000 ПЗ, де 013 – порядковий номер індивідуального завдання на курсову роботу.

Перераховується вчений ступінь та звання керівника. Також на титульному листі після захисту курсової роботи має бути виставлена оцінка з підписом викладача.

Робота, яка подана у вигляді ксерокопії чи сканованого матеріалу, до захисту не приймається.

2.3 Індивідуальне завдання

Індивідуальне завдання до змісту не вноситься, не нумерується та має бути другою сторінкою після титульного листа.

Індивідуальне завдання повинно містити дату видачі, а також підписи керівника та студента. Зразок оформлення індивідуального завдання наведено в додатку Б.

2.4 Анотація

Анотація – короткий зміст того, що необхідно виконати у курсовій роботі. Не повинна перевищувати 2-3 абзаци за обсягом. Заголовок „Анотація” пишуть посередині рядка великими літерами з більш високою насиченістю шрифту.

2.5 Зміст

Зміст розташовують безпосередньо після індивідуального завдання, починаючи з нової сторінки. До змісту включають: перелік умовних позначень, символів, одиниць, скорочень і термінів; вступ; послідовно перелічені назви всіх розділів, підрозділів, пунктів і підпунктів (якщо вони мають заголовки) суті проєкту; висновки; рекомендації; перелік посилань;

назви додатків і номери сторінок, які містять початок матеріалу. У змісті можуть бути перелічені номери й назви ілюстрацій та таблиць з зазначенням сторінок, на яких вони розташовані. Зміст є четвертим аркушем (титульний лист, індивідуальне завдання та анотація не нумеруються).

Назви заголовків змісту повинні однозначно відповідати назвам заголовків пояснювальної записки за текстом. Нумерація сторінок повинна бути наскрізною. Форми подачі розділів та підрозділів в змісті для курсових робіт показані нижче.

- 1 Аналіз ...
 - 1.1 Огляд ...
 - 1.1.1 ...
- 2 Заголовок другого розділу
 - 2.1 Заголовки підрозділів
 - 2.1.1 ...
- 3 Заголовок третього розділу
 - 3.1 Заголовки підрозділів
 - 3.1.1 ...

Обсяг пояснювальної записки враховується без додатків.

2.6 Вступ

Вступ пишуть з нової пронумерованої сторінки з заголовком „Вступ” посередині (ДСТУ 3008-95) великими літерами з більш високою насиченістю (жирністю) шрифту. Обсяг вступу 1 - 2 сторінки.

Текст вступу повинен бути коротким і висвітлювати питання актуальності, значення, сучасний рівень і призначення курсової роботи. У вступі і далі за текстом не дозволяється використовувати скорочені слова, терміни, крім загальноприйнятих.

Вступ висвітлює:

- стан розвитку проблеми в даній галузі, до якої має відношення розробка;
- галузь використання та призначення;
- мету та загальну постановку задачі;
- актуальність, яка повинна подаватись в останньому абзаці вступу, з метою стислого викладання суті розробки цього напрямку.

2.7 Висновки

Висновки оформлюють з нової пронумерованої сторінки посередині (ДСТУ 3008-95) великими буквами більш високої насиченості.

Висновки є заключною частиною, підсумком прийнятих рішень виконаної роботи.

2.8 Перелік посилань

Форма запису "Перелік посилань" відповідає формі запису вступу, основної частини та висновків та повинна бути оформлена згідно з ДСТУ 3582-97.

Список містить перелік літературних джерел, на які повинні бути обов'язкові посилання в тексті пояснювальної записки. Література (книги, статті, патенти, журнали) в загальний список записується в порядку посилання на неї в тексті. Посилання на літературу наводять в квадратних дужках [...], вказуючи порядковий номер за списком.

Літературу записують мовою оригіналу. В списку кожне літературне джерело записують з абзацу, нумерують арабськими цифрами, починаючи з одиниці (далі показано).

1. Прізвище та ініціали. Назва книги. – Місце видання: Видавництво, рік. – Кількість сторінок с.

(1. Максимович Н. Г. *Теорія графів і електричних кіл.* / Максимович Н. Г. – Львів : Вища школа, 1987. – 216 с.)

2. Назва книги / Прізвище та ініціали. – Місце видання: Видавництво, рік. – Кількість сторінок с.

Примітка. Великі міста такі, як Київ, Москва дозволяється записувати однією великою буквою з крапкою.

(2. *Вимірювання і комп'ютерно-вимірювальна техніка: Навч. посібник* / В. О. Поджаренко, В. В. Кухарчук. – К. : НМК ВО, 1991. – 240 с.)

3. Прізвище та ініціали. Назва частини книги // Прізвище та ініціали. Назва книги. – Місце видання: Видавництво, рік. – С. Інтервал сторінок.

(3. Хоор К. *О структурной организации данных* // Дал У., Дейкстра Э., Хоор К. *Структурное программирование.* – М. : Мир, 1975. – С. 98-197.)

4. Прізвище та ініціали. Назва частини видання // Назва видання. – рік. – № Число. – С. Інтервал сторінок.

(4. Dreiheller A. *Programming Language Incorporating Units of Measure* // *Informationstechnik.* – 1997. – № 1. – P. 83-88.)

(5. Ершов А. А. *Стабильные методы оценки параметров* // *Автоматика и телемеханика.* – 1978. – № 8. – С. 86-91.)

5. Нормативно-технічні та патентні документи.

(6. ГОСТ 7.9-77. *Реферат и аннотация.* – М. : Издательство стандартов, 1981. – 6 с.)

2.9 Додатки

У курсовій роботі у вигляді додатків можуть подаватися креслення, граф-схеми та таблиці. Додатки оформлюють як продовження документа на його наступних сторінках, розташовуючи в порядку посилань на них у тексті ПЗ.

Посилання на додатки в тексті ПЗ дають за формою:

“... наведено в додатку А”, „... наведено на кресленні В.5 ” або (додаток Б); (додатки В, Г).

Кожен додаток необхідно починати з нової сторінки, вказуючи зверху посередині рядка слово “Додаток” і через пропуск його позначення. Додатки позначають послідовно великими українськими буквами, за винятком букв І, Є, З, І, Ї, О, Ч, Ь, наприклад, *Додаток А, Додаток Б*.

Під позначенням для обов’язкового додатку пишуть в дужках слово (*обов’язковий*), а для інформативного – (*довідниковий*).

Кожен додаток повинен мати тематичний (змістовний) заголовок, який записують посередині рядка малими буквами починаючи з великої.

Нумерація аркушів документа і додатків, які входять до його складу, повинна бути наскрізна.

Всі додатки включають у зміст, вказуючи номер, заголовок і сторінки, з яких вони починаються.

3 ОСНОВНІ ВИМОГИ ДО ОФОРМЛЕННЯ ТЕКСТУ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ

3.1 Загальні правила

При оформленні текстової частини КР необхідно дотримуватись вимог ДСТУ 3008-95.

Текст ПЗ виконується відповідно до вимог ГОСТ 2.105-95 із застосуванням друкувальних та графічних пристроїв виведення ЕОМ розміром не менше 2,5 мм (редактор Word, шрифт – Times New Roman 14 кегль) через 1 інтервал.

Допускається текст оформлювати машинописним (друкарським) чітким шрифтом (1,5-2 інтервали) або рукописним основним креслярським шрифтом за ГОСТ 2.304-81 з висотою букв та цифр не менше 2,5 мм.

Допускається оформлення пояснювальної записки КР за вимогами ЄСКД. Пояснювальна записка, яка викладається технічною мовою, відноситься до текстових документів. Графічна інформація має подаватись у вигляді ілюстрацій (схеми, графіки, діаграми тощо). Цифрова – у вигляді таблиць.

Відступи від країв аркуша: зверху, знизу і зліва – 20 мм; справа – 10 мм. Абзац – 5 знаків.

Нумерація сторінок в правому верхньому куту, починаючи зі змісту.

3.2 Вимоги до оформлення розділів та підрозділів

Структурними елементами основної частини є розділи, підрозділи, пункти, підпункти, переліки та примітки.

Розділ – головний ступінь поділу тексту, позначений номером і має заголовок.

Підрозділ – частина розділу, позначена номером і має заголовок.

Пункт – частина розділу чи підрозділу, позначена номером і може мати заголовок.

Підпункт – частина пункту, позначена номером і може мати заголовок.

Заголовки структурних елементів необхідно нумерувати тільки арабськими цифрами.

Допускається розташовувати текст між заголовками розділу і підрозділу, між заголовками підрозділу і пункту.

Кожен розділ рекомендується починати з нової сторінки.

Заголовок розділу записують посередині (ДСТУ 3008-95) великими літерами з більш високою насиченістю.

Заголовки підрозділів, пунктів та підпунктів (при наявності заголовка) записують з абзацу малими буквами починаючи з великої.

Розділи нумерують порядковими номерами в межах всього документа (1, 2, і т. д.). Після номера крапку не ставлять, а роблять один пропуск.

Підрозділи нумерують в межах кожного розділу, пункти в межах підрозділу і т. д. за формою (3.1, 3.2, 3.2.1, 3.2.2, 3.2.2.1 і т. д.).

Цифри, які вказують номер, не повинні виступати за абзац.

Посилання в тексті на розділи виконується за формою: "... наведено в розділі 3".

В тексті документа може наводитись перелік, який рекомендується нумерувати малими літерами українського алфавіту з дужкою або виділяти дефісом перед текстом. Для подальшої деталізації переліку використовують арабські цифри з дужкою.

Кожну частину переліку записують з абзацу, починаючи з малої букви і закінчуючи крапкою з комою, в кінці останньої ставлять крапку.

Приклад:

а) текст переліку та його...

продовження;

б) текст переліку:

1) текст переліку подальшої деталізації та його продовження;

2) ...;

в) останній перелік.

3.3 Правила написання тексту

При написанні тексту слід дотримуватися таких правил:

а) текст необхідно викладати лаконічним технічним стилем;

б) умовні буквені позначення фізичних величин і умовні графічні позначення компонентів повинні відповідати установленим в стандартах. Перед буквеним позначенням фізичної величини повинно бути її пояснення (*резистор R, конденсатор C*);

в) числа з розмірністю слід записувати цифрами, а без розмірності словами (*відстань – 2 мм, відміряти три рази*);

г) позначення одиниць слід писати в рядок з числовим значенням без перенесення на наступний рядок. Між останньою цифрою числа і позначенням одиниці слід робити пропуск (*100 Вт, 2 А*);

д) якщо наводиться ряд числових значень однієї і тієї самої фізичної величини, то одиницю фізичної величини вказують тільки після останнього числового значення (*1,5; 1,75; 2 мм*);

е) позначення величин з граничними відхиленнями слід записувати так: *100 ± 5 мм*;

ж) при написанні одиниць, знак множення замінюють крапкою на середній лінії (*·*); знак ділення замінюють косою рисою (*/*);

и) порядкові числівники слід записувати цифрами з відмінковими закінченнями (*9-й день, 4-а лінія*); при кількох порядкових числівниках

відмінкове закінчення записують після останнього (3,4,5-й графіки); кількісні числівники записують без відмінкових закінчень (на 20 аркушах); не пишуть закінчення в датах (21 жовтня) та при римських числах (XXI століття);

к) скорочення слів в тексті не допускаються, крім загальноприйнятих в українській мові і установлених в стандарті ГОСТ 2.316-68, а також скорочень, які прийняті для надписів на виробі (в тексті вони повинні бути виділені великим шрифтом: **ON**, **OFF**), якщо надпис складається з цифр або знаків, то в лапках. Лапками також виділяють найменування команд, режимів, сигналів (“**Запуск**”);

л) дозволяється виконувати записи математичних виразів за формою:

$$\frac{ABC}{DE} = ABC/DE;$$
$$A_1 = A1 = A(1);$$
$$\pm 20^{\circ}\text{C} = + - 20^{\circ}\text{C} = + - 20 \text{ ЦЕЛ};$$
$$200 = 200 (+20; -30);$$

м) не дозволяється:

- допускати написання професійних або місцевих слів і виразів (техніцизмів);

- після назви місяця писати слово “місяць” (не “в травні місяць”, а “в травні”);

- використовувати вирази: “цього року”, “минулого року”, слід писати конкретну дату “в червні 2001 року”;

- використовувати позначення одиниць фізичних величин без цифр, необхідно писати повністю: “кілька кілограмів” (за винятком оформлення таблиць і формул);

- поєднувати текст з умовним позначенням фізичних величин за допомогою математичних знаків (не “швидкість = 5 км/год”, а “швидкість дорівнює 5 км/год”, не “температура дорівнює - 5° С”, а “температура дорівнює мінус 5° С”);

- використовувати математичні знаки <>, 0, №, %, sin, cos, tg, log та ін. без цифрових або буквених позначень. В тексті слід писати словами “нуль”, “номер”, “логарифм” і т. д.;

- використовувати індекси стандартів (ДСТУ, СТП) без реєстраційного номера.

3.4 Оформлення формул

Кожну формулу записують з нового рядка, симетрично до тексту. Між формулою і текстом пропускають один рядок.

Умовні буквені позначення (символи) в формулі повинні відповідати установленим у міждержавному стандарті ГОСТ 1494-77. Їх пояснення наводять в тексті або зразу ж під формулою. Для цього після формули

ставляють кому і записують пояснення до кожного символу з нового рядка в тій послідовності, в якій вони наведені у формулі, розділяючи крапкою з комою. Перший рядок повинен починатися з абзацу з слова “де” і без будь-якого знака після нього.

Всі формули нумерують в межах розділу арабськими числами. Номер вказують в круглих дужках з правої сторони, в кінці рядка, на рівні закінчення формули. Номер формули складається з номера розділу і порядкового номера формули в розділі, розділених крапкою. Дозволяється виконувати нумерацію в межах всього документа.

Приклад

Таким чином, момент тертя в кернових опорах

$$M_m = -\kappa G^{1,5}, \quad (3.1)$$

де κ – коефіцієнт пропорційності;

G – вага рухомої частини вимірювального механізму.

Одиницю вимірювання, при необхідності, беруть в квадратні дужки

$$I = \frac{U}{R} [A]. \quad (3.2)$$

Числову підстановку і розрахунок виконують з нового рядка не нумеруючи. Одиницю вимірювання беруть в круглі дужки. Наприклад,

$$I = \frac{220}{100} = 2,2 (A).$$

Розмірність одного й того самого параметра в межах документа повинна бути однаковою.

Якщо формула велика, то її можна переносити на наступні рядки. Перенесення виконують тільки математичними знаками, повторюючи знак на початку наступного рядка. При цьому знак множення “ \cdot ” замінюють знаком “ \times ”.

Формула є частиною речення, тому до неї застосовують такі ж правила граматики, як і до інших членів речення. Якщо формула знаходиться в кінці речення, то після неї ставлять крапку. Формули, які слідує одна за одною і не розділені текстом, розділяють комою.

Посилання на формули в тексті дають в круглих дужках за формою: “... в формулі (5.2)”; “... в формулах (5.7, ..., 5.10)”.

3.5 Оформлення ілюстрацій

Для пояснення викладеного тексту рекомендується його ілюструвати графіками, кресленнями, фрагментами схем та ін., які можна виконувати чорною тушшю, простим олівцем середньої твердості та комп'ютерною графікою. Розміщують ілюстрації в тексті або в додатках.

В тексті ілюстрацію розміщують симетрично до тексту після першого посилання на неї або на наступній сторінці, якщо на даній вона не уміщується без повороту.

На всі ілюстрації в тексті ПЗ мають бути посилання. Посилання виконують за формою: "...показано на рисунку 3.1." або в дужках за текстом (рисунком 3.1), на частину ілюстрації: "... показані на рисунку 3.2, б)". Посилання на раніше наведені ілюстрації дають зі скороченим словом *дивись* (див. рисунок 1.3).

Наведена форма запису (Рисунок ...) відповідає вимогам ГОСТ 2.105-95. ДСТУ 3008-95 допускає скорочення лише в посиланнях, тобто замість „рисунок ...” – „рис. ...”. Між ілюстрацією і текстом пропускають один рядок.

Всі ілюстрації в ПЗ називають рисунками і позначають під ілюстрацією симетрично до неї за такою формою: “Рисунок 3.5 – Найменування рисунка”. Крапку в кінці не ставлять, знак переносу не використовують. Якщо найменування рисунка довге, то його продовжують у наступному рядку, починаючи від найменування.

Нумерують ілюстрації в межах розділів, вказуючи номер розділу і порядковий номер ілюстрації в розділі розділяючи крапкою. Дозволяється нумерувати в межах всього документа.

Пояснювальні дані розміщують під ілюстрацією над її позначенням.

У випадку, коли ілюстрація складається з частин, їх позначають малими буквами українського алфавіту з дужкою (а), (б) під відповідною частиною. В такому випадку після найменування ілюстрації ставлять двокрапку і дають найменування кожної частини за формою:

а – найменування першої частини; б – найменування другої частини
або за ходом найменування ілюстрації, беручи букви в дужки:

Рисунок 3.2 - Структурна схема (а) і часові діаграми (б) роботи фазометра

Якщо частини ілюстрації не вміщуються на одній сторінці, то їх переносять на наступні сторінки. В цьому випадку на першій сторінці вказують назву ілюстрації, а під її продовженнями позначають “Рисунок ..., аркуш...”. Пояснювальні дані розміщують під кожною частиною ілюстрації.

Якщо в тексті є посилання на складові частини зображеного засобу, то на відповідній ілюстрації вказують їх порядкові номери в межах ілюстрації.

Якщо ілюстрація є фрагментом повної розробленої схеми, то для всіх компонентів вказують ті позиційні позначення, які вказані на схемі.

Якщо ілюстраціями є фотографії, то останні повинні бути наклеєні на стандартні аркуші білого паперу і позначені як рисунки.

3.6 Оформлення таблиць

Таблицю розміщують симетрично до тексту після першого посилання на даній сторінці або на наступній, якщо на даній вона не вміщується, і таким чином, щоб зручно було її розглядати без повороту, або з поворотом на кут 90° за годинниковою стрілкою.

На всі таблиці мають бути посилання за формою: “*наведено в таблиці 3.1*”; “... *в таблицях 3.1 – 3.5*” або в дужках по тексту (*таблиця 3.6*). Посилання на раніше наведену таблицю дають з словом *дивись* (*див. таблицю 2.4*) за ходом чи в кінці речення.

Таблицю розділяють на графи (колонки) і рядки. В верхній частині розміщують шапку таблиці, в якій вказують найменування граф. Діагональне ділення шапки таблиці не допускається. Ліву графу (боковик) часто використовують для найменування рядків. Допускається не розділяти рядки горизонтальними лініями. Мінімальна відстань між основами рядків – 8 мм. Розміри таблиці визначаються об'ємом матеріалу.

Графу “№ п/п” в таблицю не включають. При необхідності нумерації номери вказують в боковику таблиці перед найменуванням рядка.

Найменування граф може складатися із заголовків і підзаголовків, які записують в однині, симетрично до тексту графи малими літерами, починаючи з великої. Якщо підзаголовок складає одне речення із заголовком, то в цьому випадку його починають з малої букви. В кінці заголовків і підзаголовків граф таблиці крапку не ставлять. Дозволяється заголовки і підзаголовки граф таблиці виконувати через один інтервал.

Якщо всі параметри величин, які наведені в таблиці, мають одну й ту саму одиницю фізичної величини, то над таблицею розміщують її скорочене позначення (*мм*). Якщо ж параметри мають різні одиниці фізичних величин, то позначення одиниць записують в заголовках граф після коми (*Довжина, мм*).

Текст заголовків і підзаголовків граф може бути замінений буквеними позначеннями, якщо тільки вони пояснені в попередньому тексті чи на ілюстраціях (*D* - діаметр, *H* - висота і т. д.). Однакові буквени позначення групують послідовно в порядку росту їх індексів, наприклад: (*L₁, L₂, ...*).

Найменування рядків записують в боковик у таблиці у вигляді заголовків в називному відмінку однини, малими буквами, починаючи з великої і з однієї позиції. В кінці заголовків крапку не ставлять. Позначення одиниць фізичних величин вказують в заголовках після коми.

Для опису визначеного інтервалу значень в найменуваннях граф і рядків таблиці можна використовувати слова: “більше”, “менше”, “не більше”, “не менше”, “в межах”. Ці слова розміщують після одиниці фізичної величини (*Напруга, В, не більше*), а також використовують слова “від”, “більше”, “до” (*Від 10 до 15; більше 15; до 20*).

Дані, що наводяться в таблиці, можуть бути словесними і числовими.

Слова записують в графах з однієї позиції. Якщо рядки таблиці не розділені лініями, то текст, який повторюється і складається з одного слова, дозволяється замінювати лапками ("). Якщо текст складається з двох і більше слів, то при першому повторенні його замінюють словами “*те ж*”, а далі лапками. При розділенні таблиці горизонтальними лініями – ніякої заміни не виконують.

Числа записують посередині графи так, щоб їх однакові розряди по всій графі були точно один під одним, за винятком випадку, коли вказують інтервал. Інтервал вказують від меншого числа до більшого з тире між ними:

$$12 - 35$$
$$122 - 450.$$

Дробові числа наводять у вигляді десяткових дробів з однаковою кількістю знаків після коми в одній графі. Розміри в дюймах можна записувати у вигляді: $1/2"$, $1/4"$, $1/8"$.

Ставити лапки замість цифр чи математичних символів, які повторюються, не можна. Якщо цифрові чи інші дані в таблиці не наводяться, то ставиться прочерк.

Таблиці нумерують в межах розділів і позначають зліва над таблицею за формою: “*Таблиця 4.2 – Найменування таблиці*”. Крапку в кінці не ставлять. Якщо найменування таблиці довге, то продовжують у наступному рядку починаючи від слова “*Таблиця*”. Номер таблиці складається з номера розділу і порядкового номера таблиці в розділі, розділених крапкою. Дозволяється нумерувати в межах всього документа.

Таблиця може мати велику кількість граф і рядків. У таких випадках таблицю розділяють на частини і переносять на інші сторінки.

Якщо частини таблиці розміщують поряд, то в кожній частині повторюють головку таблиці, а при розміщенні однієї частини під іншою – повторюють боковик.

Якщо в кінці сторінки таблиця переривається і її продовження буде на наступній сторінці, в першій частині таблиці нижню горизонтальну лінію, що обмежує таблицю, не проводять.

При перенесенні частин таблиці на інші сторінки, повторюють або продовжують найменування граф. Допускається виконувати нумерацію граф на початку таблиці і при перенесенні частин таблиці на наступні сторінки повторювати тільки нумерацію граф.

У всіх випадках найменування (при його наявності) таблиці розміщують тільки над першою частиною, а над іншими частинами зліва пишуть “Продовження таблиці 4.2” без крапки в кінці.

Інші вимоги до виконання таблиць – відповідно до чинних стандартів на технічну документацію.

ДОДАТКИ

Додаток А
Зразок титульного аркуша курсової роботи

Міністерство освіти і науки України
Вінницький національний технічний університет
Кафедра обчислювальної техніки

КУРСОВА РОБОТА

з дисципліни «Комп'ютерна логіка»
на тему «Синтез операційного пристрою»

Студента 2 курсу _____ групи
напряму підготовки «бакалавр –
Комп'ютерна інженерія» спеціальності
«Комп'ютерні системи та мережі»

(прізвище та ініціали)

Керівник: к.т.н., доц. каф ОТ Біліченко Н. О.

Національна шкала _____

Кількість балів: _____

Оцінка: ECTS _____

Члени комісії

_____ (підпис) _____ (прізвище та ініціали)

_____ (підпис) _____ (прізвище та ініціали)

_____ (підпис) _____ (прізвище та ініціали)

м. Вінниця – _____ рік

Додаток Б
Зразок індивідуального завдання

Міністерство освіти і науки України
Вінницький національний технічний університет
Інститут інформаційних технологій і комп'ютерної інженерії

ЗАТВЕРДЖУЮ

Зав. кафедри ОТ, проф., д.т.н.
_____ О. Д. Азаров
« ___ » _____ 20__ р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на курсову роботу з дисципліни «Комп'ютерна логіка»,
варіант № _____

студенту _____ групи _____

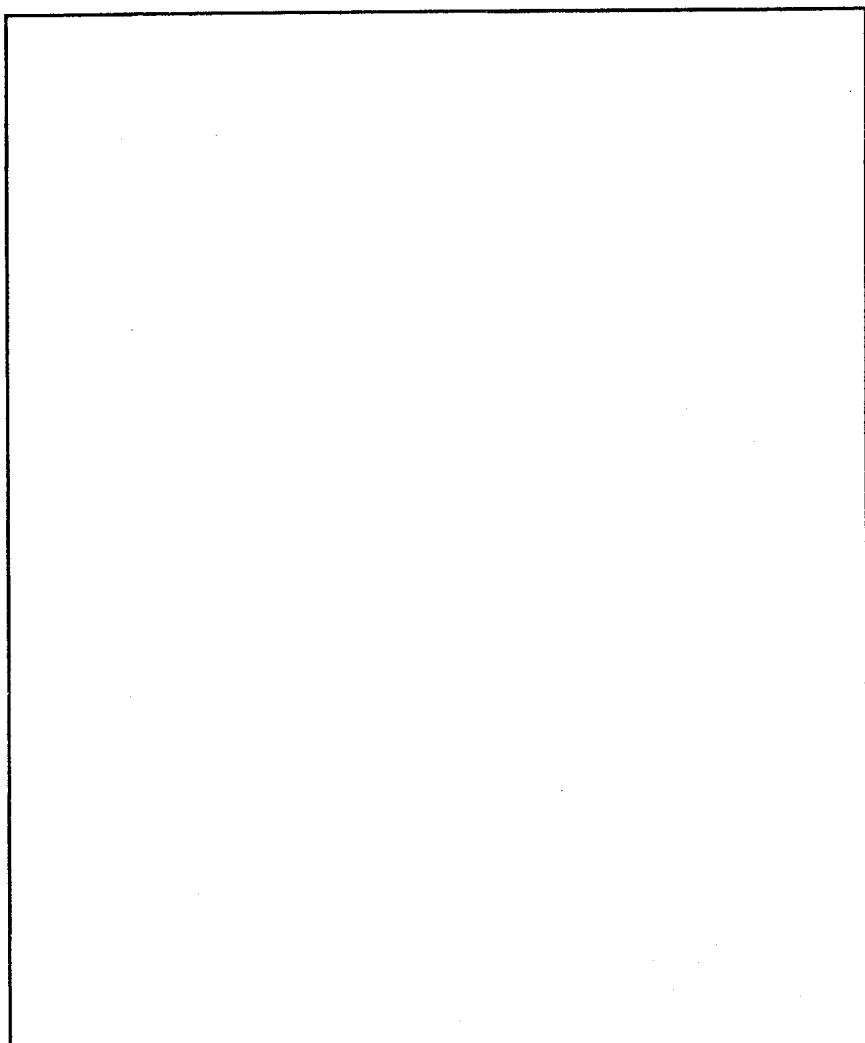
Тема «Синтез операційного та керуючого автоматів»

1. Розробити словесний алгоритм та граф-схему виконання операції множення на суматорі оберненого коду, починаючи зі старших розрядів множника, з пропуском тактів додавання. Вхідні дані:
 - а) форма подання операндів: плаваюча кома;
 - б) розрядність: 64;
 - в) контрольні числа: $A = 1118$, $B = -31$.
2. Синтезувати операційний та керуючий автомати для виконання заданої операції. Вхідні дані:
 - а) тип керуючого автомата: Мура;
 - б) тип елементів пам'яті: RS-тригер.
3. Описати методику контролю заданої операції за допомогою цифрового контролю за модулем 5. Навести приклад на контрольних числах.

Дата видачі « ___ » _____ 20__ р. Керівник _____ ПІБ викладача

Завдання отримав _____ ПІБ студента

Додаток В
Зразок штампа для схеми керуючого автомата



						08-23.КЛ.0XX.00.000 Е2		
Змн.	Лист	№ докум.	Підпис	Дата	Автомат керуючий. Схема електрична функціональна	Лт	Арк	Аркхів
Розроб.	Прізвище	Прізвище						
Перевір.	Прізвище							
Реценз.								
Н. контр.								
Затверд.						група		

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

Основна:

1. Савельев А. Я. Прикладная теория цифровых автоматов. / Савельев А. Я. – Москва : ВШ, 1987. – 270 с.
2. Самофалов К. Г. Прикладная теория цифровых автоматов. / Самофалов К. Г. – Київ : ВШ, 1987. – 375 с.
3. Каган Б. М. Электронные вычислительные машины и системы. / Каган Б. М. – Москва : Энергоатомиздат, 1991 г. – 592 с.
4. Соловьев Г. Н. Арифметические устройства ЭВМ. / Соловьев Г. Н. – М. : Энергия, 1978. – 177 с.

Додаткова:

1. Карцев М. А. Арифметика цифровых машин. / Карцев М. А. – М. : Наука, 1969. – 576 с.
2. Самофалов К. Г. Электронные цифровые вычислительные машины. / Самофалов К. Г., Корнейчук В. И., Тарасенко В. П. – К. : Вища школа, 1983. – 455 с.
3. Самофалов К. Г. Цифровые ЭВМ: Теория и проектирование. / Самофалов К. Г., Корнейчук В. И., Тарасенко В. П. – К. : Вища школа, 1989. – 424 с.

Інструктивно-методичне видання

Методичні вказівки до виконання курсової роботи
з дисципліни «Комп'ютерна логіка»
для студентів напряму підготовки 6.050102 «Комп'ютерна
інженерія»

Редактор В. Дружиніна
Коректор З. Поліщук

Укладачі : Наталія Олександрівна Біліченко
Дмитро Васильович Кисюк

Оригінал-макет підготовлено укладачами

Підписано до друку 19.08.2013 р.
Формат 29,7 × 42¼. Папір офсетний.
Гарнітура Times New Roman.
Друк різнографічний. Ум. друк. арк. 3,2.
Наклад 75 прим. Зам. № 2013-125.

Вінницький національний технічний університет,
навчально-методичний відділ ВНТУ,
21021, м. Вінниця, Хмельницьке шосе, 95,
ВНТУ, к. 2201.
Тел. (0432) 59-87-36.
Свідоцтво суб'єкта видавничої справи
серія ДК № 3516 від 01.07.2009 р.

Віддруковано у Вінницькому національному технічному університеті
в комп'ютерному інформаційно-видавничому центрі
21021, м. Вінниця, Хмельницьке шосе, 95,
ВНТУ, к. 114.
Тел. (0432) 59-87-38.
Свідоцтво суб'єкта видавничої справи
серія ДК № 3516 від 01.07.2009 р.