

Затверджено

482.362.705010201-79 33-5 ЛЗ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЧЕРНІВЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ЮРІЯ ФЕДЬКОВИЧА

Інститут фізико-технічних та комп'ютерних наук

Відділ комп'ютерних технологій

Кафедра комп'ютерних систем та мереж

**СИМУЛЯТОР ТРІЙКОВИХ КВАНТОВИХ МЕРЕЖ**

**482.362. 705010201-79 81 33-5**

**(Пояснювальна записка)**

Сторінок 21

2015

**АНОТАЦІЯ**

Пояснювальна записка складається з основних розділів, які пов'язані з аналізом й обґрунтуванням теми дипломного проекту, призначенням і областю застосування, описом функціональних можливостей програми, вибором технічних і програмних засобів, організації вхідних та вихідних даних, розглядом очікуваних техніко – економічних показників та списком використаних джерел при розробці програмного продукту.

Пояснювальна записка містить: 3 розділи, 28 сторінок, 13 рисунків, 1 таблицю, 18 посилань на літературні джерела.

**ЗМІСТ**

ВСТУП.....	4
1. ПРИЗНАЧЕННЯ І ОБЛАСТЬ ЗАСТОСУВАННЯ.....	4
2. ТЕХНІЧНІ ХАРАКТЕРИСТИКИ .....	4
2.1. Постановка задачі на розробку програми .....	4
2.2. Трійкова система числення.....	8
2.3. Квантові комп'ютери .....	6
2.4. Огляд базових трійкових зворотних логічних елементів.....	7
2.5. Опис алгоритму і функціонування програми .....	8
2.5.1. Quantum Computer Simulator.....	8
2.5.2. JQuantum.....	9
2.5.3. Моделювання примітивів.....	10
2.5.4. Загальний принцип роботи квантового симулятора.....	11
3. ОЧІКУВАНІ ТЕХНІКО – ЕКОНОМІЧНІ ПОКАЗНИКИ .....	13
3.1. Визначення трудомісткості розробки програмного продукту.....	15
3.3. Розрахунок собівартості години роботи на ПК.....	16
3.4. Розрахунок собівартості програмного продукту.....	17
3.5. Розрахунок вартості (ціни) програмного продукту. ....	18
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	<b>ОШИБКА! ЗАКЛАДКА НЕ ОПРЕДЕЛЕНА.</b>

## **ВСТУП**

В пояснювальній записці розглядається розробка на тему «Симулятор трійкових зворотних квантових мереж» з умовним позначенням 482.362.705010201-79 33-5.

Завдання на дипломний проект затверджено на засіданні кафедри КСМ відділу комп'ютерних технологій інституту фізико-технічних та комп'ютерних наук Чернівецького національного університету імені Юрія Федьковича протоколом № 3 від 24 жовтня 2014 р.

## **1. ПРИЗНАЧЕННЯ І ОБЛАСТЬ ЗАСТОСУВАННЯ**

Для досягнення високої продуктивності треба робити обчислення без втрати інформації, тобто зворотньо. Чарльз Беннет (Charles Bennett) показав, що нульова втрата енергії можлива тільки при використанні зворотних обчислювальних блоків. Такий блок дозволяє точно відновити вхідні дані з вихідних даних. Так розроблена програма дозволяє створювати трійкові зворотні мережі. За допомогою бібліотеки примітивів можна створити, наприклад, зворотні суматори, які забезпечують додавання двох чисел без втрати інформації.

## **2. ТЕХНІЧНІ ХАРАКТЕРИСТИКИ**

### **2.1. Постановка задачі на розробку програми**

Розроблена програма повинна надавати можливість конструювати зворотні трійкові мережі з можливістю переглядати результат складеної схеми. Створено бібліотеку примітивів.

Область застосування програми – квантові технології.

## 2.2. Трійкова система числення

Трійкова система числення – позиційна система числення з цілочисельною основою, рівною 3. Існує в двох варіантах: несиметрична і симетрична.

У несиметричній трійковій системі числення частіше застосовуються цифри  $\{0, 1, 2\}$ , а в симетричній трійковій системі числення знаки  $\{-, 0, +\}$  або  $\{-1, 0, +1\}$ .

Звичайний приклад трійкової логіки в житті пов'язаний з постійним струмом: струм рухається в одну сторону, в іншу сторону або він відсутній. У чому ж плюси трійкової СЧ над двійковою? Розглянемо ці плюси:

1. Менше розрядів. Візьмемо число 10 в десятковій СЧ і переведемо його в двійкову СЧ, отримаємо 1010, переведемо в трійкову симетричну СЧ, отримаємо +0+, ну а якщо в трійкову несиметричну СЧ, то отримаємо 101. Ми бачимо, що к-ть розрядів одного і того самого числа в трійковій симетричній і несиметричній СЧ менша, ніж у двійковій СЧ.

2. Ємність. Трійкова СЧ при тій самій к-ті розрядів вміщує більший діапазон чисел:  $3^n > 2^n$ , де  $n$  – натуральне число. Наприклад, якщо  $n = 9$ , то  $3^9 = 19683 > 2^9 = 512$ .

3. Економічність системи числення. Економічність системи числення — запас чисел, який можна записати в даній системі за допомогою певної кількості знаків. Чим більше запас тим економічніша система. За витратами числа знаків (в трьохрозрядному десятковому числі  $3 \cdot 10 = 30$  знаків) найбільш економічна з позиційних показових несиметричних систем числення. Позначимо  $p$  основу системи числення,  $n$  кількість необхідних знаків. Тоді отримаємо  $n/p$  розрядів необхідних для запису цього набору знаків в заданій системі числення, а кількість чисел яке при цьому можна записати дорівнюватиме  $p \cdot n/p$ .

### 2.3. Квантові комп'ютери

Квантовий комп'ютер – обчислювальний пристрій, що працює на основі квантової механіки. Квантовий комп'ютер принципово відрізняється від класичних комп'ютерів, що працюють на основі класичної механіки.

Завдяки величезній швидкості розкладання на прості множники, квантовий комп'ютер дозволить розшифровувати повідомлення, зашифровані за допомогою популярного асиметричного криптографічного алгоритму RSA. До цих пір цей алгоритм вважається порівняно надійним, так як ефективний спосіб розкладання чисел на прості множники для класичного комп'ютера в даний час невідомий. Для того, наприклад, щоб отримати доступ до кредитної карти, потрібно розкласти на два простих множника число завдовжки в сотні цифр. Навіть для найшвидших сучасних комп'ютерів виконання цього завдання зайняло б більше часу, ніж вік Всесвіту, в сотні разів. Завдяки алгоритму Шора це завдання стає цілком здійсненною, якщо квантовий комп'ютер буде побудований.

Канадська компанія D-Wave заявила в лютому 2007 року про створення зразка квантового комп'ютера, що складається з 16 кубіт. Цей пристрій працює на кубітах – квантових аналогах бітів.

Але можна побудувати комп'ютери не на бітах, а на кутрітах – аналогах трітів в квантовому комп'ютері.

Кутріт (квантовий тріт) – квантова комірка, яка має три можливих стани.

Справжнє новаторство методу Ланьон в тому, що, використовуючи в універсальних квантових вентилях кутріти замість кубітів, дослідники можуть істотно знизити кількість необхідних вентилів.

Ланьон стверджує, що комп'ютер, який в звичайному випадку використовував би 50 традиційних квантових вентилів, зможе обійтися всього дев'ятьма, якщо буде заснований на трійковому поданні.

Також, згідно з деякими дослідженнями, використання кутрітів замість кубітів дозволить спростити реалізацію квантових алгоритмів і комп'ютерів.

## 2.4. Огляд базових трійкових зворотних логічних елементів

Існує декілька базисів трійкової зворотної логіки, але в основі більшості з них лежать вже готові примітиви квантова ціна яких більша за одиницю.

Існує 6 примітивів, їх таблиці істинності наведені в табл. 2.1.

Таблиця 2.1.

Таблиця переходів примітивів

A	A(0) = A	A(+1) = A+1	A(+2) = A+2	A(01) = 2A+1	A(02) = 2A+2	A(12) = 2A
0	0	1	2	1	2	0
1	1	2	0	0	1	2
2	2	0	1	2	0	1

Також існує 5 двотрітових примітивів, які представляють собою керовані варіанти одотрітових примітивів за виключенням примітива A(0). Їх графічне позначення наведене на рис. 1.1. Принцип роботи двотрітових примітивів наступний, якщо на вхід A подається 2, то виконується перетворення Y вхідного сигналу B, де  $Y \in \{+1, +2, 01, 02, 12\}$ , відповідно до табл. 1.1. В іншому випадку сигнал B проходить незмінним на вихід D. Вихід C є керуючим і повторює вхідний сигнал A ( $B = A$ ) для забезпечення зворотності вентилі.

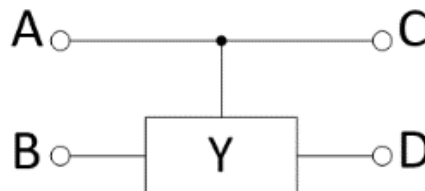


Рис. 1.1 Керовані зворотні примітиви

## **2.5. Опис алгоритму і функціонування програми**

Багато науковців сьогодні займаються проблемою квантового комп'ютингу, але для створення будь яких логічних схем потрібно спочатку створити їх комп'ютерну модель. Тут постає потреба у створення програмного забезпечення яке б з легкістю, зручністю, швидкістю і гнучкістю могло симулювати роботу різних квантових пристроїв, які зовсім скоро повині про себе заявити.

В Світі існує кілька аналогів стимуляторів, які виконують схожі дії і є подібними за своєю структурою, але всі вони працюють з двійковими сигналами(кубітами). Ми не змогли знайти жодного аналогічного симулятора, який міг би працювати з трійковими сигналами (кутрітами). Але для одного тільки створення схеми потрібно прикласти велику кількість зусиль і часу. Простому пересічному інженеру-схемотехніку треба добре розбиратись в квантовій схемотехніці, для розуміння всіх тонкощів.

### **2.5.1.Quantum Computer Simulator**

Перший аналог програмного забезпечення представленого в магістерській роботі. Симулятор створений корпорацією SENCO в 1999 році – є першим офіційним квантовим симулятором. На рис. 2.1. зображено вікно програми Quantum Computer Simulator.

Недоліки Quantum Computer Simulator корпорації SENCO:

1. Основним недоліком є те що він платний вільно розповсюджується тільки в демо версії.
2. Важкий в роботі.
3. Не дозволяє оперувати з вентилями Фредкіна.
4. Для створення нового елемента потрібно вибирати багато опцій.
5. Результати роботи не можна подивитися в покроковому режимі.
6. Не можливо дізнатися які операції відбуваються з квантовими бітами.



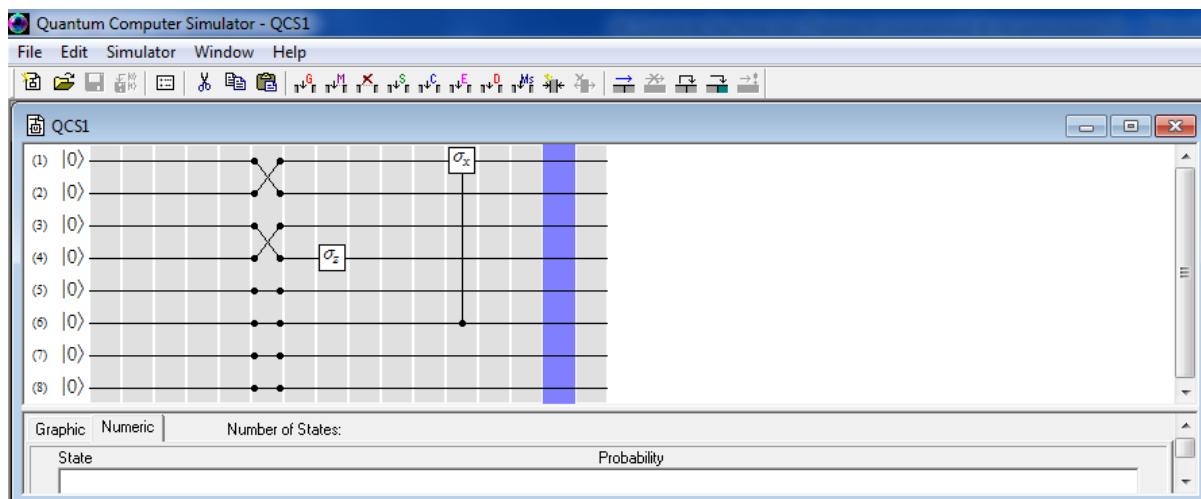


Рис. 2.1 Quantum Computer Simulator корпорації SENCO

Переваги Quantum Computer Simulator корпорації SENCO:

1. Відображення результатів на графіку.
2. Є можливість створювати користувацький вентиль, задаючи його матрицю, та вибираючи контрольовані лінії.

### 2.5.2.JQuantum

JQuantum – симулятор дуже функціональний, розроблений на мові Java що робить його повністю не залежним від операційної системи. Існує набір схем, що робить роботу з симулятором простішу. Симулятор створений групою людей Andreas de Vries, Dave Boden, Axel Thuresson і Вадим Міргород. Програма створена в 2004 році, і підтримувалася до 2010, на даний момент версія симулятора А 2.3.1. На рис. 2.2. зображено вікно програми Quantum Computer Simulator.

Недоліком JQuantum є те, що немає чітких числових даних для відображення.

Переваги JQuantum:

1. Великий набір вентилів.
2. Підтримка програмного продукту протягом тривалого часу.
3. Група розробників, які підтримують і оновлюють симулятор.
4. Покроковий режим відображення даних.

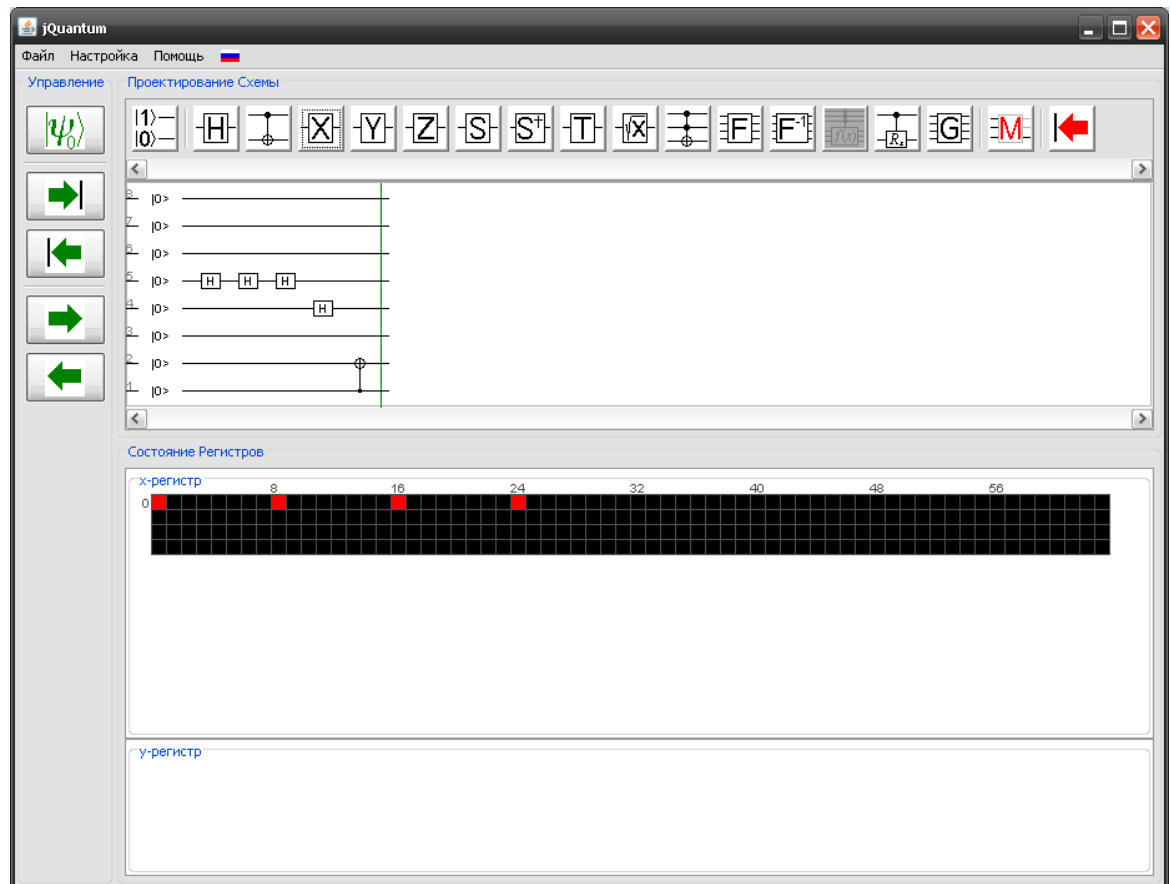


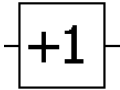
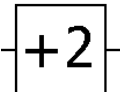
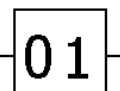
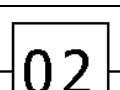

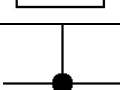
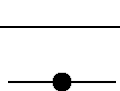
Рис.2.2 Симулятор работы комп'ютера JQuantum

### 2.5.3.Моделювання примітивів

Робота квантових комп'ютерів буде реалізована на елементах які не несуть до втрати інформації, тобто з будь якого кінцевого стану можна отримати початковий. Такі елементи називаються «Зворотніми Елементами». Наприклад, зворотнім логічним елементом є елемент НЕ, внаслідок того, що з будь якого кінцевого стану можна отримати початковий .

Кожен вентиль(елемент) виконує свою специфічну функцію. При розміщенні примітива на лінію відповідний елемент матриці змінює своє значення. Наприклад, при розміщенні примітива 12 на лінію мереж, відповідний елемент матриці поміняє своє значення на «12». В таблиці 2.2 показані примітиви.

## Примітиви

Назва елемента	Графічне представлення
+1 (змінює елем. матриці на 1).	
+2(змінює елем. матриці на 2).	
01(змінює елем. матриці на 01).	
02 (змінює елем. матриці на 02).	
12 (змінює елем. матриці на 12).	
Контролюючий елемент (вверх)	
Контролюючий елемент (вниз)	

Коли на одному стовбці розміщенні примітив і контролюючий елемент, то при натисканні кнопки «Старт» буде побудований контрольований елемент.

#### 2.5.4. Загальний принцип роботи квантового симулятора

В квантових симуляторах, для здійснення обчислень кожен елемент представлений квадратною матрицею, кількість стовпців і рядків пропорційна кількості спінів на яких «стоїть елемент», тобто якщо елемент однокубітовий то його матриця буде розмірністю два на два.

482.362. 705010201-79 81 33-5

$$\begin{matrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \\ \text{Output} \end{matrix} = \begin{matrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \\ \text{Input} \end{matrix}$$

Рис. 2.5 Результат роботи вентиля CNOT

Кожен симулятор і взагалі кожен логічний схем можна уявити в вигляді таблиці де рядочки це розрядність, а стовпчики це елементи розміщені в певному порядку.

Для розміщення елементів на мережі доцільно використовувати таблицку в якій стовпчики відповідають тактам, а рядочки лініям. Тоді кожна комірка має свою адресу і позицію в яку можна занести певний однокубітовий вентиль. Тоді як двокубітові вентиля будуть займати 2 комірки і т д.

Кожен елемент має своє унікальне значення. Створюється матриця з цих значень, тоді легко зробити інтерфейс для розміщення вентилів в мережі. Ця матриця ініціалізується деякими початковими значеннями які відповідають які відповідають відсутністю елементів, або просто лінією(спіном). При розміщенні елементу на певну позицію відповідні значення в матриці змінюються на значення які відповідають елементу(контрольовані елементи), таким чином створюється матриця заповнена елементами.

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Рис. 2.6. Матриця, ініціалізована початковими значеннями

В нашому симуляторі вхідні значення для кожної лінії буде  $[0;1;2]$  або  $[2;1;0]$ , виходячи з цього для  $n$ -бітного квантового симулятора вхідні дані представляються кількістю  $n*3$  значень. Далі ці значення передаються в створену квантову мережу.

При обчисленні вхідні дані поступово проходять кожен такт обчислень, на якому перевіряється яке значення в даній комірці матриці, далі викликається функція що визначає дії при знаходженні певного значення. В функцію передаються поточний номер комірки, і якщо елемент не однотрітовий то значення комірок на яких розміщується елемент або номер контрольованої лінії(спіна). Далі два значення поточного спіна додаються по модулю три з матрицею яка відповідає поточному елементу і дані змінюються відповідно до елемента і вхідних даних.

Коли застосувати цей принцип до кожної з комірок в циклі, по стовпчиках а потім по рядочках матриці, можна досягти таких же кінечних результатів як і при роботі ідеальних квантових мереж.

Якщо рознести всі елементи таким чином щоб в одному такті виконувалась операція тільки з одним вентилям можна досягти зручності і наочності в покрокових результатах, тобто на кожному кроці ми будемо бачити як і які дані будуть мінятися в процесі обчислень.

Таким чином роботу програми можна поділити умовно на 2 частини:

1. Моделювання схеми.
2. Виконання обчислень.

### **3. ОЧІКУВАНІ ТЕХНІКО – ЕКОНОМІЧНІ ПОКАЗНИКИ**

#### **3.1. Визначення трудомісткості розробки програмного продукту**

Основним завданням техніко-економічного обґрунтування дипломного проекту є визначення величини економічного ефекту від використання основних та другорядних результатів, одержаних в результаті розробки. Оцінка ефективності прийнятого рішення повинна бути комплексною і враховувати всі його аспекти [17-18].

### 3.2. Розрахунок трудомісткості розробки програмного продукту

Проведемо оцінку витрат праці, виходячи з того, що розмір вихідного тексту запису алгоритму і даних в основному визначає затрати праці і час розробки  $T$  програмного продукту.

Трудомісткість розробки програмного продукту ( $t$ ) визначається за формулою (3.1), люд.-міс.

$$t = 3,6 \cdot (z_{т.в.к})^{1,2} = 3,6 \cdot (2,43)^{1,2} = 10,44, \quad (3.1)$$

де  $z_{т.в.к}$  – число тисяч команд програмного коду.

Загальна тривалість розробки ПП ( $T$ ) розраховується за формулою (3.2), міс.:

$$T = 2,5 \cdot t^{0,32} = 2,5 \cdot 10,44^{0,32} = 5,29.$$

Середня кількість виконавців ( $PL_{вик}$ ) розраховується виходячи з трудомісткості та тривалості розробки ПП за формулою (3.3), люд.

$$PL_{вик} = t/T = 10,44/5,29 = 1,97. \quad (3.3)$$

Продуктивність праці групи розробників ПП ( $P_p$ ), вихідних команд/люд.-міс. Визначається за формулою (3.4):

$$P_p = 1000 \cdot z_{т.и.к} / t = 1000 \cdot 2,43 / 10,44 = 232,75. \quad (3.4)$$

Як вихідну команду будемо розглядати один оператор програми (не враховуючи коментарі). Загальний об'єм вихідного тексту програми складає приблизно 2430 рядки.

Час, необхідний для розробки програмного продукту, можна визначити за формулою:

$$T_{заг} = T_{ПО} + T_o + T_a + T_{БС} + T_H + T_{нт} + T_o, \text{ (год)}, \quad (3.5)$$

$$T_{заг} = 2 + 53 + 44 + 30 + 66 + 185 + 72 = 452 \text{ (год)}.$$

де  $T_{ПО}$  – час на підготовку опису завдання (год) береться за фактом.

$T_o$  – час на опис завдання (год) визначається за формулою:

$$T_o = \frac{Q \cdot B}{50 \cdot K} = \frac{2430 \cdot 1,2}{50 \cdot 1,1} = 53,01, \quad (3.6)$$

де  $Q$  – кількість рядків у програмі;

482.362. 705010201-79 81 33-5

$B$  – коефіцієнт урахування змін завдання, коефіцієнт  $B$  у залежності від складності завдання і кількості змін обирається в інтервалі від 1,2 до 1,5;

$K$  – коефіцієнт враховує кваліфікацію програміста і обирається з таблиці 3.1.

Таблиця 3.1.

Значення коефіцієнта кваліфікації виконавця  $K$ 

Стаж програміста	Значення коефіцієнта $K$
до 2-х років	0,8
від 2 до 3 років	1,0
від 3 до 5 років	1,1 – 1,2
від 5 до 10 років	1,2-1,3
Понад 10 років	1,3-1,5

$T_a$  – час на розробку алгоритму (год) і  $T_{bc}$  – час на розробку блок - схеми (год) розраховуємо за формулою:

$$T_a = \frac{Q}{50 \cdot K} = \frac{2430}{50 \cdot 1,1} = 44,18, \quad (3.7)$$

$T_n$  – час написання програми мовою програмування (год) визначається за формулою:

$$T_n = \frac{Q \cdot 1,5}{50 \cdot K} = \frac{2430 \cdot 1,5}{50 \cdot 1,1} = 66.27, \quad (3.8)$$

$T_{nt}$  – час налагодження та тестування програми (год) визначається за формулою:

$$T_{nt} = \frac{Q \cdot 4,2}{50 \cdot K} = \frac{2430 \cdot 4,2}{50 \cdot 1,1} = 185.54, \quad (3.9)$$

$T_o$  – час витрачений на оформлення документації (год) обирається за фактом.

### 3.3. Розрахунок собівартості години роботи на ПК

Основою для розрахунку видатків на утримання та експлуатацію ПЕОМ, що відносяться до даного програмного продукту, є собівартість 1-єї машино-години роботи ПЕОМ, тобто витрати, які виконуються за годину роботи на комп'ютері при використанні програми, і визначається за формулою:

$$C_{M.год} = B_{сум} / T_{роб} = \frac{31023,24}{3375} = 9,19 \text{ (грн/год)}, \quad (3.10)$$

де  $B_{сум}$  – сумарні річні витрати (грн);  $T_{роб}$  – час роботи комп'ютера, який визначається як добуток кількості робочих днів в році на час роботи комп'ютера в день (год), помножені на коефіцієнт (0,9), що позначає ремонт і профілактику обладнання.

Сумарні річні витрати на утримання й експлуатацію персонального комп'ютера визначаються за формулою:

$$B_{сум} = B_{ЕН} + B_M + B_{проф} + A + 3П_{обсл}^{осн} + B_{обсл}^{€CB}, \quad (3.11)$$

$$B_{сум} = 1216 + 84 + 126 + 1050 + 21193,2 + 7354,04 = 31023,24 \text{ (грн)},$$

де  $B_{ЕН}$  – витрати на електроенергію;

$B_M$  – витрати на витратні матеріали;

$B_{проф}$  – витрати на профілактику;

$A$  – амортизаційні відрахування;

$3П_{обсл}^{осн}$  – основна заробітна плата обслуговуючого персоналу;

$B_{обсл}^{€CB}$  – єдиний соціальний внесок.

Витрати на електроенергію в рік  $B_{ЕН}$  визначаються як сума витрат електроенергії на роботу ЕОМ ( $B_{ПК}$ ), і витрат на освітлення приміщення ( $B_{ОСВ}$ ) за формулою:

$$B_{ЕН} = B_{ПК} + B_{ОСВ} = 1094 + 122 = 1216 \text{ (грн)}, \quad (3.12)$$

де  $B_{ПК}$  і  $B_{ОСВ}$  визначаються як:

$$B_{ПК} = T_{роб} \cdot \mathcal{U} \cdot P = 3375 \cdot 0,36 \cdot 0,9 = 1094 \text{ (грн)},$$

$$B_{ОСВ} = T_{роб} \cdot \mathcal{U} \cdot P = 3375 \cdot 0,36 \cdot 0,1 = 122, \quad (3.13)$$

де  $\mathcal{U}$  – вартість 1 кВт електроенергії (0,36 грн),  $P$  – потужність ПК або освітлювальних приладів (кВт/год).



482.362. 705010201-79 81 33-5

Основна заробітна плата обслуговуючого персоналу ( $ЗП_{обсл}^{осн}$ ) визначається за формулою:

$$ЗП_{обсл}^{осн} = З^I \cdot K_T \cdot 12 = 1218 \cdot 1,45 \cdot 12 = 21193,2 \text{ (грн)}, \quad (3.14)$$

де  $З^I$  – мінімальна зарплата 1-го розряду (1218 грн);

$K_T$  – тарифний коефіцієнт, що відповідає розряду тарифної сітки по якому працює оператор. Наприклад, обираємо 6 розряд, якому відповідає тарифний коефіцієнт 1,45.

Єдиний соціальний внесок ( $B_{обсл}^{ЄСВ}$ ) складає 34,7% від основної заробітної плати, тобто  $B_{обсл}^{ЄСВ} = 21193,2 \cdot 0,347 = 7354,04$  грн

Витрати на витратні матеріали ( $B_m$ ), такі як папір, CD/DVD-диски, картриджі, тощо, як правило беруться за фактом і становлять 2 % від балансової вартості обчислювальної техніки ( $B_б = 4200$  грн,  $B_m = 84$  грн).

Витрати на профілактику ( $B_{проф}$ ) становлять 3% від балансової вартості ПЕОМ з периферією ( $B_б = 4200$  грн,  $B_{проф} = 126$  грн).

Амортизаційні відрахування в рік ( $A$ ) визначаються як відношення балансової вартості ПЕОМ до кількості років її експлуатації ( $N_p$ ):

$$A = B_б / N_p = 4200 / 4 = 1050 \text{ (грн)}. \quad (3.15)$$

Визначивши загальну суму витрат  $B_{сум}$  за формулою (3.11), одержимо собівартість 1-єї машино-години роботи ЕОМ за формулою (3.10).

Знаючи собівартість 1 години роботи на ПЕОМ і час створення ПП ( $T_{заг}$  за формулою (3.5)), можна визначити витрати на утримання й експлуатацію ПЕОМ при розробці ПП:

$$B_{ПП} = C_{М.год} \cdot T_{заг} = 9,19 \cdot 452 = 4153,88 \text{ (грн)}. \quad (3.16)$$

### 3.4. Розрахунок собівартості програмного продукту

Собівартість програмного продукту визначається загальними витратами на виготовлення програмного продукту і обчислюється як сума таких показників за формулою (3.17):

1. Основна заробітна плата виконавця робіт зі створення програмного продукту  $ЗП_{вих}^{осн}$ , (грн).

482.362. 705010201-79 81 33-5

2. Додаткова заробітна плата виконавця робіт зі створення програмного продукту  $ЗП_{вих}^{доп}$ , (грн).
3. Нарахування на заробітну плату (єдиний соціальний внесок)  $B_{вих}^{ЕСП}$ , (грн).
4. Витрати на утримання і експлуатацію ПЕОМ, що відносяться до програмного продукту  $B_{ПП}$ , (грн).

$$C_{ПП} = ЗП_{вих}^{осн} + ЗП_{вих}^{доп} + B_{ЕСП} + B_{ПП}, \text{ (грн)}, \quad (3.17)$$

$$C_{ПП} = 6519,59 + 655 + 2489 + 2827,11 = 12490,7, \text{ (грн)},$$

де основна заробітна плата виконавців програмного продукту визначається як

$$ЗП_{вих}^{осн} = (З^I * K_T * T_{заг}) / (Ч_P * T_{р\partial}) \left(1 + \frac{П}{100}\right) = \frac{1218 \cdot 1,73 \cdot 452}{21 \cdot 8} \left(1 + \frac{15}{100}\right) = 6519,59, \quad (3.18)$$

де  $З^I$  – мінімальна зарплата 1-го розряду (1218 грн);

$K_T$  - тарифний коефіцієнт, що відповідає розряду тарифної сітки по якому працює виконавець. Розряд інженера-програміста – 9, якому відповідає тарифний коефіцієнт 1,73;

$T_{заг}$  – загальний час створення програмного продукту (год);

$Ч_P$  – число робочих днів на місяць (в середньому – 21 день);

$T_{р\partial}$  – тривалість робочого дня в годинах (8 год.).

$П$  – відсоток премії (15 %).

Додаткова заробітна платня (грн) складає 10-15 % від основної зарплати виконавця,  $ЗП_{вих}^{доп} = 655$  (грн).

Нарахування на заробітну плату (єдиний соціальний внесок)  $B_{вих}^{ЕСП}$ , складає 34,7% від основної і додаткової зарплати виконавця,  $B_{вих}^{ЕСП} = 2489$  грн

### 3.5. Розрахунок вартості (ціни) програмного продукту.

Вартість (ціна) програмного продукту, запропонована розробником, визначається за формулою:

$$Ц_{ПП} = C_{ПП} \left(1 + \frac{P}{100}\right) = 12490,7 \cdot \left(1 + \frac{40}{100}\right) = 17486,98 \text{ (грн)}, \quad (3.19)$$

де  $C_{ПП}$  – загальні витрати на створення програмного продукту (грн),  $P$  – рентабельність розробки (40 %).

482.362. 705010201-79 81 33-5

Річний економічний ефект визначається за формулою:

$$E_p = Z_0 \cdot T - Z_{\text{прив}} = 1537,2 \cdot 200 - 7820,75 = 299619,25 \text{ (грн)}, \quad (3.20)$$

де  $Z_0$  – витрати на розв'язання задачі традиційними методами;

$T$  – періодичність розв'язання задачі, для нашого випадку  $T = 200$ ;

$Z_{\text{прив}}$  – приведені витрати.

Для визначення параметру  $Z_0$  використовується формула:

$$Z_0 = T_0 \cdot 3\Pi_{\text{вик}}^{\text{год}} = 36 \cdot 42,7 = 1537,2 \text{ (грн)}, \quad (3.21)$$

де  $T_0$  – трудомісткість на складання документу, (люд.-год.);

$3\Pi_{\text{вик}}^{\text{год}}$  – заробітна плата виконавця за одну годину.

$$3\Pi_{\text{вик}}^{\text{год}} = (3\Pi_{\text{вик}}^{\text{осн}} + 3\Pi_{\text{вик}}^{\text{доп}}) / \text{Ч}_p \cdot T_{pb} = (6519,59 + 655) / 21 \cdot 8 = 42,7.$$

Для визначення параметру  $Z_{\text{прив}}$  використовується формула:

$$Z_{\text{прив}} = Q \cdot C_{\text{М.год}} \cdot \Pi_p + E_k \cdot \text{Ц}_{\text{ПП}}, \quad (\text{грн}), \quad (3.22)$$

$$Z_{\text{прив}} = 2,43 \cdot 9,19 \cdot 232,75 + 0,15 \cdot 17486,98 = 7820,75,$$

де  $Q$  – загальний об'єм вихідного тексту програмного додатка (тис. команд);

$C_{\text{М.год}}$  – собівартість години роботи на комп'ютері (грн);

$\Pi_p$  – продуктивність праці розробників програмного забезпечення (команд/люд.-год.);

$E_k$  – нормативно-галузевий коефіцієнт, який враховує використання програмного продукту протягом декількох років ( $E_k = 0,15$ );

$\text{Ц}_{\text{ПП}}$  — ціна програмного продукту.

Тоді річний економічний ефект за формулою (3.20) складає 299619,25 (грн).

**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ЛІТЕРАТУРИ**

1. Трійкова система числення. [Електронний ресурс]. — Режим доступу : [http://uk.wikipedia.org/wiki/Трійкова\\_система\\_числення](http://uk.wikipedia.org/wiki/Трійкова_система_числення).
2. А. Китаев, А. Шень, М. Вялый. Классические и квантовые вычисления, М.: МЦНМО, 1999. 192 с.
3. Квантовый логический элемент Тоффли [Електронний ресурс] Режим доступу  
[http://neerc.ifmo.ru/mediawiki/index.php/Квантовый\\_логический\\_элемент\\_Тоффли](http://neerc.ifmo.ru/mediawiki/index.php/Квантовый_логический_элемент_Тоффли)
4. Крохмальський Т. Квантові комп'ютери: основи й алгоритми(короткий огляд)// Журнал фізичних досліджень. - Інститут фізики конденсованих систем Національної академії наук України. – 2003. – №1. – с 1-15
5. Квантовый комп'ютер. Режим доступу :  
[http://ru.wikipedia.org/wiki/Квантовый\\_компьютер](http://ru.wikipedia.org/wiki/Квантовый_компьютер).
6. Берман Г. П., Дулен Г. Д., Майньери Р., Цифринович В. И. Введение в квантовые компьютеры. – М.-Ижевск: Институт компьютерных исследований, 2004. – 188 с.
7. McMahon D. Quantum computing explained./ D.McMahon. – Wiley-Interscience, 2008. – 332p.
8. А. Китаев, А. Шень, М. Вялый. Классические и квантовые вычисления, М.: МЦНМО, 1999. 192 с.
9. Валиев К.А. Квантовые компьютеры и квантовые вычисления // Успехи физических наук. – 2005. – Т.175,.N1. – С.35-39.
- 10.Desurvire E. Classical and quantum information theory./ E. Desurvire – Thales, France, 2009. – 714p.
11. Quantum Computer Simulator – User Guide./ SENKO Corporation – Japan, 1999. – 102p.

12. Md. Saiful Islam, M. M. Rahman, Zerina Begum Synthesis of Fault Tolerant Reversible Logic Circuits// Institute of Information Technology University of Dhaka, - Dhaka – 100 , Bangladesh, p 1-4.
13. H.R. Bhagyalakshmi, M.K. Venkatesha, Multiplier using reversible logic gates/ International Journal of Engineering Science and Technology, 2010. – 8p.
14. R. Landauer, Irreversibility and Heat Generation in the Computational Process/ IBM Journal of Research and Development, pp. 183-191, 1961.
15. C.H. Bennett, Logical Reversibility of Computation/ IBM J. Research and Development, pp. 525-532, November 1973.
16. T. Toffoli., Reversible Computing/ MIT Lab for Computer Science (1980).
17. E. Fredkin and T. Toffoli, Conservative logic/ Int’l J. Theoretical Physics, Vol. 21, pp. 219–253, 1982.
18. Azad Khan, Design of full adder with reversible gate/ International Conference on Computer and Information Technology, Dhaka, Bangladesh, pp: 515-519, 2002.
19. R. Feynman, Quantum Mechanical Computers, Optics News, Vol. 11, pp. 11–20, 1985. B.
20. Я.І. Щедрій, Ю.Л., Дещинський, О.С. Мурін та ін. Основи охорони праці: Навчальний посібник/ За ред. Я.І. Бедрія. – 3-тє вид., переробл. і доп. – Львів: “Магнолія плюс”, видавець СПД ФО В.М. Піча, 2004.- 240с.
21. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин ДСанПІН 33 2 007 98.