Вінницький національний технічний університет
(повне найменування вищого навчального закладу)

Факультет інформаційних технологій та комп'ютерної інженерії
(повне найменування інституту)

Кафедра обчислювальної техніки
(повна назва кафедри)

Пояснювальна записка

до магістерської кваліфікаційної роботи
магістра
(освітньо-кваліфікаційний рівень)

на тему: «Емуляція квантових логічних функцій і розрахунків за допомогою мов програмування високого рівня»

Виконав: студент _2_ курсу, групи 1КІ-18м
напряму підготовки (спеціальності)
123 – «Комп'ютерна інженерія»
(шифр і назва напряму підготовки, спеціальності)

Берко С.Г.
(прізвище та ініціали)

Керівник к.т.н., доц. Гарнага В.А.
(прізвище та ініціали)

Рецензент к.т.н., проф. Яремчук Ю.Є.
(прізвище та ініціали)

м. Вінниця – 2019 року

АНОТАЦІЯ

Дана магістерська кваліфікаційна робота присвячена розробці засобів емуляції квантових логічних функцій і розрахунків за допомогою мов програмування високого рівня.

В магістерській кваліфікаційній роботі зроблено аналіз методів емуляції квантових логічних функцій та виконано огляд способів побудови систем з використанням цих методів, вдосконалено метод поширення внутрішнього стану для віртуальних частинок та розроблено алгоритм емуляції квантових логічних функцій, а також розроблена програма з прикладами базових квантових розрахунків.

Також у магістерській роботі розглянуті питання економічної доцільності розробки.

ABSTRACT

This master's qualification is dedicated to the development of emulation tools for quantum logic functions and calculations using high-level programming languages.

In the master's qualification work the analysis of methods of emulation of quantum logic functions is made and the methods of construction of systems using these methods are reviewed, the method of propagation of the internal state for virtual particles is improved, the algorithm of emulation of quantum logic functions is developed, and the program with examples of basic quantum calculations is developed.

Also, in the master's work the questions of economic expediency of development are considered.

ЗМІСТ

ВСТУП

Актуальність теми дослідження.

Квантові комп'ютери в найближчому майбутньому стануть ключовою системою розробки для багатьох галузей науки та виробництва. Надійність криптосистем, безпечна передача даних по мережі, вирішення задач побудови маршрутів і логістики, хімічна промисловість і вирішення багатьох питань в області квантової фізики – все це галузі які отримають значний поштовх з розвитком квантових обчислень. Основною проблемою поки є недоступність таких систем рядовим користувачам і незначна зацікавленість спеціалістів в галузі. Вирішенням даної проблеми може стати емуляція кантових логічних функцій та розрахунків на класичних мовах програмування високого рівня[1]. Перевагою даного підходу є зручність налаштування робочого оточення звичного рядовому користувачу і наявність зрозумілої реалізації базових логічних функцій та розрахунків.

Підсумовуючи розглянуте, завдання подальшої розробки нових та удосконалення існуючих методів емуляції квантових логічних функцій та розрахунків на мовах високого рівня, що є кращими за критеріями достовірність та швидкість роботи, на теперішній час є досить актуальними.

Об'єкт дослідження – процес емуляції квантових логічних функцій.

Предмет дослідження – методи поширення внутрішнього стану віртуальних кубітів та реалізація базових квантових розрахунків.

Метоюдослідження є вдосконалення методів емуляції квантових функцій та розрахунків за допомогою мов програмування високого рівня.

Задачі дослідження: 1) здійснити аналіз існуючих методів емуляції квантових логічних функцій; 2) запропонувати поліпшений метод емуляції квантових логічних функцій для віртуальних ; 3) запропонувати алгоритм та розробити програму для поширення внутрішнього стану для віртуального кубіта; 4) здійснити обґрунтування доцільності виконання нового наукового рішення, розрахувати економічні витрати для створення програмних засобів

емуляції квантових логічних функцій і розрахунків з допомогою мов програмування високого рівня та визначити переваги від впровадження нового програмного продукту.

Методи дослідження: використовувались методи диференційного числення, методи дискретної математики, методи математичної статистики, методи теорії множин. Використано засади об'єктно-орієнтованого програмування для реалізації запропонованого методу по емуляції квантових логічних функцій.

Наукова новизна одержаних результатівполягає у тому, що:

– отримав подальший розвиток метод поширення внутрішнього стану для віртуальних кубітів, що дозволяє більш ефективно проводити емуляцію квантових логічних функцій і розрахунків.

Практичне значення одержаних результатів*:*

– створено алгоритм поширення внутрішнього стану для віртуальних кубітів;

– розроблено програму з прикладами реалізації квантових розрахунків з використанням віртуальних кубітів та логічних функцій.

Апробація результатів магістерської роботи: зроблено доповідь на молодіжній науково-практичній інтернет-конференції студентів, аспірантів та молодих науковців *«Молодь в науці: дослідження, проблеми, перспективи (МН-2019)».*

Публікації : Берко С.Г.Оцінкакоректувальноїздатності ітеративнихзавадостійкихкодів. IV Міжнароднанауково-практична конференція"Потенціалсучасної науки" м. Київ, 10-11грудня 2019 р. [1]

1 АНАЛІЗ МЕТОДІВ ЕМУЛЯЦІЇ КВАНТОВИХ ЛОГІЧНИХ ФУНКЦІЙ І РОЗРАХУНКІВ

Квантові обчислення часто згадуються в новинах: Китай телепортував кубіт з Землі на супутник; алгоритм Шора поставив під загрозу нині використовувані методи шифрування; квантове розподіл ключів знову зробить шифрування надійним засобом захисту; алгоритм Гровера збільшить швидкість пошуку даних. Але що все це означає насправді? Як все це працює? Чи можна освоїти цю тему без знання математики? Ні, якщо ви хочете по-справжньому зрозуміти суть того, що відбувається. Основні ідеї беруть початок в квантовій механіці і часто суперечать здоровому глузду. Спроби описати їх звичайними словами приречені на провал, тому що ці явища не мають відображення в повсякденному житті. Гірше того, словесні описи часто створюють враження, що ми щось зрозуміли, хоча насправді все не так погано - нам не доведеться сильно заглиблюватися в математику, досить того, що намагалися вбити в наші голови в старших класах школи. Квантові обчислення - це дивовижний сплав квантової фізики та інформатики, який об'єднує найяскравіші ідеї з фізики двадцятого століття і дозволяє по-новому поглянути на комп'ютерні технології[2].

1.1Квантові логічні функції як парадигма цифрових розрахунків

Квантові комп'ютери - надзвичайно захоплююча технологія, що подає надії на створення потужних обчислювальних можливостей для рішення раніше нерозв'язних проблем.Експерти стверджують, що IBM лідирувала в області квантових обчислень, тому Google, Intel, Microsoft і безліч стартапів знаходяться під її впливом.Інвесторів приваблюють стартапи в області квантових обчислень, в їх числі IonQ, ColdQuanta, D-WaveSystems і Rigetti, які зможуть змінити цей ринок.Однак, є заковика: сучасні квантові комп'ютери, як правило, не так потужні і не так надійні, як існуючі сьогодні суперкомп'ютери,

а також їм потрібні особливі умови для запуску і завантаження.У січні IBM викликала фурор, коли оголосила про випуск IBM Q SystemOne, першої в світі моделі квантового комп'ютера доступною для бізнесу. Пристрій, було поміщено в гладкий скляний корпус обсягом 9 кубічних футів.Це важлива віха для квантових комп'ютерів, які до сих пір розташовуються в дослідницьких лабораторіях. На думку IBM, покупці вже мають намір взяти в свої руки цю технологію, що подає надії в різних областях: хімії, матеріалознавстві, виробництві продуктів харчування, авіакосмічній промисловості, розробці ліків, прогнозуванні фондового ринку і навіть в боротьбі з кліматичними змінами.[3]Причина заворушень полягає в тому, що квантовий комп'ютер володіє, здавалося б, магічними властивостями, які дозволяють йому обробляти експоненціально більше інформації. Квантовий комп'ютер не просто дуже швидкий комп'ютер, скоріше, це зовсім інша парадигма обчислень, яка вимагає радикального переосмислення.Переможцем в гонці технологій стане та компанія, яка скористається можливостями, наданими цією технологією. IBM, Microsoft, Google і інші технічні гіганти, а також стартапи роблять ставку на цю технологію.БусінесInsider поставив кілька запитань віце-президенту IBM Q StrategyandEcosystem Бобу Сютору про те як зробити ці системи доступними для людей: як люди отримають до них доступ? Яким чином безліч людей зможе навчитися використовувати квантові комп'ютери для виконання своїх завдань? Мало шансів побачити квантові комп'ютери в офісі найближчим часом. Експерти, з якими ми поговорили, вважають, що, незважаючи на те, що вони доступні IBM, пройде ще п'ять-десять років, перш ніж квантові обчислення дійсно стануть мейнстримом. IBM Q SystemOne в даний час доступна тільки в якості служби хмарних обчислень для обраних покупців. Пройде ще якийсь час, перш ніж щось подібне люди зможуть придбати і змусити працювати в особистих цілях.Дійсно, експерти запевняють, що квантові комп'ютери подають великі надії, але вони далекі від масового виробництва. Вони надзвичайно тендітні і вимагають спеціальних умов для роботи. Більш того, квантові комп'ютери сьогодні не так надійні і не так потужні, як комп'ютери, які у нас

вже є.«Ми вважаємо, що приблизно через десять років квантовий комп'ютер змінить ваше життя або мою, - заявив BusinessInsiderДжимКларк, директор по квантовому обладнання Intel. - Насправді ми зараз тільки на першій милі марафону. Це не означає, що ми не стурбовані цим ».Одного разу Білл Гейтс сказав, що математика, що лежить в основі кванта, була за межами його розуміння, але не всі з цим погодилися.«Кілька помилково уявлення про те, що квантова фізика - це теж фізика і вона занадто складна, - переконує BusinessInsider Кріс Монро, генеральний директор і співзасновник IonQ. - Незбагненною для багатьох людей її робить те, що вона - незрозуміла, але вона також незрозуміла для мене, як і для вас. Якщо щось може перебувати в суперпозиції, це означає, що воно може бути в двох станах одночасно. Це дивно, тому що ми не стикаємося з подібним в реальному світі ".Комп'ютери, які ми використовували відображають дані у вигляді рядка 1 або 0, названої двійковим кодом. Проте, квантовий комп'ютер може представляти дані у вигляді 1, 0 або, що особливо важливо, обидва числа одночасно.Коли система може перебувати в більш ніж одному стані одночасно, це називається «суперпозиція» - одне із, здавалося б, магічних властивостей квантових обчислень. Іншим ключовим принципом тут є «заплутаність», яка є квантовим властивістю, що дозволяє двом частинкам рухатися абсолютно синхронно, незалежно від того, наскільки далеко вони фізично розділені. Ці дві якості об'єднуються в комп'ютер, який може обробляти одночасно набагато більше даних, ніж будь-яка система на сьогоднішньому ринку. Потужність квантового комп'ютера вимірюється в кубітах, базової одиниці виміру в квантовому комп'ютері. Точно так же, як сучасні комп'ютери мають 32-х або 64-розрядні процесори (міру того, скільки даних вони можуть обробляти одночасно) квантовий комп'ютер з великою кількістю кубітів має значно більший обсяг обчислювальної потужності. Все це означає, що квантовий комп'ютер може вирішувати проблеми, які раніше були обмежені обчислювальною потужністю. Наприклад, квантовий комп'ютер може грубим методом дозволити відому проблему комівояжера - складну обчислювальну

задачу, яка вимагає знаходження найкоротшого маршруту між кількома містами, перш ніж повернутися додому. Це звучить просто, але якщо дивитися на це з точки зору математики, знайти єдиний оптимальний шлях стає складніше, коли ви додаєте більше міст в його маршрут. Точно так же квантовий комп'ютер міг би пробратися через найхитріші, самі трудомісткі завдання, відсіваючи величезні обсяги фінансових, фармацевтичних або кліматологічних даних, щоб знайти оптимальні рішення. Дійсно, квантовий стартап D-Wave вже співпрацює з Volkswagen, щоб проаналізувати моделі руху і відсіяти величезна кількість перешкод для того, щоб дістатися до суті. Обговорюється його корисність в сфері криптографії. Квантовий комп'ютер здатний подужати метод шифрування, відмінний від раніше відомого шифру, що дозволяє йому легко розшифрувати навіть державну таємницю. Виявляється великий інтерес з боку світових урядів до цієї корисної функції, в той час як активісти побоюються, що поява квантових обчислень може знищити конфіденційність. "Оскільки квантові обчислення все ще знаходяться на ранніх стадіях свого розвитку, є багато інформації, яка до сих пір залишається недоведеною", - вважає Метью Бріссо, віце-президент Gartner по НДДКР.[3][4] "Але покупці вже шукають область застосування, щоб визначити конкурентні переваги квантових обчислень для їх бізнесу", - говорить він.Незважаючи на весь галас, експерти вважають, що квантові комп'ютери так само далекі від лідируючих позицій, як ПК в 1950-х роках. Звичайно, вони набирають обертів, але повільно.«Квантові обчислення можна порівняти з повільно рухомим поїздом, - заявив BusinessInsider Брайан Хопкінс, віце-президент і головний аналітик Forrester. - Якщо він проходить один дюйм в секунду, то через місяць він пройде вже два дюйма в секунду. Досить скоро він почне рухатися швидше ». Велика проблема зараз полягає в тому, що квантовий комп'ютер не може нічого зробити того, що не зміг бути зробити класичний комп'ютер. Промисловість з нетерпінням чекає моменту, названого квантовим перевагою, коли квантові комп'ютери вийдуть за межі поточних обмежень. «Коли клієнти приходять до нас, головне, що вони нам кажуть - це те, що їм все одно якась

модель, аби вона була корисна для їхнього бізнесу, - говорить аналітик Бріссо. - Не існує будь-якої моделі, яка могла б випередити класичні алгоритми. Нам дійсно потрібно почекати, поки апаратура квантового комп'ютера почне вдосконалюватися ».

Великою проблемою залишається брак обчислювальної потужності. Передбачається, що для квантового переваги потрібно комп'ютер з потужністю в 50 кубіт. Хоча даний рубіж був подоланий в лабораторії, він непостійний і його неможливо підтримувати. Дійсно, кубіти можуть як піддаватися помилок, так і бути непостійними, що призводить до проблем з їх генерацією і знижує їх потенціал.Інший важливий фактор - більш матеріальний. Квантові комп'ютери повинні бути повністю ізольовані від навколишнього середовища, щоб функціонувати, і їм потрібні дуже низькі температури. Навіть найслабші вібрації можуть привести до руйнування кубітів, виводячи їх з суперпозиції, подібно до того, як дитина, стукаючи по столу, змушує обертаються монети падати на стіл.Попередні квантові комп'ютери, такі як IBM Q SystemOne, настільки громіздкі, що необхідні умови ізоляції і охолодження стають реальною проблемою. Ускладнюють цю проблему брак необхідних компонентів: надпровідних кабелів і низькотемпературних рефрижераторів. Вони в сильному дефіциті.В кінцевому рахунку це означає, що, хоча знання удосконалюються і технологія розвивається, квантові обчислення все ще практично нездійсненні.«Однією з проблем в моїй робочій групі є маніпулювання матеріалами, кремнієм, металами, щоб ми могли створити дуже однорідне середовище, - сказав Кларк з Intel. - Це в основному найкраща напівпровідникова техніка. Технологій, які нам потрібні для створення квантових обчислень у великих масштабах, поки не існує ».Інша проблема полягає в тому, що квантові комп'ютери володіють незаперечною потенціалом для забезпечення непередбачених обчислювальних потужностей. Однак, в цьому світі не так багато людей, які насправді мають досвід програмування або управління цими системами, а зачаровані потенційні покупці намагаються

з'ясувати, як насправді цим користуватися.Аналітики стверджують, що IBM в даний час лідирує в гонці квантових обчислень завдяки обмеженою комерційної доступності IBM Q SystemOne. Оскільки доступ до нього здійснюється через хмару, IBM може підтримувати ці особливі умови, щоб цей квантовий комп'ютер функціонував, в той же час дозволяючи обраним клієнтам ним користуватися. «Я думаю, що квантовий комп'ютер IBM розгойдується, - сказав аналітик Бріссо. - Я думаю, що модель квантових обчислень в якості сервісу - вірна модель. Помістивши її в контейнер і звертаючись з конкретними завданнями, вони дійсно намагаються поліпшити його якість ». При цьому аналітики відзначають, що у будь-якого з гравців цього ринку може статися прорив в будь-який момент, який дозволить йому вирватися вперед, і що це як і раніше необхідне суперництво. Різні IT - гіганти по-різному підходять до це проблеми. Intel, IBM, Google і стартап квантових обчислень Rigetti будують системи, створені на основі надпровідних схем, спираючись на сучасні суперкомп'ютери.Microsoft використовує зовсім інший і, можливо, більш ризикований підхід, намагаючись створити кращий кубіт. Топологічний кубіт, який намагається створити Microsoft, фрагментірует електрони для зберігання інформації в декількох місцях одночасно, роблячи її більш стабільною і менш схильною до руйнування. За словами аналітика Хопкінса, це менш надійно, ніж те, що намагаються створити його конкуренти, але результат стане важливим кроком вперед для всієї області квантових обчислень.«Вони вплуталися в авантюру і багато хто вважає, що їм це ніколи не вдасться», - заявляє Хопкінс.Що стосується авантюр, такі стартапи, як IonQ і D-Wave, роблять ставку на передові технології, такі як іонну уловлювання і квантовий отжиг. Простіше кажучи, намагаються різними способами домогтися більшої продуктивності і стабільності від кожного кубіта, використовуючи абсолютно новий методи. «Це дозволяє нам створювати квантовий комп'ютер, який вирішує складні завдання і безперервно прогресує в цьому», - заявив BusinessInsider Марк Джонсон, віце-президент з проектування та розробки процесорів і квантових продуктів в D-Wave.

Навіть за межами квантового переваги експерти запевняють нас, що для традиційних комп'ютерів і суперкомп'ютерів все ж знайдеться місце. До тих пір є ще варто вирішити проблеми з вартістю, розміром, надійністю і обчислювальною потужністю, перш ніж ми зможемо це обговорити. «Слід перевести подих, - сказав аналітик Бріссо. - У цій області відбувається багато захоплюючих речей, що віднімають час. Це конгломерат фізики, інформатики та, відверто кажучи, наукового аналізу. Нам не довелося б вивчати це, якби ми знали всі відповіді, але в майбутньому нас чекає великий обсяг дослідницької роботи ".Проте, для багатьох ясно, що за цим майбутнє. Точно так же, як виробники першого мейнфрейм-комп'ютера не усвідомлювали, що це в кінцевому підсумку призведе до збільшення числа кишенькових смартфонів розміром з долоню. Квантовий комп'ютер може стати першим кроком на абсолютно новому шляху. Мало хто, подібні віце-президенту Microsoft з корпоративного управління Тодду Холмдалу, досить оптимістичні, щоб заявити, що це може бути більш значущим, ніж штучний інтелект і машинне навчання сьогодні. Раніше він говорив своїм дітям, що вони повинні займатися тим, чим захоплені, і що вони завжди можуть отримати роботу в галузі штучного інтелекту. Тепер він скаже те ж саме про квантових обчисленнях. «Це область, яка буде розвиватися. Нам потрібні люди, щоб заповнити її і не дати зачахнути, - сказав Холмдал. - Вона грає важливу роль для нашого покоління, що дає можливість створювати дивовижні речі в майбутньому ».[5][6]

1.2 Засоби емуляції квантових розрахунків

Перш ніж ми розпочнемо, ярекомендую підійти до квантового програмування з чистого листа. Не шукайте, як оголосити та встановити змінні, перекинути код, створити функції тощо. Будь-які попередні уявлення про програмування, ймовірно, не будуть корисні. Квантове програмування - це не просто спосіб зробити наші існуючі програми швидшими - квантове програмування принципово відрізняється від сучасного

програмування.Почнемо з того, що таке кубіт.Кубіт - вектор двох складних чисел з одиничною довжиною.
Давайте ознайомимося з тим, чому кубіти є таким і що це насправді означає. Кубіти сильно відрізняються від бітів. Для початків, біт або 0, або 1. Тут немає ймовірностей, це або відомо, що це 0, або відомо, що це 1. Кубіт, навпаки, по суті є ймовірнісним, тобто два однакових кубіта можуть мати різні значення раз виміряли. Це означає, що квантові обчислення по суті є ймовірнісними. Тепер друга ключова різниця. За допомогою бітів ми можемо читати біт стільки разів, скільки хочемо, не впливаючи на стан біта. Але з кубітами, щойно виміряні, він декогерується (втрачає свої квантові властивості) і руйнується до одного з двох вимірюваних станів (звідси "біт" у "кубіті"). Отже, ми не можемо «відміряти» кубіт; Після вимірювання квантова природа знищується і її неможливо відновити. Ми кількісно оцінюємо ймовірнісний характер вимірювання кубіта, використовуючи два числа: $|\alpha|^2$, ймовірність того, що кубіт буде виміряно як 0, і $|\beta|^2$, ймовірність того, що кубіт буде вимірюватися як 1.[3] Хоча $|\alpha|^2$ і $|\beta|^2$ відображають ймовірність того, як буде вимірюватися кубіт, ми вважаємо внутрішній стан кубіта як дві „амплітуди ймовірності“, α і β. Це складні числа, які визначають суперпозицію між 0 і 1 (суперпозиція є лінійною комбінацією) і їх неможливо виміряти.[6]

Іншими словами, ми вважаємо кубіт як вектор двох складних чисел з одиничною довжиною (довжина вектора дорівнює 1). Ми можемо коротко висловити це математикою, як показано на наступному малюнку (вектор, що містить альфа і бета - кубіт; смужка вище альфа і бета позначає складний кон'югат):

$$\begin{pmatrix}\alpha\\ \beta\end{pmatrix} \text{ such that } |\alpha|^2 + |\beta|^2 = \alpha \cdot \overline{\alpha} + \beta \cdot \overline{\beta} = 1$$

Рисунок1.1–Кубіт як вектор двох складних чисел

Дляповторного підрахунку, кубіти - це вектор двох складних чисел α і β, де вектор має одиничну довжину. Ймовірність того, що кубіт буде вимірюватися як 0, дорівнює величині в квадраті α, $|\alpha|^2$. Ймовірність того, що кубіт буде вимірюватися як 1, дорівнює величині в квадраті β, $|\beta|^2$. Стан кубіта α і β неможливо виміряти. Виміряти можна лише значення, на яке кубіт згортається.Ми часто позначаємо кубіти, використовуючи позначення Dirac, також відомі як позначення Bra-ket. Ця нотація - просто зручний спосіб написання векторів. Braявляє собою вектори рядків і позначається ⟨|; ket являє собою вектори стовпців і позначається|⟩. Наприклад, ми можемо записати стани «0» і «1» кубіта в позначення Bra-ket наступним чином (будьте обережні, щоб не плутати те, що знаходиться в Bra/ ket, з тим, що знаходиться всередині вектора):

$$\langle 0| := \begin{pmatrix} 1 & 0 \end{pmatrix} \qquad |0\rangle := \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$\langle 1| := \begin{pmatrix} 0 & 1 \end{pmatrix} \qquad |1\rangle := \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Рисунок 1.2–Bra-ket позначення кубітів

Кубіти можуть бути чистими, або змішаними. Якщо стан кубіта можна повністю описати, використовуючи лінійну комбінацію |0⟩ і |1⟩, то ми говоримо його в чистому стані. Ми часто позначаємо кубіти чистого стану за допомогою наступних позначень:

$$|\Psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$$

Рисунок 1.3–Кубіти чистого стану

Ось кілька прикладів чистого стану кубітів та загальної стенограми для їх позначення.

$$|+\rangle := \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$$
$$|-\rangle := \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$$
$$|i\rangle := \frac{1}{\sqrt{2}}|0\rangle + \frac{i}{\sqrt{2}}|1\rangle$$
$$|-i\rangle := \frac{1}{\sqrt{2}}|0\rangle - \frac{i}{\sqrt{2}}|1\rangle$$

Рисунок 1.4–Приклади чистих кубітів

Інші кубіти потребують сумішей чистих станів, щоб повністю описати їх, тому ми називаємо їх кубітами змішаного стану. Іншими словами, кубіт змішаного стану описується розподілом ймовірності на чисті стани. Ми побачимо приклад змішаного стану кубітів пізніше.Досі ми визначали лише стан одного кубіта. Як виглядає комбінований стан декількох кубітів?Комбінований стан декількох кубітів - тензорний добуток усіх кубітів.Не хвилюйтеся, якщо ви не знаєте, що таке продукт тензору; ми розглянемо приклад ($\otimes$ символ роботи тензорного виробу).Загалом ми можемо тензорувати добуток будь-яких двох матриць, виконавши два кроки:

$$|0\rangle \otimes |1\rangle = \begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}1 \cdot \begin{pmatrix}0\\1\end{pmatrix}\\ \\0 \cdot \begin{pmatrix}0\\1\end{pmatrix}\end{pmatrix} = \begin{pmatrix}1\cdot 0\\1\cdot 1\\0\cdot 0\\0\cdot 1\end{pmatrix} = \begin{pmatrix}0\\1\\0\\0\end{pmatrix}$$

Рисунок 1.5– Тензорний добуток двох кубітів

Скалярно множте кожен елемент у першій матриці на всю другу матрицю.[7]

Поєднайте отримані матриці відповідно до вихідного положення їх елементів. Ось другий приклад того, як він працює для двовимірних матриць:

$$I \otimes H = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} 1 \cdot \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} & 0 \cdot \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \\ 0 \cdot \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} & 1 \cdot \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \end{pmatrix} =$$

$$\begin{pmatrix} 1 \cdot \frac{1}{\sqrt{2}} & 1 \cdot \frac{1}{\sqrt{2}} & 0 \cdot \frac{1}{\sqrt{2}} & 0 \cdot \frac{1}{\sqrt{2}} \\ 1 \cdot \frac{1}{\sqrt{2}} & 1 \cdot -\frac{1}{\sqrt{2}} & 0 \cdot \frac{1}{\sqrt{2}} & 0 \cdot -\frac{1}{\sqrt{2}} \\ 0 \cdot \frac{1}{\sqrt{2}} & 0 \cdot \frac{1}{\sqrt{2}} & 1 \cdot \frac{1}{\sqrt{2}} & 1 \cdot \frac{1}{\sqrt{2}} \\ 0 \cdot \frac{1}{\sqrt{2}} & 0 \cdot -\frac{1}{\sqrt{2}} & 1 \cdot \frac{1}{\sqrt{2}} & 1 \cdot -\frac{1}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}$$

Рисунок 1.6–Тензорний добуток для двовимірних матриць

Наприклад, ми можемо позначити кілька кубітів у позначенні Bra-ket як $|0\rangle \otimes |1\rangle$, наприклад. Як скорочення ми можемо опустити$\otimes$ і просто записати $|0\rangle|1\rangle$. Як ще коротша стенограма, ми можемо написати лише один кет, $|01\rangle$.Тепер давайте розглянемо, що таке квантові функції.Квантовий функції - це унітарна матриця.Розберемося, чому квантові функції є унітарними матрицями. Перш за все, квантові функціїбудуть реалізовані фізичними пристроями, і тому вони повинні дотримуватися законів квантової фізики. Одним із відповідних законів фізики є те, що жодна інформація ніколи не втрачається при переході між точками в минуле та майбутнє. Це відомо як унітарність. Оскільки наші квантові ворота визначають, як ми переходимо між джерелами, вони теж повинні дотримуватися унітарності.По-друге, зауважте, що наші квантові ворота будуть застосовані до кубітів. Ми раніше дізналися, що кубіти справді є просто векторами, і це означає, що квантові ворота повинні якось діяти на векторах. На щастя, ми нагадаємо, що матриця насправді є лише лінійним перетворенням для векторів. Поєднуючи ці дві ідеї, ми розглядаємо квантові ворота як унітарні матриці. Унітарна матриця - це будь-яка квадратна матриця складних чисел, така, що кон'югат транспозиції дорівнює її оберненій. Як швидке оновлення, кон'югат транспозиції матриці виявляється шляхом взяття кон'югату кожного елемента в матриці (a + bi → a - bi), а потім прийняття транспозиції матриці (елемент ij →

елемент ji). Ми зазвичай позначаємо кон'югат, транспонований кинджалом, †. Ключовим спостереженням щодо унітарних матриць є те, що вони зберігають норму (довжину вектора). Припустимо, ми допустили ворота, які змінили норму, то ймовірність нашого кубіта може дорівнювати чомусь іншому, ніж одному! Це не має сенсу, оскільки сума всіх ймовірностей завжди повинна дорівнювати одиниці.Також зауважте, що за визначенням унітарні матриці мають обернену форму. Одним із наслідків цього є те, що ми не можемо «призначити» кубіти довільним станам. Щоб зрозуміти, чому ні, зробимо вигляд, що у нас був квантовий затвор, який міг би «призначити» значення, отже, перетворити будь-який вектор із двох складних чисел у конкретний вектор із двох складних чисел. Цей квантовий затвор мав би деяке базове представлення як унітарна матриця, і ця матриця мала б обернену здатність перетворювати конкретний вектор назад у будь-який стан, у якому був кубіт до операції! Але кубіт міг бути в будь-якому стані до початку операції, і немає ніякого способу дізнатися який! Отже, ми не можемо «присвоїти» кубіти довільному стану. На більш високому рівні факт, що всі квантові ворота є незворотними, тому ми часто думаємо про квантові обчислення як про форму оборотних обчислень.[8]

Нарешті, зауважте, що, оскільки наші квантові ворота є унітарними матрицями, вони є квадратними за визначенням, і тому наші квантові ворота повинні мати рівну кількість кубітів введення та виведення (оскільки квадратні матриці відображають n стандартних базових векторів до n стовпців)! Це сильно відрізняється від більшості логічних воріт; наприклад, ворота AND приймає два входи і виробляє один вихід.Квантові ворота H і CNOT. Тепер, коли ми трохи знаємо про те, з чим працюємо, розглянемо приклад, ворота Хадамарда, H.

$$\begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}$$

Рисунок 1.7–Ворота Хадамара

Ми можемо перевірити, що H є унітарною, перевіривши, чи є спряжений кон'югат рівним його зворотній, або іншими словами, що H, помножений на його кон'югат, транспонується на матрицю тотожності:

$$HH^\dagger = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}^\dagger = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Рисунок 1.8– Перевірка унарності логічної функції

Іншию важливою квантовою функцією є функціяControlled NOT, також відома як CNOT. CNOT діє на два кубіти, контрольний кубіт і цільовий кубіт. Ми можемо розглядати CNOT як "if твердження" - якщо контрольний кубіт дорівнює 1, то CNOT застосовує HE (зворотний затвор) до цільового кубіту (звідси назва, контрольована HE).

Ось матриця, що представляє CNOT. Ця матриця розглядає кубіт керування як найменше право значення у кеті, а цільовий кубіт - як найменше ліве значення.[7][8]

[[1, 0, 0, 0], [0, 0, 0, 1], [0, 0, 1, 0], [0, 1, 0, 0]]

Давайте подивимось його вплив на |00⟩.

CNOT ket 00 = ket 00

У цьому прикладі ми бачимо, що CNOT не змінює значення |00⟩. І це очікувана поведінка, оскільки CNOT інвертує ціль лише у тому випадку, якщо управління дорівнює 1.

Давайте подивимось його вплив на |01⟩.

CNOT ket 01 = ket 11

Тут ми можемо побачити, що контроль дорівнює 1, тому CNOT перевертає ціль. Отже, результат - |11⟩.

Спробуйте опрацювати інші два випадки - |10⟩ та |11⟩. Ви повинні виявити, що CNOT має таку поведінку:

|00⟩ ->|00⟩

|01⟩ ->|11⟩

|10⟩ ->|10⟩

|11⟩ ->|01⟩

І зауважте, що саме така поведінка застосовується НЕ до цільового біта, коли біт керування дорівнює 1.

Для резюме ми можемо розглядати квантові ворота як єдині матриці. Ця унітаріальність примушує обмеження, що ймовірності кубіта дорівнюють одиниці і призводять до того, що квантові обчислення є оборотними. Оскільки унітарні матриці є квадратними, ми виявляємо, що квантові ворота повинні мати рівну кількість кубітів введення та виведення. Ми дізналися про Хадамард та CNOT, які є двома важливими квантовими воротами. Існує ще багато квантових воріт.[9]

1.3 Аналіз методів емуляції квантових розрахунків

У переважній більшості джерел з квантових обчислень читач знайде опису декількох алгоритмів, які, власне, зазвичай використовуються для демонстрації потужності обчислювальної моделі. Тут ми теж коротко і поверхнево розглянемо такі алгоритми (два з них, які демонструють різні базові принципи квантових обчислень). Ну а для детального ознайомлення з ними знову адресую до своєї нової книги.Це перший алгоритм, який був розроблений для того, щоб показати суть і ефективність квантових обчислень. Завдання, яке вирішує це алгоритм, зовсім відірвана від реальності, однак на ній якраз можна показати базовий принцип, покладений в основу моделі.Отже, нехай є деяка функція, яка отримує на вхід один біт і повертає на виході теж один біт. Чесно кажучи, таких функцій може бути всього 4. Дві з них є константними, тобто одна завжди повертає 0, а інша завжди повертає 1. Дві інші є збалансованими, тобто повертають 0 і 1 в рівних кількостях випадків. Питання: як за один виклик цієї функції визначити, константная вона або збалансована?Очевидно, що в класичній обчислювальної моделі цього зробити не можна. Необхідно двічі викликати функцію і порівняти результати. А ось в моделі квантових обчислень це зробити можна, оскільки функція буде викликана тільки один раз. Як вже було написано, ми підготуємо равновероятностних суперпозицію всіх можливих значень вхідного параметра функції. Оскільки на вході у нас один кубіт, то його равновероятностних суперпозиція готується за допомогою одного застосування гейтаАдамара. Далі знову застосовується гейтАдамара, який працює таким чином, що якщо йому на вхід подається равновероятностних суперпозиція, то він перетворює її назад в стану | 0> або | 1> в залежності від того, в якій фазі знаходиться равновероятностних суперпозиція. Після цього проводиться вимірювання кубіта, і якщо він дорівнює | 0>, то розглянута функція константна, а якщо | 1>, то збалансована.Що виходить? Як вже було сказано, при вимірюванні ми не можемо отримати всі значення функції. Але ми можемо зробити певні висновки

про її властивості. Завдання Дойча якраз і запитує про властивості функції. І це властивість дуже просте. Адже як виходить? Якщо функція константна, то додавання по модулю 2 всіх її вихідних значень завжди дає 0. Якщо ж функція збалансована, то додавання по модулю 2 всіх її вихідних значень завжди дає 1. Саме цей результат ми і отримали в результаті виконання алгоритму Дойча. Ми не знаємо, який саме значення повернула функція на равновероятностних суперпозиції всіх вхідних значень. Ми знаємо тільки, що це теж суперпозиція результатів, і якщо тепер цю суперпозицію перетворити спеціальним чином, то будуть зроблені однозначні висновки про властивості функції. Інший алгоритм, який показує квадратичний виграш в порівнянні з класичною обчислювальної моделлю, вирішує більш наближену до реальності завдання. Це алгоритм Гровера, або, як називає його сам Лов Гровер, алгоритм пошуку голки в стозі сіна. Цей алгоритм заснований на іншому принципі, що лежить в основі квантових обчислень, а саме ампліфікації.Уже згадувалася якась фаза, яка може бути у квантового стану в складі кубіта. Як такої фази немає в класичній моделі, це щось новеньке саме в рамках квантових обчислень. Фазу можна розуміти як знак у коефіцієнта при квантовому стані в суперпозиції. Алгоритм Гровера заснований на тому, що спеціально підготовлена функція змінює фазу у стану | 1>.

Алгоритм Гровера вирішує зворотну задачу.[10] Якщо є невпорядкованість набір даних, в якому треба знайти один елемент, що задовольняє критерію пошуку, алгоритм Гровера допоможе це зробити ефективніше, ніж простий перебір. Якщо простий перебір вирішує завдання за O (N) звернень до функції, то алгоритм Гровера ефективно знаходить заданий елемент за O (√N) звернень до функції.Алгоритм Гровера складається з наступних кроків:

1) Ініціалізація початкового стану. Знову готується равновероятностних суперпозиція всіх вхідних кубітів.

2) Застосування ітерації Гровера. Дана ітерація складається з послідовного застосування функції пошуку (вона визначає критерій пошуку елемента) і спеціального гейта дифузії. Гейт дифузії змінює коефіцієнти при квантових станах, обертаючи їх навколо середнього. Тим самим проводиться ампліфікація, тобто збільшення амплітуди шуканого значення. Фішка в тому, що здійснити застосування ітерації необхідно певну кількість разів ($\sqrt{2n}$), інакше алгоритм поверне не ті результати.

3) Вимірювання. Після вимірювання вхідного квантового регістра з великою ймовірністю буде отримано потрібний результат. Якщо необхідно збільшити достовірність відповіді, то алгоритм проганяється кілька разів і обчислюється сукупний ймовірність правильної відповіді.[11]

Тепер ми можемо перейти до розгляду алгоритму, якийнаробивнайбільше шуму, і, власне, через якого, на думку багатьох, нова обчислювальна модель, заснована на законах квантовоїмеханіки, отрималатакийрозвиток. Це алгоритм Шора для факторизаціїцілих чисел, які є творомдвохпростихнепарних чисел. Вся справа в тому, щосаме на гіпотезіалгоритмічноїскладностізавданняфакторизації числа заснованічисленнісучасніалгоритми і системикриптографії. ЗнайденийПітеромШором в 1994 році алгоритм дозволяє вирішитицю задачу за поліноміальний час і на поліноміальномукількостікубітів, в той час як кращікласичніалгоритмивирішуютьїї за суперполіноміальное (субекспоненціальное) час. А це значить, що як тількиквантовийкомп'ютер з достатньоюкількістюкубітів буде створений, вся сучаснакриптографіяопинитьсяпідзагрозоюкомпрометації. Власне, вона і буде скомпрометована тут же, оскільки будь-яка інформація, прихована з використаннямцьогопідходу, може бути отримана будь-якою особою, у кого є доступ до такого квантовому комп'ютера. Оскільки алгоритм Шораразючевідрізняєтьсявідусіхранішерозглянутихалгоритмів в

частинінаявностісерйозноїприкладноїзначущості, сам алгоритм більшскладний з точки зору математики і архітектури. Тут вжезадіянідвіобчислювальніпарадигми - класичначастинаготуєвхіднідланідля алгоритмуШора, а такожуправляє циклами і поверненнями з метою знаходження правильного результату; ну а квантовачастина, як цезазвичайбуває, просто виконуєлінійнупослідовністьунітарнихперетворень над спеціальнопідготовленими станами вхіднихкубітів. Отже, алгоритм факторизації Шора. Його суть полягає в зведеннізадачіфакторизації до задачіпошукуперіодуфункції. Якщовідомийперіодфункції, то факторизация здійснюється за допомогою алгоритму Евкліда за поліноміальний час на класичномукомп'ютері. І ось квантовачастина алгоритму факторизаціїякраззаймаєтьсяпошукомперіодуфункції. А класичначастина алгоритму спочаткуспеціальним чином готуєонуфункцію, а потімперевіряєзнайденийквантовоїчастиноюперіод на достатність для вирішеннязавдання. Якщоперіодзнайдений правильно (алгоритм імовірнісний, так щоможезнайти не те, щохочеться), то задача вирішена. Якщо ж ні, то квантовачастина 180 Душкін Р. В. алгоритму проганяється ще раз. А, оскільки, перевіркаправильностірішення для задачіфакторизаціїдуже проста (Помножили два числа так порівняли з третім), то цючастину алгоритму взагаліможна не брати до уваги з точки зорупідрахункускладності. Щоб не бути голослівними і занадтотеоретизувати, давайте спробуєморозглянути алгоритм факторизаціїШора на простому прикладі. Для цього просто розкладемо на простімножникидеяке число, яке є творомдвох і в точностідвохпростих чисел, жодне з яких не є числом 2. Оскількикомпанія IBM вжефакторізовано на своємупрототипі квантового комп'ютера число 15, тут пропонуєтьсяфакторізовать число 21, хочацевзагалі не принципово, оскільки головне в розгляді то, щобвистачилообчислювальнихпотужностей.[12]

1.4Огляд функціонального базису квантових логічних функцій

Вентиль кероване НЕ (controllednot, або CNOT) має два входи і два виходи. Перший вхід називається керуючим бітом. Якщо він 140 Глава 6. Класична логіка, вентилі і ланцюги дорівнює 0, тоді вентиль пропускає другий вхідний біт без зміни. Якщо керуючий біт дорівнює 1, тоді вентиль перетворює другий вхідний біт як вентиль НЕ. Біт управління є першим вхідним бітом і позначається як x. Цей біт передається на перший вихід без змін. Другий вихід дорівнює другого входу, якщо керуючий біт дорівнює 0, і отримує протилежне значення, якщо керуючийбіт дорівнює 1.

Вентиль Тоффолі, запропонований Томасом Тоффолі (TommasoToffoli), має три входи і три виходи. Перші два - керуючі біти. вони інвертують третій вхідний біт, якщо обидва рівні 1, інакше третій біт передається на вихід без змін. Так як цей вентиль подібний вентиля кероване НЕ (CNOT), але має два керуючих біта, іноді його називають кероване НЕ {CCNOT). Дія цього вентиля описується функцією: Г (x, г /, г) = (x, г /, (x y) z).

Вентиль Фредкіна також має три входи і три виходи. Перший вхід - керуючий біт. Якщо він дорівнює 0, другий і третій вхідні біти передаються на вихід без змін. Якщо керуючий біт дорівнює 1, він змінює другий і третій вхідні біти місцями - на другий вихід подається третій вхід, а на третій вихід - другий вхід.У класичній теорії обчислень для виконання операцій над бітами використовуються логічні вентилі. Для маніпуляцій над кубитами застосовуються аналогічні конструкції - квантові вентилі. Наприклад, вентиль NOT виконує перетворення 0 → 1 і 1 → 0. Квантовий вентиль NOT схожий на свого класичного предка: він виконує перетворення | 0> → | 1> і | 1> → | 0>. Це означає, що після проходження такого вентиля кубіт зі стану α | 0> + β | 1> перейде в стан α | 1> + β | 0>. Вентиль NOT можна записати у вигляді матриці (X), яка міняє місцями 0 і 1 в матриці стану: Як бачимо, X | 0> = | 1>, а X | 1> = | 0> Оскільки | 0> і | 1> в векторній формі записуються як і, перший стовпець матриці X можна розглядати як перетворення вектора | 0>, а другий - як

перетворення вектора | 1>. Здавалося б, відміну від класичного випадку не настільки велика. Але не забувайте, про що ми говорили в попередньому розділі: вимір стану кубіта носить імовірнісний характер. Як відомо з елементарної теорії ймовірностей, сума ймовірностей повної групи несумісних подій дорівнює одиниці. Тому для квантового стану α | 0> + β | 1>. Звідси випливає, що в квантовому світі можуть існувати не всі мислимі вентилі. Ось одне з обмежень: умова нормалізації квантового стану кубіта,, має дотримуватися як до проходження вентиля, так і після нього. У термінах матричної алгебри ця умова буде виконана в тому випадку, якщо матриця є унітарною. Я постараюся пояснити, що означає математичне поняття унітарності. Якщо прочитати його досить швидко, ви просто опинитеся на наступному реченні. Вентиль називається унітарною, якщо отримана шляхом транспонування і комплексного сполучення і є одиничною матрицею рангу 2.[13] Якщо говорити людською мовою, це означає, що перетворення не змінює довжину вектора. Якщо довжина вектора не змінюється з часом, то сума всіх ймовірностей незмінно дорівнює одиниці, або 100% (як і повинно бути). Викладки, в результаті яких сума всіх ймовірностей виявляється рівною 200% або 25%, були б позбавлені сенсу. Унітарні матриці захищають принаймні від такого божевілля (хоча в квантовому світі його залишається предостатньо). Хороші новини: це обмеження є єдиним. Зважаючи на це умови у деяких класичних вентилів немає квантового аналога, однак і у деяких квантових вентилів немає класичного прототипу. Далі ми розберемо найважливіші з квантових вентилів. Описані нижче вентилі будуть використовуватися в нашій першій квантової програмі, тому постарайтеся їх запам'ятати. Вентиль Z працює дуже просто: він зберігає компонент | 0> і змінює знак компонента | 1>. Його можна записати у вигляді матриці яка перетворює стану кубітів наступним чином: | 0> → | 0>, | 1> → - | 1> (пам'ятаєте, що перший стовпець матриці описує перетворення вектора | 0>, другий - перетворення вектора | 1>). Вентиль Адамара створює суперпозицію станів | 0> і | 1>, подібних розглянутим вище. Його матричний запис виглядає так: що відповідає

наступним перетворенням станів кубітів:, Більш детальна інформація про унітарних матрицях і про способи наочного подання вентилів наводиться в ресурсах, посилання на які містяться в розділі «Додаткові матеріали». Розглянемо щось більш звичне. Класичні біти існують не тільки поодинці, але і у вигляді сполучень: наприклад, 00, 01, 10 і 11. У квантових обчисленнях використовуються у схожому: | 00>, | 01>, | 10> і | 11>. Як і раніше, ймовірність отримати в результаті вимірювання величину 00 дорівнює, для 01 ймовірність дорівнює і т. д. Припустимо тепер, що ми хочемо виміряти стан не обох кубітів, а тільки першого. Імовірність отримати при цьому 0 дорівнює. Як ми пам'ятаємо, вимір змінює стан, тому після нього вектор матиме значення. Зверніть увагу на чисельник: ми прибрали всі складові, для яких перший біт дорівнює 1 (оскільки за умовою результат вимірювання дорівнює 0). Для того щоб вектор описував допустиме квантовий стан, необхідно, щоб квадрат сум амплітуд дорівнював одиниці (як до, так і після перетворення). Щоб ця умова виконувалася, ми додаємо нормувальний множник - величину, зворотну квадратному кореню з визначника. Роботу вентиля NOT ми вже розібрали. Наступний на черзі - вентиль CNOT (controlled-NOT, «кероване НЕ»). На його вхід подається два кубіта. Перший називається керуючим, другий - керованим. Якщо керуючий кубіт дорівнює | 0>, то стан керованого кубіта не змінюється. Якщо керуючий кубіт дорівнює | 1>, то до керованого кубіти застосовується операція NOT. Операцію CNOT можна інтерпретувати декількома способами. Подібно вентилів X, Z і H, її можна записати в матричній формі, яка позначається буквою U.Знайомлячись з класичними обчисленнями, ми бачили, що будь-яку булеву функцію можна виразити у вигляді ланцюга, що складається тільки з вентилів Фредкіна, тобто вентиль Фредкіна є універсальним. Ми також бачили, що вентиль І-НЕ (NAND), разом з розгалуженням, теж є універсальним. А чи існують універсальні квантові вентилі? У класичному випадку існує кінцеве число булевих функцій з даним числом змінних. Для однієї змінної існують тільки дві булеві функції, для двох змінних - чотири. У загальному випадку для n змінних існує 2n функцій. Але

ситуація з квантовими вентилями в корені інша. Як ми вже бачили, існує безліч вентилів, що впливають на один кубіт. Якщо взяти кінцеве число вентилів і з'єднати їх кінцевим числом способів, ми отримаємо кінцеве число ланцюгів. Тобто неможливо з кінцевого числа вентилів отримати нескінченне число ланцюгів. Н існують універсальні квантові вентилі? Таким чином, відповідь на питання про існування кінцевого числа квантових вентилів, які є універсальними, - «ні». Квантові вентилі і ланцюги при тому, що за допомогою кінцевого числа вентилів неможливо отримати всі можливі квантові ланцюга, дослідники показали, що існує кінцеве безліч вентилів, яке можна використовувати для апроксимації всіх можливих ланцюгів, але ми не будемо вдаватися в подробиці докази. Всі ланцюга, які нам знадобляться, можна побудувати з вентилів, описаних вище: п'яти, що впливають на один кубіт, і одного вентиля керованого НЕ, що впливає на два кубіта.[14]

1.5Аналіз систем поширеннявнутрішнього стану

Звичайний комп'ютер заснований на бітах: мова йде змінних, які мають тільки два можливих значення. Ми часто називаємо їх 0 і 1, хоча в контексті булевої алгебри ми можемо назвати їх «Справжні» і «Помилкові». З битами можна виконувати прості булеві операції, такі як NOT, AND і OR. Будь-яка змінна, більш складна, ніж біт (наприклад, int або float), являє собою просто набір з безлічі біт. Квантові комп'ютери засновані на квантових бітів або кубітах. Вони також мають два можливих значення, які ми можемо назвати 0 і 1. Але закони квантової механіки також допускають інші значення, які ми називаємо станами суперпозиций. В якомусь сенсі стану суперпозиції є значення, які існують між крайнощами 0 і 1. Ми можемо уявити кубіт як сферу, причому 0 і 1 розташовані на протилежних полюсах. Стану суперпозиції - всі інші можливі точки на поверхні. Справа не в тому, що кубіт може мати проміжне значення, наприклад 0,63; коли вимірюється стан кубіта, результат завжди дорівнює 0 або 1. Але в ході обчислення кубіт може діяти так, як якщо б

він був сумішшю станів - наприклад, 63% нуля і 37% одиниці. Іншим ключовим аспектом поведінки кубітів є інтерференція, явище, добре відоме у фізиці хвиль. Коли дві хвилі перекриваються, вони можуть підсилювати один одного (якщо піки і хвилеподібні спади збігаються), або вони можуть нівелювати один одного (якщо хвилі не відповідають фазі). Математично інтенсивність комбінованих хвиль в будь-якій точці визначається квадратом суми індивідуальних амплітуд хвиль. Коли дві амплітуди мають один і той же знак, інтерференція конструктивна; коли одна амплітуда позитивна, а інша негативна, що виникає в результаті деструктивна інтерференція дає інтенсивність менше, ніж у одній хвилі. Подібно хвилям, стану 0 і 1 кубіта мають амплітуди, які можуть бути або позитивними, або негативними. Залежно від знаків амплітуд квантова інтерференція може або збільшувати, або зменшувати ймовірність того, що певний стан буде спостерігатися, коли ми вимірюємо стан кубіта. Інтерференція грає роль у всіх цікавих алгоритмах для квантових комп'ютерів, тобто алгоритмах, які можуть дозволити такій машині перевершувати класичний комп'ютер. Загальна ідея організувати еволюцію квантової системи полягає в тому, щоб помилкові відповіді були придушені деструктивними перешкодами, а правильні відповіді посилювалися конструктивної інтерференцією. Таким чином, алгоритми використовують форму паралелізму, унікальну для квантових систем. Одним з останніх аспектів квантової дивацтва є заплутаність. Ви не можете змінити один кубіт всередині квантового регістра, залишивши інше без змін. Для початку кожна функція, обчислена квантовою системою, повинна бути повністю оборотною. Якщо машина отримує на вхід A для виведення результату B, то повинна бути можливість відновити A при наявності B. Кожна функція повинна мати однакову кількість входів і виходів. Одним махом це правило забороняє більшу частину арифметики. Звичайний алгоритм складання, наприклад, є незворотнім. Ви можете додати 3 і 4, щоб отримати 7, але ви не можете «від'єднати» 7, щоб відновити вихідні 3 і 4. Іншим забороною в квантових обчисленнях є копіювання кубіта (цей принцип називається теоремою про заборону

клонування). Також ви не можете довільно встановити або перезавантажити кубіти в середині обчислення. Спроба зробити це знищила б квантову суперпозицію. У сукупності обмеження на операції кубіта на увазі, що будь-яка квантова програма повинна мати архітектуру димохідної труби - інформація проходить прямо від одного кінця до іншого. Особливо важливо, що в структурі програми не може бути циклів, де управління передається назад в більш ранню точку. Відповідь на ці складності знаходять в мовах квантового програмування. Фактично, квантові комп'ютери являють собою гібридні пристрої: частково квантові і частково класичні комп'ютери. Використання елементів звичайного комп'ютера необхідно для обробки входів і виходів, щоб взаємодіяти з зовнішніми додатками. Таким чином, в одній програмі можна комбінувати класичний код і квантовий код.

Мало хто з нас звертав увагу на невеликій символ замочка, що з'являється в наших веб-браузерах кожен раз, коли ми заходимо на сайт інтернет-магазину, відправляємо і отримуємо емейли, перевіряємо наш банківський рахунок або кредитну карту. Однак він сигналізує про те, що онлайн-сервіси використовують HTTPS, веб-протокол, шифрує дані, які ми відправляємо по інтернету, і ті відповіді, що ми отримуємо. Ця та інші форми шифрування захищають різні електронні комунікації, а також такі речі, як паролі, цифрові підписи та історії хвороб. Квантові комп'ютери можуть підірвати цю криптографічний захист. Сьогодні ці машини поки недостатньо потужні, але вони швидко еволюціонують. Можливо, що не пізніше, ніж через десять років - а можливо, і раніше - ці машини зможуть стати загрозою для широко використовуваних методів криптографії. Саме тому дослідники і компанії, що займаються безпекою, наввипередки розробляють нові підходи до криптографії, які зможуть витримати майбутні квантові атаки хакерів. Існують два основних типи шифрування. Симетричне шифрування вимагає, щоб у відправника і одержувача були в наявності ідентичні цифрові ключі для шифрування і розшифровки даних, а асиметричне - або шифрування з

публічним ключем - використовує публічно доступна ключ, що дозволяє людям зашифровувати повідомлення для одержувача, одноосібно володіє приватним ключем, що дозволяє їх розшифровувати. Іноді два цих підходу використовуються разом. У випадку з HTTPS, наприклад, веб-браузери використовують публічні ключі для перевірки достовірності сайтів і отримання ключа для симетричного шифрування комунікацій. Мета - не дати хакерам за допомогою значних обчислювальних потужностей спробувати вгадати використовувані ключі. Для цього популярні криптографічні методи, включаючи RSA і шифрування за допомогою еліптичних кривих, зазвичай використовують т.зв. односторонні функції з потайним входом - математичні конструкції, які відносно легко обчислювати в одну сторону для отримання ключів, але дуже складно для зловмисника піддати зворотному інжинірингу. Хакери можуть спробувати зламати код, підбираючи всі можливі варіанти ключа. Але захищаються боку дуже ускладнюють їм це завдання, використовуючи пари дуже довгих ключів - як 2048-бітної RSA, що використовує ключі довжиною в 617 десяткових чисел. Перебір всіх можливих варіантів приватних ключів займе тисячі - якщо не мільйони - років у звичайних комп'ютерів. Оскільки вони можуть допомогти хакерам набагато швидше пробиратися через алгоритмічні потаємні ходи. На відміну від класичних комп'ютерів, що використовують біти, здатні приймати лише значення 1 або 0, квантові машини використовують кубіти, здатні одночасно представляти різні можливі стани, проміжні між 0 і 1 - це явище називається суперпозицією. Вони також можуть впливати один на одного на відстані завдяки такому явищу, як заплутаність. Завдяки цьому явищу додавання декількох додаткових кубітів може привести до експоненціальним стрибків обчислювальної потужності. Квантова машина з 300 кубитами здатна представляти більше значень, ніж кількість атомів в спостережуваного Всесвіту. Припускаючи, що квантові комп'ютери зможуть подолати деякі властиві їм обмеження, що стосуються швидкодії, коли-небудь їх можна буде використовувати для перевірки всіх можливих варіантів криптографічного

ключа за відносно недовгий час.[15] Хакери також, швидше за все, спробують використовувати на шкоду алгоритми оптимізації певних завдань. Один з таких алгоритмів, опублікований ловом Гровером з AT & T BellLabs, допомагає квантових комп'ютерів набагато швидше шукати варіанти. Інший алгоритм, опублікований в 1994 році Пітером Шором, тоді також працювали в BellLabs, а тепер професором в MIT, допомагає квантових комп'ютерів неймовірно швидко знаходити прості множники цілих чисел. Алгоритм Шора загрожує таким системам з публічним ключем, як RSA, чия математична захист, зокрема, залежить від того, наскільки складно провести зворотний інжиніринг результату перемноження дуже великих простих чисел (розкладання на множники). У звіті з квантових обчислень, опублікованому в минулому році національною академією наук, інженерної справи і медицини США, передбачається, що потужний квантовий комп'ютер, на якому працює алгоритм Шора, зможе зламувати 1024-бітові варіанти RSA менш, ніж за день.Малоймовірно. Дослідження національних академій стверджує, що для того, щоб представляти реальну загрозу, квантових комп'ютерів знадобиться набагато більше обчислювальної потужності, ніж є у найкращих з них на сьогодні. Однак, рік, в якому квантовий злом кодів стане серйозною головним болем - який деякі дослідники безпеки охрестили Y2Q - може підібратися несподівано швидко. У 2015 році дослідники зробили висновок, що для досить швидкого злому 2048-бітного RSA шифрування квантовому комп'ютера знадобиться мільярд кубітів. У більш сучасної роботі вказується, що комп'ютер з 20 млн кубітів зможе впоратися з цим завданням всього за 8 годин. Це знаходиться далеко за межами можливостей найпотужніших з сьогоднішніх комп'ютерів, що володіють всього 128 кубитами. Але прогрес квантових обчислень непередбачуваний. Без криптографічного захисту, що враховує квантові обчислення, всякі сервіси - від робомобіль до військового обладнання, фінансових транзакцій і комунікацій - схильні загрозу атаки з боку хакерів, які отримали доступ до квантових комп'ютерів. Будь-яке підприємство чи уряд, котре розраховує зберігати дані кілька десятиліть, має вже зараз замислитися

над тим, які ризики несе в собі нова технологія, оскільки використовується ними сьогодні шифрування може бути зламано в майбутньому. На перекодування величезних обсягів історичних даних в більш надійний вид можуть піти роки, тому краще буде застосовувати надійне кодування сьогодні. Звідси виникає запит на постквантовую криптографію. Це розробка нових видів криптографічних методів, які можна застосовувати з використанням сьогоднішніх класичних комп'ютерів, але які при цьому будуть невразливі перед завтрашніми квантовими. Одна з ліній оборони - збільшення розмірів цифрових ключів для значного збільшення кількості варіантів, в яких потрібно буде вести пошук перебором. Наприклад, просте подвоєння розміру ключа з 128 до 256 біт учетверяется кількість можливих варіантів, які доведеться перебрати квантової машині, що використовує алгоритм Гровера. Ще один підхід включає використання більш складних функцій з потайним входом, таких, щоб з ними важко було впоратися навіть потужному квантовому комп'ютера, виконуючому алгоритм Шора. Дослідники працюють над широким спектром підходів, включаючи такі екзотичні, як криптографія на гратах і протокол обміну ключами з використанням суперсінгулярнихізогенна. Мета досліджень - вибрати один або кілька методів, які потім можна буде широко застосовувати. Національний інститут стандартів і технологій США запустив в 2016 році розробку стандартів постквантового шифрування для урядового використання. Він уже звузив початковий набір заявок з 69 до 26, але каже, що перші чернетки стандартів, швидше за все, з'являться не раніше 2022 року. Критична важливість цього завдання пов'язана з тим, що технології шифрування глибоко впроваджені в безліч різних систем, тому на їх переробку і впровадження нових алгоритмів потрібно дуже багато часу. У дослідженні національних академій від минулого року відзначено, що на повне позбавлення від одного широко використовувався криптографічного алгоритму, який опинився вразливим, пішло більше 10 років. З огляду на швидкість розвитку квантових комп'ютерів, можливо, у світу залишилося не так вже й багато часу, щоб розібратися з цією новою проблемою безпеки. У грудні 2018 року вчені з

Google AI провели обчислення на кращому квантовому процесорі від Google. Їм вдалося відтворити ці обчислення на звичайному ноутбуці. Потім в січні вони запустили той же тест на поліпшеної версії квантового чіпа. На цей раз для симуляції результату їм знадобився вже потужний настільний комп'ютер. А до лютого у них вже не знайшлося класичних комп'ютерів, здатних симулювати своїх квантових суперників. Для цього дослідникам довелося запитувати процесорний час на величезній мережі серверів. «Десь у лютому мені довелося зробити кілька дзвінків і сказати:" Ей, нам потрібно побільше квоти ", - сказав ХартмутНивен, директор лабораторії квантового штучного інтелекту Google. "Ми запускали завдання, які вимагали мільйона процесорів". Таке швидке поліпшення привело до появи т.зв. "Закону Нівена", нового правила, що описує, наскільки швидко квантові комп'ютери наганяють класичні. Правило народилося, як внутрішнє спостереження, і тільки потім Нивен згадав про нього в травні на симпозіумі Google "Квантова весна". Там він розповів, що квантові комп'ютери збільшують обчислювальну потужність в порівнянні з класичними з "двічі експоненційної" швидкістю - приголомшливо швидкий рух. З подвійним ростом "спочатку здається, що нічого не відбувається, нічого не відбувається, а потім другий - і раптово ви вже в іншому світі, - сказав Нивен. - Саме це ми і спостерігаємо ".[16]

У 1950-х четверо військовослужбовців армії США, які захоплювалися математикою, використовували примітивні електронні калькулятори для розрахунку оптимальної стратегії гри в блекджек. Їх результати були пізніше опубліковані в журналі американської статистичної асоціації, і описували найкращі рішення, які може приймати гравець в будь-якій ситуації в грі. Однак така стратегія, яку любителі азартних ігор пізніше охрестять «правилами» [thebook], не гарантує перемоги гравцеві. У блекджека, а також пасьянсу, шашок або безлічі інших ігор, є певний «стелю» по відсотку ігор, в які гравець може виграти - навіть якщо він буде кожен раз грати ідеально. Однак існують особливо дивні ігри, в яких в принципі неможливо підрахувати максимальну

ймовірність виграшу. Замість цього математики та фахівці з інформатики намагаються визначити, чи можна хоча б дати приблизну оцінку відсотка виграшів для таких ігор. І існування цієї можливості залежить від сумісності двох дуже різних підходів до фізики. Такі «нелокальних» гри вперше придумав в 1960-м фізик Джон Стюарт Белл, намагаючись зрозуміти таке дивне квантове явище, як квантова заплутаність. Хоча заплутаність - штука складна, нелокальних гри за своєю суттю прості. Є два гравці, кожному з яких задають просте питання. Вони виграють, якщо їх відповіді певним чином пов'язані. На жаль, один з одним спілкуватися вони не можуть, тому їм доводиться здогадуватися про відповідь іншого. Белл довів, що якщо гравці зможуть використовувати пари заплутаних квантових частинок, вони можуть поліпшити кореляцію відповідей і вигравати ігри частіше, ніж можна було б очікувати. В останні роки дослідники розвивали роботи Белла, про що ми вже писали в статті "Прості квантові гри розкривають первинну складність Всесвіту". Робота Вільяма Слофстри від 2016 року і АндреаКоладанджело і Ялекса Старка від 2018 довели, що в деяких нелокальних іграх дотримується закономірність - чим більше пар заплутаних часток є у гравців, тим краще вони грають. І це взаємовідношення зберігається в нескінченності, тобто, для найкращої можливої гри гравцям потрібно нескінченну кількість пар частинок (або частки з безліччю незалежних властивостей). Один з наслідків цих результатів - неможливо підрахувати ймовірність максимального відсотка виграшу для деяких нелокальних ігор. Комп'ютери не працюють з нескінченними величинами, тому якщо ідеальна стратегія вимагає нескінченного числа заплутаних часток, комп'ютер не може підрахувати, як часто стратегія виправдовує себе. «Немає такого узагальненого алгоритму, щоб можна було ввести опис гри і отримати відповідь у вигляді ймовірності максимального відсотка виграшів», - сказав Генрі Юйень, фахівець з теоретичної інформатики з університету Торонто. Але якщо ми не знаємо точну вірогідність максимального відсотка виграшів, чи не можемо ми підрахувати її хоча б з якоюсь похибкою? Математики активно трудяться над цим питанням. Як не

дивно, їх успіх залежить від сумісності двох дуже різних підходів до фізики. Згадаймо, що гравцям в нелокальної грі не можна координувати відповіді. Цього можна досягти двома способами. Перший - фізично ізолювати їх один від одного, розмістивши їх в різних кімнатах або на різних кінцях Всесвіту. Просторова ізоляція забезпечує відсутність комунікацій. Дослідники аналізують цю ситуацію, використовуючи модель "тензорного твори". Однак є й інший спосіб не дати гравцям змовитися. Замість їх поділу можна висунути іншу вимогу: послідовність, в якій два гравці вимірюють заплутані частки і видають відповідь, не може впливати на їхні відповіді. «Якщо порядок, в якому вони проводять вимірювання, не має значення, то вони явно не можуть спілкуватися один з одним», - сказав Юйень. Коли в математиці порядок дій не впливає на відповідь, кажуть, що операція коммутативна: $a \times b = b \times a$. Такий підхід до нелокальним ігор - на основі незалежності послідовності, а не просторового поділу - називається моделлю «комутуючого оператора». Твір тензорів і коммутирующий оператор використовуються у фізиці, особливо при вивченні взаємодій субатомних частинок у квантовій теорії поля. Ці моделі - два різних підходи до міркувань про причинно-наслідкового незалежності фізичних явищ. І хоча модель твори тензорів більш інтуїтивна - ми зазвичай уявляємо собі причинно-наслідковий незалежність як просторове розділення - модель комутуючого оператора дає більше логічну математичну платформу. Все тому, що «просторова незалежність» - ідея розмита, а комутуюче взаємовідношення можна описати чітко. «Для людей, які вивчають квантову теорію поля, поняття просторового поділу об'єктів неприродно, - сказав Юйень. - На математичному рівні не завжди можна розмістити дві незалежні речі в двох окремих місцях Всесвіту ».

Не всі переконані в цьому. По-перше, класичні комп'ютери не стоять на місці. Звичайні чіпи продовжують поліпшуватися, навіть якщо закон Мура вже не працює. Крім того, фахівці з інформатики постійно придумують більш ефективні алгоритми, які допомагають класичним комп'ютерам не відставати. »З огляду на всі рухомі частини, включаючи поліпшення з класичної та

квантової сторін, важко назвати це зростання подвійним експоненціальним", - сказав Ендрю Чайлдс, один з директорів спільного центру квантової інформації та інформатики Мерілендського університету. І хоча точна швидкість, з якою квантові комп'ютери наздоганяють класичні, може бути предметом суперечок, сумнівів у швидкому поліпшенні квантової технології немає. «Думаю, що незаперечна реальність цього прогресу передала м'яч на сторону людей, які вважають, що масштабовані квантові комп'ютери не зможуть працювати, - написав Худоба Ааронсон, фахівець з інформатики з Техаського університету в Остіні, нам по емейл. - Тепер їм доведеться чітко сформулювати, де і чому зупиниться цей прогрес ».
Основна мета області квантових обчислень - виробляти ефективні квантові підрахунки, які неможливо симулювати за розумний час на найпотужніших класичних комп'ютерах (а найпотужнішим зараз вважається суперкомп'ютер SummitОк-Ріджской національної лабораторії). І серед різних дослідницьких груп, які розробляють квантові комп'ютери, Google особливо голосно заявляє про своє переслідування цієї мети, відомої, як «квантову перевагу». Поки що квантова перевага залишається невловимим - іноді здається, що його ось-ось досягнуть, але поки не вдається. Але якщо закон Нівена буде виконуватися, то до цієї мети залишилося недовго. Нивен не говорить, коли саме, на його думку, команда Google досягне квантового переваги, але допускає, що це може статися скоро.[17]

1.6Висновок аналізу методів емуляції

У даному розділі зроблено огляд та виконано аналіз методівемуляції квантових логічних функцій і розрахунків за допомогою мов програмування високого рівня. Проаналізовані відомі варіанти побудови системи поширення внутрішнього стану для віртуальних кубітів, який буде кращим підходом до емуляції квантових логічних фкнкцій, які пропонується реалізувати у вигляді програмного продукту.

2РОЗРОБКА ТЕХНОЛОГІЇ ЕМУЛЯЦІЇ КВАНТОВИХ ЛОГІЧНИХ ФУНКЦІЙ І РОЗРАХУНКІВ

В данномурозідлі описаний процес розробки алгоритму емуляції квантових логічних функцій і розрахунків. В залежності від вибраного способу емуляції логічних функцій і розрахунківці етапи можуть бути реалізовані у різній послідовності. Розробці технології та алгоритмуформування ознак об'єктів цифрових зображень присвячений даний розділмагістерської роботи.

2.1Розробкавіртуальноїчастинки та системивзаємодії

Перейдемо до кубітів. Як було визначено в попередньому розділі, кубітом називається просто список квантових станів. Ось тут-то і криється причина, чому поле label у визначенні типу QuantumState має тип String. Справа все в тому, що з точки зору реалізації немає ніякої різниці, чи є у кубіта два базисних стану, або їх довільне число (але, як відомо, рівне ступеню двійки). Чим один кубіт в двомірному гільбертовому просторі відрізняється від многокубітовой системи в 2 n -мірному просторі? Та нічим! Тому визначення кубіта в даному випадку буде виглядати як ізоморфний тип, що приховує в собі список квантових станів: newtypeQubit a = Qubit [QuantumState a] І тепер за допомогою цього типу можна цілком собі уявляти многокубітовие квантові системи, у яких позначки у індивідуальних квантових станів вже не є просто символами, але є рядки (наприклад, | 01⟩). Насамперед для цього нового типу можна реалізувати екземпляр класу Show для перетворення в рядки і приємного відображення на екрані.Для квантових функцій є єдина подвійна функція fromLists, яка отримує на вхід два списки - список комплексних амплітуд і список міток. Однак перетворення кубіта в список - це занадто просто, так само як і створення кубіта зі списку. Всі ці функції є службовими, які годяться тільки для спрощення процесу побудови описів кубітів. В математиці квантової механіки,

як це було зазначено в першому розділі, є векторні уявлення кубітів. Має сенс реалізувати функції для перетворення кубіта у внутрішньому поданні (тип Qubit) в векторне і назад. Тут буде показано трохи функціональної магії, але зовсім небагато, оскільки справжні функції для таких перетворень повинні відстежувати численні нестандартні ситуації, а у що наводиться тут визначенні всі вони або залишаються за реалізацією, або скидаються в виклик системної функції error, яка зупиняє виконання програми. Отже, ось визначення функції, яка переводить кубіт в векторне подання:

toVector :: Num a =>Qubit a -> [Complex a] toVector q = ifall `elem`" 01 ") labelsthenmap (fromMaybe (0: + 0). fliplookupqsPairs) basiselseerror" Некоректні мітки кубіта . " where n = length $ label $ head $ quantumStates q labels = concatMaplabel $ quantumStates q qsPairs = map (swap. toPair) $ sortBy (comparinglabel) $ quantumStates q basis = replicateM n "01"

Для початку необхідно розглянути всі локальні визначення. Замикання n представляє обчислення кількості кубітів в квантовій системі, поданої на вхід. Тут є логічна помилка, оскільки якщо в поданому на вхід описі кубіта є мітки різної довжини (чого бути не може, але фреймворк цього не перевіряє), то в якості значення n візьметься довжина першої мітки. Також обчислення цього замикання небезпечно, оскільки програма може вивалитися, якщо у кубіта взагалі немає квантових станів (цього теж бути не може, але фреймворк знову ж таки не відстежує). Друге замикання labels повертає зчеплені мітки всіх квантових станів кубіта. Це службове замикання, яке використовується тільки в перевірці того, що в мітках Квантові обчислення та функціональне програмування квантових станів кубіта використовуються тільки символи з обчислювального базису, тобто «0» і «1». Знову ж таки, якщо так вийшло, що у кубіта немає квантових станів, то функція в цілому відпрацює некоректно (поверне порожній векторне подання). Замикання qsPairs повертає всі квантові стани кубіта, відсортовані по мітках і перетворені в пари (тобто вийшов

асоціативний список), причому першим елементом пари йде мітка, а другим - амплітуда. Це потрібно для пошуку відсутніх квантових станів, якщо кубіт є змішані квантові стану. Ну і, нарешті, замикання basis повертає всі можливі рядки із символів «0» і «1» довжиною n. Тут використана Монадический функція replicateM для монади списку, яка і вирішує цю задачу. В результаті для n = 2, наприклад, виходить список ["00", "01", "10", "11"]. Сама функція toVector бере цей список базисних квантових станів (basis) і кожне з них шукає в асоціативному списку, отриманому з кубіта (qsPairs). Якщо мітка знайдена, то в результат записується відповідна амплітуда. Якщо мітка не знайдено, то в результат записується число 0. Так що, наприклад, кубіт | Φ +⟩ = 1 √2 | 00⟩ + 1 √2 | 11⟩ буде перетворений в список [1 √2, 0,0, 1 √2]. Як не дивно, зворотна функція fromVector в порівнянні з тільки що певною, виглядає зовсім просто. Єдиний нюанс - щоб не обчислювати логарифм за основою 2, другим параметром в неї необхідно передавати кількість кубітів в квантовій системі - це число потрібно для автоматичної генерації всіх необхідних міток в обчислювальному базисі. Якщо зробити визначення таким, то воно виглядає зовсім просто:fromVector :: Int -> [Complex a] ->Qubit a fromVectornq = fromLists q $ replicateM n "01" Тут видно те ж саме побудова списку всіх можливих комбінацій рядків довжини n із символів «0» і «1», на підставі якого далі будується кубіт за допомогою вже певної функції fromLists. Само собою зрозуміло, що довжина векторного уявлення кубіта повинна бути в точності відповідної. Фреймворків це знову ж таки не перевіряється. Важливою сервісного режиму, яка виконує приблизно ті ж дії, що і функція applyQS для квантових станів, є функція вищого порядку для «втягування» заданої функції в кубіт і застосування її до безлічі його квантових станів. Ось її визначення: liftQubit :: ([QuantumState a] -> [QuantumState b]) ->Qubit a ->Qubit b liftQubit f (Qubitqs) = QubitЧи не головною функцією для обробки кубітів, яка повинна бути описана в цьому модулі, є функція entangle, яка отримує на вхід два кубіта, а повертає один, що представляє собою зчеплену квантову систему з двох вхідних кубітів.[18]

2.2 Розробка способу поширення внутрішнього стану

Тут представлені різні квантові комп'ютери і програмні забезпечення, які використовуються для підключення до пристроїв. В даний час, чотири програмні платформи дозволяють підключатися до чотирьох різних квантових комп'ютерів - один від Rigetti, 8-ми кубітний квантовий комп'ютер, підключитися можна за допомогою pyQuil [41]; і три від IBM, з найбільшим доступним кількістю в 16 кубіт, які можуть бути підключені за допомогою QISKit або ProjectQ. Крім того, IBM пропонує четвертий 20-кубітний квантовий комп'ютер, але це пристрій доступно тільки членам IBM Q Network [42]: групі компаній, університетів і національних лабораторій, зацікавлених і інвестують в квантові обчислення. На Фігурі 1 також показані квантові комп'ютери таких компаній, як Google, IBM і Intel, які були оголошені, але в даний час недоступні для звичайних користувачів. Фігура 1. Схематична діаграма, що показує способи підключення персонального комп'ютера до використовуваному квантовому комп'ютера на гейтовом рівні. Починаючи з персонального комп'ютера (знизу в центрі), вузли зеленого кольору показують програмне забезпечення, яке можна встановити на персональний комп'ютер користувача. Сірі вузли показують, що симулятори запускаються локально (тобто на комп'ютері користувача). Пунктирні лінії показують API / хмарні підключення до ресурсів компанії, показаним в жовтих «хмарах». Квантові симулятори і використовуються квантові комп'ютери, що забезпечуються цими хмарними ресурсами, показані в синьому і золотом відповідно. Червоні рамки показують вимоги на обраному способі. Наприклад, щоб підключитися до RigettiForest і використовувати квантовий комп'ютер Agave 8 qubit, необхідно завантажити та встановити pyQuil (доступний на MacOS, Windows і Linux), зареєструватися на веб-сайті Rigetti для отримання ключа API, а потім запросити доступ до пристрою через онлайн- форму. Примітки: (i) Квантова віртуальна машина Rigetti вимагає підвищення прав для більш 30 кубітів, (ii) локальні симулятори залежать від комп'ютера користувача, тому наведені цифри є приблизними, і

(iii) в сірій рамці показані квантові комп'ютери, які були анонсовані, але які в даний час недоступні для звичайних користувачів. Технологія квантового обладнання швидко змінюється. Досить імовірно, що до кінця року з'являться нові комп'ютери, і через два-три роки цей список може бути повністю застарілим. Однак, що залишиться, так це програмне забезпечення, що використовується для підключення до цієї технології. Було б дуже просто використовувати нові квантові комп'ютери, змінивши лише кілька рядків коду, не змінюючи фактично синтаксис, який використовується для генерації або запуску квантової схеми. Наприклад, в QISKit потрібно просто змінити ім'я backend-пристрої при виконанні схеми: execute (quantum_circuit, backend = "name", ...) Лістинг 1. Рядок «name» вказує на backend-пристрій для запуску квантових програм з використанням QISKit. Коли майбутні квантові комп'ютери будуть випущені, виконання на новому обладнанні буде такий же простий, як і зміна імені. Хоча програмне забезпечення також змінюється з виходом нової версії [25], це, здебільшого, відносно незначні синтаксичні зміни, які суттєво не змінюють функціональність програмного забезпечення. У цьому розділі ми по черзі розглянемо кожну з чотирьох платформ, обговорюючи вимоги і установку, документацію і керівництва, синтаксис мови, квантовий мову, квантове обладнання і можливості симулятора. Цей огляд не призначений для повного навчання мови, але він дає читачеві розуміння про кожній платформі, перш ніж зануритися в одну (або більше) з обраних платформ. Поточний аналіз включає в себе достатню інформацію для запуску алгоритмів на квантових комп'ютерах. Проте, читач, коли він вибрав певну платформу, направляємо на спеціальну документацію для отримання повної інформації.

Були додані посилання на документацію і джерела навчальних посібників для кожного програмного пакету. Також передбачається, що є базові знання про квантових обчисленнях, для чого зараз існує безліч хороших посилань [21, 22]. Всі фрагменти коду і програми, включені в цей документ, були протестовані і запущені на ноутбуціDell XPS 13 DeveloperEdition під управлінням

LinuxUbuntu 16.04 LTS, повні специфікації якого перераховані в [23]. Хоча все програмні пакети працюють у всіх трьох основних операційних системах, з досвіду автора значно простіше встановити і використовувати програмне забезпечення на платформі, на якій вона була розроблена. У середовищі LinuxUbuntu при установці цих пакетів програмного забезпечення не виникало ніяких труднощів і незвичайних повідомлень про помилки.Термін «підтримка на рівні бібліотек» використовується для позначення прикладів квантових алгоритмів (в навчальних програмах) або конкретної функції для квантового алгоритму.[19]

Ми вже згадали деякі з них в попередньому розділі. Більш детальна таблиця, що показує підтримку на рівні бібліотек для чотирьох програмних платформ. Зауважимо, що будь-який алгоритм, звичайно, може бути реалізований на будь-який з цих платформ. Тут ми виділимо існуючу функціональність, яка може бути корисною для користувачів, які є новачками в цій області або навіть для досвідчених, щоб не програмувати все самим.Як видно, pyQuil, QISKit і QDK мають відносно велику бібліотечну підтримку. ProjectQ містить FermiLib, плагіни для FermiLib, а також він сумісний з OpenFermion, які є проектами з відкритим вихідним кодом для алгоритмів квантового моделювання. Всі приклади, які працюють з цими проектами, природно, працюють з ProjectQ. Microsoft QDK відрізняється кількістю вбудованих функцій, які виконують ці алгоритми автоматично, без необхідності явно програмувати квантову схему. Зокрема, бібліотека QDK пропонує детальну ітеративну фазу оцінки, важливу процедуру в багатьох алгоритмах, які можуть бути легко реалізовані на QDK, не жертвуючи адаптацією.

QISKit відрізняється великою кількістю навчальних матеріалів з широкого кола тем від фундаментальних квантових алгоритмів до повчальних квантових ігор.У цьому розділі ми обговоримо тільки pyQuil і QISKit, оскільки це єдині платформи зі своїм власним квантовим обладнанням. Кубіт є важливою характеристикою в квантових комп'ютерах, але не менш важлива - якщо не

важливіше - «якість кубіта». Під цим мається на увазі час когерентності (як довго кубіти живуть до моменту руйнування в біти), час на застосування гейта, коефіцієнт помилки гейта і топологію / зв'язність кубітів. В ідеалі можна було б мати нескінченні час когерентності, нульовий час на застосування гейта, нульовий коефіцієнт помилок і зв'язаність багато-до-багатьох. У наступних параграфах ми опишемо деякі з характеристик IBMQX5 і Agave, двох найбільших загальнодоступних квантових комп'ютерів. Детальну інформацію дивіться в онлайн-документації кожної платформи.

2.3Алгоритм для реалізації емуляції базових логічних функцій

IBMQX5 - 16-ти кубітний надпровідний квантовий комп'ютер з обмеженістю найближчих сусідів .Мінімальний час когерентності (T2) становить мікросекунд на 0-му кубіті, а максимум - мікросекунд на 15-му кубіті. Однокубітномугейт необхідно 80 наносекунд на виконання і плюс 10 наносекунд на амортизацію після кожного імпульсу. Гейт необхідно приблизно два-чотири рази більше, починаючи з 170-ти наносекунд для до 348 наносекунд для. Якість відтворення однокубітовогогейта дуже хороша з точністю понад 99,5% для всіх кубітів (точність = 1 - помилка). Мульти-кубітная точність вище 94,9% для всіх пар кубітів в топології. Найбільша помилка зчитування досить велика приблизно на 12,4% при середньому близько 6%. Ці статистичні дані були отримані з [30] .Нарешті, ми згадаємо, що для використання будь-якого доступного квантового комп'ютера IBM користувач відправляє свою роботу в чергу, яка визначає, коли запуститься завдання. Це є відмінністю від використання Agave від Rigetti, в якому користувачі повинні спочатку запитувати доступ через онлайн-форму, а потім запланувати час, щоб отримати доступ до пристрою для запуску завдань. З досвіду автора це робиться по електронній пошті, і персонал дуже чуйний.Квантовий комп'ютер Agave складається з 8 надпровідних трансмонових (transmon) кубітів з фіксованою ємнісний зв'язком і связностью, показаної на Фігурі 2. Мінімальний час когерентності (T2) становить 9,2 мікросекунди на 1-му кубите, а максимум становить 15,52 мікросекунди на 2 -омкубите. Час реалізації на гейтControlled- становить від 118 до 195 наносекунд. Точність однокубітовогогейта становить в середньому 96,2% (знову ж, точність = 1 - помилка) і мінімум 93,2%. Точність мульти-кубитовуюгейта в середньому становить 87% для всіх пар кубіт-кубіт в топології. Помилки зчитування невідомі. Ці статистичні дані можна знайти в онлайн-документації або через pyQuil. Платформи, які забезпечують підключення до реальних квантовим пристроїв,

обов'язково повинні мати засіб перекладу даної схеми в операції, які комп'ютер може зрозуміти. Цей процес відомий як трансляція чи більш докладніше трансляція квантової схеми / квантова трансляція. Кожен комп'ютер має базовий набір гейтов і певну зв'язаність - завдання компілятора полягає в тому, щоб отримати задану схему і повернути еквівалентну схему, яка підпорядковується вимогам базису і вимогам до пов'язаності. У цьому розділі ми обговоримо тільки QISKit і Rigetti, оскільки це платформи з реальними квантовими комп'ютерами.Приклад однаковою квантової схеми, компілювати обома платформами, показаний на Фігурі 5. Тут, використовуючи pyQuil, транслюється програма під характеристики Agave, а з QISKit - IBMQX5. Як видно, QISKit створює довше схему (т. Е. Має велику глибину), ніж pyQuil. Недоцільно стверджувати, що один з компіляторів є переважаючим через це приклад. Схеми на мові, який розуміє IBMQX5 природніше, приведуть до більш короткої схемою, ніж ті ж на pyQuil, і навпаки. Відомо, що будь-яка квантова схема (унітарна матриця) може бути розкладена на послідовність одно- і двох-кубітнихгейтов (дивіться, наприклад, [15]), але в цілому для цього необхідна значна кількість гейтов. Зараз значний інтерес представляє собою питання [16], що полягає в пошуку оптимального компілятора для заданої топології.Не всі програмні платформи забезпечують зв'язок з реальними квантовими комп'ютерами, але будь-яка доцільна програма включає в себе симулятор квантової схеми. Це програма, яка працює на класичному процесорі, який моделює (симулює) процес зміни квантового комп'ютера. Як і у випадку з квантовим обладнанням, важливо розглянути не тільки кількість кубітів, які може обробляти симулятор, але і як швидко він може обробляти їх, на додаток до інших параметрів, таким як додавання шуму для моделювання квантових комп'ютерів і т. Д. Спочатку згадаємо про продуктивність симулятора QVM pyQuil.Симулятор Rigetti, званий QuantumVirtualMachine (QVM), що не запускається на локальному комп'ютері користувача, а швидше за рахунок використання обчислювальних ресурсів в хмарі. Як уже згадувалося, для цього потрібно ключ API. Більшість ключів API дають доступ до 30-ти кубіти

спочатку, і може бути запитаний доступ на більшу кількість. Автор може імітувати 16-ти кубітную схему і глибиною 10 в середньому за 2,61 секунди. Розмір схеми в 23 кубіта і глибиною 10 була промоделювати за 56,33 секунди, але ніякі більші схеми не могли бути змодельовані, тому що QVM завершує роботу після однієї хвилини обчислення за допомогою поточного ключа доступу автора до API. Через обмеженого часу і з-за того, що QVM не починається на локальному комп'ютері користувача, ми не перевіряємо продуктивність QVM так само, як тестуємо симулятори ProjectQ і QISKit. QVM містить складні і гнучкі моделі шуму для емуляції процесу роботи реального квантового комп'ютера. Це є ключем до розробки алгоритмів малу глибину на квантових комп'ютерах в найближчому майбутньому, а також для теоретичного оцінювання результатів обчислень для конкретного квантового чіпа. Користувачі можуть визначати довільні моделі шуму для тестування програм, зокрема визначати зашумленігейти, додавати шум декогеренції і шум зчитування моделі. Докладні відомості та корисні приклади програм дивіться в розділі NoiseandQuantumComputation документації pyQuil. ISKit має кілька квантових симуляторів, доступних в якості бекенд. Відмінною особливістю цих симуляторів є принцип моделювання квантових схем. Унітарний симулятор реалізує базову (унітарна) матричне множення і різко обмежується оперативною пам'яттю. У симуляторі векторного стану не зберігається повна унітарна матриця, а завантажується тільки вектор стану і один / кілька кубітнийгейт. Обидва методи обговорюються в [23], а [24-26] містять детальну інформацію про інші методи. Подібно міркування про в ProjectQ, здатний ефективно імітувати групових схем (stabilizercircuits), які не є універсальними.[19]

Використовуючи локальний унітарний симулятор, схема на 10 кубітів і глибиною 10 моделюється за 23,55 секунди. Додавання ще одного кубіта збільшує час приблизно в десять разів - 239,97 секунди, а при 12 кубітах симулятор перевищує час очікування через 1000 секунд (близько 17 хвилин).

Цей симулятор швидко досягає тривалих часів моделювання та обмеження пам'яті, оскільки для кубітів повинна зберігатися в пам'яті унітарна матриця розміру.[20]

Симулятор векторного стану значно перевершує унітарний симулятор. Ми можемо змоделювати схеми з 25 кубітів всього за три хвилини. Всі схеми до 20 кубіт з глибиною до тридцяти моделюються менш ніж за п'ять секунд.Доброю особливістю pyQuil є Grove, розташований в окремому сховищі на GitHub, який може бути встановлений з навчальними посібниками і прикладами алгоритмів з використанням pyQuil. Rigetti також створює солідне співтовариство користувачів, прикладом якого є їх виділений канал в Slack для RigettiForest. Також корисний компілятор Quil і можливість компіляції для будь-якого заданого набору інструкцій архітектури (топологія і базові гейти). Нарешті, pyQuil сумісний з OpenFermion [37], з відкритим вихідним кодом Python для компіляції та аналізу квантових алгоритмів для моделювання Ферміон систем, включаючи квантову хімію. QISKit також доступний користувачам з досвідом роботи на мовах JavaScript і Swift. Для початківців мову Python є дуже хорошим стартовим мовою програмування через його легкого і інтуїтивного синтаксису. Подібно Grove, QISKit також містить спеціальний репозиторій прикладів алгоритмів і посібників. Крім того, бібліотека ACQUA в QISKit містить численні алгоритми для квантової хімії і штучного інтелекту. Ця бібліотека може бути запущена через графічний користувальницький інтерфейс або з інтерфейсу командного рядка. IBM не має собі рівних для створення активної спільноти студентів і дослідників, що використовують їх платформу. Компанія може похвалитися більш ніж 3 мільйонами віддаленими запусками хмарних квантових обчислювальних ресурсах, використовуючи QISKit, якими керують більш 80000 зареєстрованих користувачів, і було написано понад 60 наукових публікацій, написаних з використанням технології від IBM [21]. QISKit також має виділений канал в Slack з можливістю перегляду завдання в черзі, що є корисною функцією для

визначення того, як довго буде виконуватися відправлене завдання. Крім того, новітня версія QISKit містить вбудований Рисователь схем. Аналогічно, ProjectQ містить Рисователь схем. Додавши лише кілька рядків коду в програму, можна створити код на TikZ для створення високоякісних зображень в. Всі квантові схеми в цій статті були зроблені з використанням ProjectQ. Локальний симулятор ProjectQ також є відмінною особливістю, так як він володіє дуже високою продуктивністю. Хоча ProjectQ не має власного квантового обладнання, користувачі можуть підключатися до квантовому обладнання від IBM. Крім того, ProjectQ має кілька бібліотек, включаючи OpenFermion, як згадувалося вище. QDK був доступний виключно в Windows до тих пір, поки в лютому 2018 року його не отримав підтримку в macOS і Linux. Можливість реалізації квантових алгоритмів без явного програмування схеми є приємною особливістю QDK, а також є багато хороших керівництв в документації і папці прикладів для квантових алгоритмів. Також примітно, що Q # забезпечує функції автоматичного генерації для, наприклад, пов'язаною або керованої версії квантової операції. У більш загальному сенсі QDK виділяє і пропонує важливі інструменти для розробки продуктивного квантового алгоритму, включаючи тестування квантових програм, оцінку потреб в ресурсах, програмування на різних моделях квантових обчислень, орієнтованих на різні апаратні засоби, і забезпечення правильності квантових програм на стадії компіляції. Ці особливості мають ключове значення для переходу до високорівневих мов квантового програмування.На цьому етапі передбачається, що у читача буде достатньо інформації і розуміння, щоб прийняти обгрунтоване рішення про те, яка (-і) платформа (-и) для квантового програмного забезпечення підходить для нього. Наступний крок - почати читати документацію по платформі, встановити її і почати кодування. За короткий час можна запустити алгоритми на реальних квантових пристроях і почати дослідження / розробку алгоритмів в своїй області.Для тих, хто ще не визначився, пропонуються наступні суб'єктивні рекомендації. Для тих, чия основна мета є використання квантових комп'ютерів, QISKit (або ProjectQ) або

pyQuil - очевидний вибір.Для тих, хто новачок в квантових обчисленнях, QISKit, pyQuil або QDK - хороший вибір.Для тих, у кого мало досвіду в програмуванні, одна з платформ на Python - хороший вибір.Для тих, хто знайомий або вважає за краще синтаксис стилю C / C #, QDK - хороший вибір.Для тих, хто хоче розробити, прототип і тестові алгоритми, ProjectQ - хороший вибір.Для тих, хто хоче використовувати гібридні квантово-класичні алгоритми, pyQuil - відмінний вибір для виділеного запланованого використання апаратного часу.Для тих, хто цікавиться безперервними змінними в квантовому обчисленні, дивіться StrawberryFields.[22]

2.4Висновкиновоїтехнологіїемуляціїквантовихрозрахунків

У даному розділі запропоновано метод емуляції логічних функцій і розрахунків.Особливістю запропонованогометоду є збільшена ефективність і якісно новий рівень стабільності системи. Розроблено спосіб емуляції логічних функцій і розрахунків. Створено алгоритм по поширенню віртуальних квантових частинок.

3 РОЗРОБКА ПРОГРАМНИХ ЗАСОБІВ ЕМУЛЯЦІЇ КВАНТОВИХ ЛОГІЧНИХ ФУНКЦІЙ І РОЗРАХУНКІВ

Розділ присвячений розробці програмних засобів емуляції квантових логічних фенкцій та розрахунків. Запропонована розробка є подальшим покращенням уже існуючих додатків, що мають на меті бути корисними для кінцевих споживачів.

3.1 Вибір інструментальних засобів програмування

Що можеперешкодитиспінах ядер зберігатиквантовуінформацію в нашомумозку? Вінсклав список речей, якіможутьзруйнуватиквантовуінформацію, і прийшов до висновку, щоіониводнюпредставляютьнайбільшузагрозу. Вони можутьзаплутуєтьсязіспінами (і приводити до декогеренції) через диполь-дипольнавзаємодія.[23] Як спин можеуникнутиціїзагрози? Наприклад, спін величиною обнулить електричнийквадрупольний момент ядра, квадрупольнівзаємодії не зможуть привести до декогеренції такого спина. А в яких атомах в нашомутілі спин дорівнює? У водні і фосфорі. Тількиводеньсхильнийіншимиджереламидекогеренції, так щоМеттьюприйшов до висновку, щоатоми фосфору можутьзберігати КІ в нашомумозку, при цьомуспини ядра фосфору працюють як кубіти (квантовібіти). Віделектричнихвзаємодій фосфор захищений, а як щодомагнітних диполь-дипольнихвзаємодій?

Таківзаємодіїзалежатьвідорієнтаціїспінівщодоїхположення в просторі. Якщо фосфор є частиноюмаленькоїмолекули, щобовтається в біологічноїрідини, положення ядра змінюєтьсявипадковим чином, і в середньомувзаємодіявиявитьсянульовим. У молекулах є й іншіатомикрім фосфору. Ядра цихатомівможутьвзаємодіятизіспіном фосфору, і руйнуватийогоквантовий стан. Цього не станетьсятільки в одному випадку:

коли все спиниць ядер дорівнюють нулю. У яких атомах в тілілюдиниспини ядра дорівнюють нулю? У кисні і кальції. Так що фосфор виявитьсязахищенимвідвзаємодії з іншими атомами в молекулах з кальцієм і киснем. Проедложеносвійваріантмолекули, яка б захищала фосфор віддекогеренції. А потімвиявив, щотака молекула дійсно описана в науковійлітературі. Молекула підназвою кластер Познера або молекула Познера (я буду називатиїї Познер для стислості).[24] Познер можутьіснувати в штучнихбіорідинах - рідинах, створених для імітаціїрідинвсередині нас. Вважається, що Познер можутьіснувати в наших тілах і брати участь в утвореннікісток. Меттьооцінює, що Познер можутьзахищатиспини фосфору віддекогеренціїпротягом 1-10 днів. Але як Познер можутьвпливати на свідомість? Меттьюзапропонувавнаступнийваріант. Молекула аденозину трифосфату (АТФ) є джереломенергії для біохімічнихреакцій. «Трифосфат» означає, що в ній три іона фосфату - з'єднання, щоскладається з одного атома фосфору і трьохатомівкисню. Два фосфату можутьвідокремитисявідмолекули АТФ, залишаючисьз'єднанимиміж собою. Пара фосфатів буде дрейфувати, поки не зустріне фермент підназвоюпірофосфатази. Цей фермент можерозділити пару фосфатів на два незалежних фосфату. При цьому, як припустив Меттью разом з Лео Раджіховскім, спини ядер фосфору проектуються в синглетний стан, яке є станом з максимальною заплутаністю.[25]

Уявітьсобібезлічфосфатів в біорідини. Шістьфосфатівможутьоб'єднатися з дев'ятьмаіонамикальцію і утворити молекулу Познера. Кожен Познер можеволодітишістьмазагальними Синглет з іншими Познер - так утворюютьсяцілі хмари заплутаних молекул Познера.Один потік Познер можепотрапити в один нейрон, в той час як іншийпотік - в інший нейрон. Познер можуть бути перенесені через клітиннімембранибілком VGLUT (BNPI). Так два нейрона виявляютьсятакожзаплутаними. Уявіть два Познера, P і Q, щозближуються в нейроні N. Обчисленняквантовоїхіміїпоказують, щоці Познер можутьоб'єднатися один з одним. Припустимо, P бувзаплутаний з Познером P

'в нейроні N'. Якщо P і Q об'єдналися в нейроні N, заплутаністьміж P і P 'дозволить збільшитиймовірністьоб'єднання P' і Q '. Об'єднані Познер будутьпересуватисяповільно - їмдоведетьсядолатиопір води. Водень і магнійможутьзаміщатикальцій в Познер, розбиваючимолекули. Фосфати з негативним зарядом будутьпритягувати позитивно заряджені і, точно так же, як фосфатипритягують. Звільненийкальцій буде заповнюватинейрони N і N '. Підвищенняконцентраціїкальціюпризводить до виникненняхімічногопотенціалу на аксоні і вивільненнюнейромедіаторів, щопередають сигнал міждвома нейронами. Якщо два нейрона N і N 'виявляютьсязаплутаними через молекули Познера, два нейрона можутьзапалитисяодночасно.Квантовісистемиобробляютьінформаціюінакше, ніжкласичнісистеми. Як швидкозможуть Познер оброблятиквантовуінформацію?[26]

3.2 Розробка структури програми

Почнемо з того, що квантові обчислення - це нова дуже модна тема, яка там у них розвивається семимильними кроками по декількох напрямках (а у нас, як будь-яка фундаментальна наука перебуває в запустінні і віддана на відкуп кільком вченим, сидячим в своїх вежах зі слонової кістки). І ось вже говорять про появу перших квантових комп'ютерів (D-Wave, але це не універсальний квантовий комп'ютер), щорічно публікуються нові квантові алгоритми, створюються мови квантового програмування, похмурий геній Міжнародних Ділових Машин в таємних підземних лабораторіях виробляє квантові обчислення на десятках кубітів. Що ж це таке? Квантові обчислення - це обчислювальна модель, яка відрізняється від моделі Тьюринга і фон Неймана, і передбачається, що для деяких завдань вона є більш ефективною. Принаймні знайдені завдання, для яких модель квантових обчислень дає поліноміальних складність, в той час як для класичної обчислювальної моделі невідомо алгоритмів, які мали б складність, нижче експоненційної (але, з

іншого боку, поки що не доведено, що таких алгоритмів не існує). Як таке може бути? Все просто. Квантова обчислювальна модель заснована на кількох досить простих правилах перетворення вхідної інформації, які забезпечують масову параллелизация обчислювальних процесів. Іншими словами, можна одночасно обчислити значення функції для всіх її аргументів (і це буде єдиний виклик функції). Це досягається спеціальною підготовкою вхідних параметрів і спеціальним же видом функції. СветітелямПроц вчить, що все це суть синтаксична маніпуляція з математичними символами, за якою, по суті, немає ніякого сенсу. Є формальна система з правилами перетворення входу у вихід, і ця система дозволяє за допомогою послідовного застосування цих правил отримувати з вхідних даних вихідні. Все це в кінцевому підсумку зводиться до перемножування матриці і вектора. Так ТакТак. Вся модель квантових обчислень заснована на одній простій операції - множенні матриці на вектор, в результаті чого на виході виходить інший вектор. СветітелямХалікаарн на противагу вчить, що існує об'єктивний фізичний процес, який виконує зазначену операцію, і тільки лише існування якого і обумовлює можливість масової паралелізації обчислень функції. Те, що ми сприймаємо це як множення матриці на вектор, є всього лише способом нашого далеко не досконалого відображення об'єктивної реальності в нашому розумі. Ми в нашій науковій лабораторії імені светітелямПроца і Халікаарна об'єднуємо ці два підходи і говоримо, що модель квантових обчислень є математична абстракція, яка відображає об'єктивний процес. Зокрема, числа в векторах і матрицях є комплексними, хоча це абсолютно не збільшує обчислювальну потужність моделі (вона б була такою ж потужною і з дійсними числами), проте обрані саме комплексні числа тому, що знайдено об'єктивний фізичний процес, який здійснює такі перетворення, як описує модель, і в якому використовуються саме комплексні числа. Цей процес називається унітарною еволюцією квантової системи. В основі квантової обчислювальної моделі лежить поняття кубіта. Це практично те ж саме, що і біт в класичній теорії інформації, однак кубіт може одночасно приймати декілька значень. Кажуть, що кубіт

знаходиться в суперпозиції своїх станів, тобто значення кубіта є лінійна комбінація його базових станів, і коефіцієнти при базових станах якраз є комплексними числами. Базовими ж станами є відомі по класичній теорії інформації значення 0 і 1 (в квантових обчисленнях їх прийнято позначати | 0> і | 1>). Ще не дуже-то й зрозуміло, в чому фішка. А фішка ось в чому. Суперпозиція одного кубіта записується як A | 0> + B | 1>, де A і B - деякі комплексні числа, єдине обмеження на які полягає в тому, що сума квадратів їх модулів завжди повинна дорівнювати 1. А якщо розглянути два кубіта? Два біта можуть отримувати 4 можливих значення: 00, 01, 10 і 11. Резонно припустити, що два кубіта є суперпозицію чотирьох базових значень: A | 00> + B | 01> + C | 10> + D | 11>. І так воно і є. Три кубіта є суперпозицію восьми базових значень. Іншими словами, квантовий регістр з N кубітів одночасно зберігає в собі 2N комплексних чисел. Ну а з математичної точки зору це є 2N-мірний вектор в комплексно значною просторі. Саме цим досягається експоненціальна потужність моделі квантових обчислень. Далі функція, яка застосовується до вхідних даних. Оскільки тепер вхідні дані представляють собою суперпозицію всіх можливих значень вхідного аргументу, функція повинна бути перетворена в такий вигляд, щоб прийняти таку суперпозицію і обробити її. Тут теж все більш-менш просто. В рамках моделі квантових обчислень кожна функція являє собою матрицю, на яку накладі.

3.3 Розробкапрограмиемуляціїквантовихлогічнихфункцій та розрахунків

Незважаючи на всі зусилля, на сьогоднішній день квантові комп'ютери дозволяють вирішувати не так багато практичних завдань питань, але потенціал виглядає вражаюче. Зараз розвиток квантових обчислень йде за двома напрямками: Спеціалізовані квантові комп'ютери, які спрямовані на вирішення однієї конкретної специфічної завдання, наприклад, завдання оптимізації. Прикладом продукту є квантові комп'ютери D-Wave. Універсальні квантові комп'ютери - які здатні реалізувати довільні квантові алгоритми. На

сьогоднішній день існують тільки невеликі прототипи універсальних квантових комп'ютерів - в цьому напрямку працюють Google, IBM і Intel. Вони закладають основу, але поки не дозволяють робити щось масштабне і не вміють справлятися з помилками. У будь-якому випадку квантові комп'ютери дозволяють оперувати великим простором станів і це може бути корисно, наприклад, для вирішення завдань пошуку, оптимізації різних процесів і моделювання складних систем. Завдяки тому, що IBM пропонує всім бажаючим скористатися квантовим комп'ютером, сучасні квантові програмісти вже тренуються в збірці завдань і запуску їх на невеликих квантових комп'ютерах. Наприклад, для пошуку по невпорядкованою базі даних квантовий алгоритм має квадратичне перевага. У такому завданні невпорядкована база даних може бути представлена як деякий "чорний ящик", на вхід якого подаються запити (адреси елементів в цій базі даних), а чорний ящик на них відповідає "так" або "ні" (чи підходить елемент, розташований по даною адресою, вимогам запиту). Уявімо, що в деякій базі даних адреса кожного з елементів складається з n біт, і в цій базі даних є лише один елемент, що задовольняє деяким умовам. Щоб знайти цей елемент нам в середньому потрібно близько 2 ^ n запитів (точніше 2 ^ (n-1)), тому що через невпорядкованості бази даних все, що нам залишається це послідовно перебирати всі можливі адреси (яких 2 ^ n штук) до тих пір поки нам врешті не пощастить і ми не потрапимо на потрібний елемент. У разі ж, якщо у нас є квантовий аналог подібного чорного ящика (його ще називають «квантовим оракулом») для того, щоб отримати відповідь нам буде потрібно близько 2 ^ (n / 2) запитів. Перевага «алгоритму квантового перебору», названого на честь Л. Гровера, обумовлено можливістю задавати безліч питань квантовому ящику одночасно - формувати суперпозицію запитів. Важливо відзначити, що завдання пошуку в невпорядкованою базі даних носить універсальний характер - до неї можна звести практично будь-яку іншу задачу (в тому числі і NP-повну). Однак для її вирішення потрібно кількість запитів, зростаюче експоненціально з ростом складності завдання (в розглянутому прикладі, їй відповідав параметр n). Таким чином, не варто ставитися до

квантовому комп'ютера як до всемогутнього інструменту, здатного вирішувати довільні обчислювальні завдання з експоненціальним прискоренням. У ряді випадків його можливості будуть значно скромніші. Проте, великий потенціал вже сьогодні очевидний для задач зі сфери квантової хімії. Наприклад, в промисловості затребуваний розрахунок параметрів хімічних сполук і моделюванні хімічних реакцій. При використанні класичних комп'ютерів, нам не вистачає можливостей і доводиться, найчастіше, йти на компроміс з точністю. Квантові комп'ютери можуть допомогти детально визначити ланцюжка реакцій, динаміку процесів, знайти каталізатори для потрібних реакцій - все це дуже корисно! Одна з найбільш обговорюваних сьогодні завдань - виробництво аміаку. Це з'єднання активно використовується в добривах для рослин, і на його виробництво витрачається 1-2% всієї енергії на землі (дані QuantumComputingReport і BP). Якби за допомогою квантового комп'ютера можна було б оптимізувати процес виробництва аміаку за рахунок точного знаннях всіх параметрів, то він вже окупив би все вкладення, які здійснені в розробки технологій (пам'ятаєте, 1-2% світової енергії). Нещодавно на стику квантової фізики і машинного навчання виник новий напрям - квантове машинне навчання або, як часто говорять, Quantum AI. При цьому важливо, що перевага квантового комп'ютера над класичними в задачах машинного навчання не вимагає повноцінного і многокубітного квантового комп'ютера. За допомогою квантового комп'ютера, наприклад, можна буде прискорити окремі елементи алгоритмів машинного навчання, а також прискорити процес їх навчання. В Google останні роки квантове машинне навчання вважається одним з топових напрямків всієї сфери квантових технологій.Для наступного прориву, однак, потрібно не тільки залізо, але і нові швидкі квантові алгоритми. Тут є помітний прогрес. Наприклад, для вивчення з'єднання Fe2S2 за допомогою алгоритмів квантової хімії раніше було потрібно тридцять років при аналізі на квантовому комп'ютері. За рахунок пошуку більш оптимального алгоритму, цей час скоротився до 2 хвилин з урахуванням використання того ж заліза. Однак квантових алгоритмів поки все ще

недостатньо. Поки їх все ще лише кілька десятків, а для повноцінного розвитку сфери квантових обчислень, алгоритмів має стати набагато більше. Квантової комп'ютер має дві сторони: темну і світлу. До сих пір ми говорили про світлій стороні - вирішенні їм практично затребуваних завдань, які не можуть бути вирішені за допомогою класичних комп'ютерів. Але і є і темна сторона: квантовий комп'ютер набагато краще класичного вирішує завдання факторизації. Складність цього завдання, як відомо, є однією з основ забезпечення стійкості поширених алгоритмів криптографії з відкритим ключем. Завдання факторизации надзвичайно складна для класичного комп'ютера, а на квантовому може бути ефективно вирішена за допомогою алгоритму Шора. Наприклад, злом RSA-ключа, що складається з 1024 біт, займе мільйони років безперервних обчислень на класичних комп'ютерах, тоді як на квантовому комп'ютері ця задача буде вирішена за 10 годин (якщо припустити, що кожна квантова операція виконується 10 нс і що в розпорядженні є комп'ютер з достатньої кількості логічних кубітів). Поки квантові комп'ютери не дозволяють нічого зламати - адже для криптоаналізу RSA потрібно кілька тисяч керованих кубітів. І хоча потенційно небезпечного комп'ютера ще не існує, спільнота вже сьогодні замислюється про захист від можливих проблем в майбутньому. Один варіант рішення - використання технології квантової розподіл ключів, яка дозволяє двом сторонам обмінюватися криптографічними ключами для симетричного шифрування. Як відомо, одиночний фотон не можна розділити, а квантовий стан не можна скопіювати - це фундаментальне обмеження квантової механіки. На такому принципі - захист переданих даних фундаментальними фізичними законами - будуються нові прилади. У цій сфері на світовій арені лідирує Китай. У Росії технологія квантового розподілу ключів розвиваються декількома групами, наприклад, в РКЦ, МДУ ім. М.В. Ломоносова і ИТМО. Прилад, розроблений в РКЦ вже проходив випробування в Ощадбанку і Газпромбанку. За рівнем помилок в каналі можна дізнатися, чи була можливість компрометації ключа. Якщо рівень помилок нижче критичного порога, то можна виправити помилки і

виключити з нього потенційно доступну зловмисникові інформацію за допомогою класичних алгоритмів і, таким чином, згенерувати фінальний секретний ключ. При цьому інформація, що захищається залишається недоступною зловмисникові. Центральна ідея полягає в тому, щоб використовувати квантово-розподілені ключі в шифрі Вернама - одноразовому блокноті. Наскільки відомо, в самих критично важливих системах Китаю реалізується саме така система. Другий принцип захисту - це постквантовая криптографія. Вона включає в себе новий клас алгоритмів з відкритим ключем, які засновані на завданнях, які є обчислювально складними як для класичного комп'ютера, так і для квантового. Багатьох цікавить питання про те, зашкодить квантовий комп'ютер блокчейну. Та це можливо. За рахунок атак на цифрові підписи, а також за рахунок використання квантового алгоритму Шора і впливу на алгоритми консенсусу з боку квантового алгоритму Гровера. Однак і захистити блокчейни можна також квантовим розподілом ключів або постквантовой криптографією.Візнер також припустив, що аналогічний механізм можна використовувати для створення каналів конфіденційного зв'язку. Уже через рік після виходу його статті вчені ЖильБрассар і Чарльз Беннет розробили перший протокол для квантової зв'язку, який вони назвали за першими літерами своїх прізвищ і року створення технології - BB84. Саме цей протокол широко застосовується в сучасних квантових мережах зв'язку. Беннет і Брассар запропонували кодувати дані в квантових станах одиночних фотонів, наприклад, в їх поляризації. Як і у випадку з іншими квантовими об'єктами, сам факт вимірювання обов'язково впливає на стан об'єкта, отже, якщо хтось третій спробує "підслухати" передачу фотонів - тобто виміряти стану фотонів, якими ми обмінюємося, ми обов'язково це зауважимо, тому що зміняться стану фотонів. Тому в теорії непомітно підключитися до каналу квантової передачі даних неможливо в принципі - не дозволяють фундаментальні закони квантової механіки (на практиці і у цій технології є деякі вразливості, але про це нижче). Протокол BB84 працює наступним чином. Один зі співрозмовників (традиційно його називають Алісою) посилає іншому (Бобу) фотони, поляризовані в одному

з двох, неортогональних один одному, базисах: прямокутному або діагональному. Боб отримує їх і вимірює поляризацію, вибираючи базиси для вимірювання випадковим чином, і записує результати вимірювань і базиси. Потім він і Аліса обмінюються інформацією про використані базисах (але не про результати вимірювання) по відкритому каналу, і дані, отримані при несовпавших базисах, скидаються. Залишаються тільки значення, виміряні в співпадаючих базисах (в технології квантового розподілу ключів це називається "просіюванням ключа"). Можливий "шпигун", який підслуховує передачу даних по цій лінії зв'язку (його зазвичай називають Єва) може перехопити одиночний фотон, виміряти його поляризацію і спробувати переслати копію фотона Бобу. Але, відповідно до теореми про неможливість клонування довільного квантового стану, це призведе до зростання числа помилок в розподіляється квантовому ключі. В результаті і Аліса, і Боб зрозуміють, що їх канал прослуховує сторонній. Для визначення рівня помилок в ключі після процедури квантового розподілу Аліса і Боб по відкритому каналу порівнюють невелику частину ключа. Вважається, що якщо рівень помилок в ключі менше 11 відсотків, то можна гарантувати безпеку лінії зв'язку. Перший експеримент по передачі інформації по квантовому каналу Беннет і Брассар провели в кінці жовтня 1989 року. Їм не щастило - їх ідею не сприйняли всерйоз, тому вчені вирішили створити прототип експериментальної установки самостійно і на свої власні гроші. Реалізувати установку допомагали друзі. Перша установка для абсолютно захищеною квантової зв'язку передавала дані на дистанцію 32,5 сантиметра. Брассар згадує, що їх система забезпечувала захист даних тільки від людини, який виявився б абсолютно глухим: блок живлення дуже сильно шумів, причому шум був різним у залежності від того, яку поляризацію фотонів установка забезпечувала в даний момент. Незважаючи на всі недоліки, установка була робочою. Власне, з цього моменту і почалася історія квантових комунікацій і квантових мереж, які сьогодні розтягуються на тисячі кілометрів і виходять в космос.Без шифрування сьогодні практично ніхто не передає даних. Найпопулярніші методи шифрування, які використовуються зараз, засновані на

одному допущенні: завдання дешифрування повідомлень настільки складна, що обчислювальних потужностей зловмисника не вистачить, щоб її вирішити. Інакше кажучи, вартість (і в грошах і в часі) дешифрування виявиться незрівнянно вищою, ніж цінність отриманої таким чином інформації. Це стосується як симетричного шифрування (AES, DES, російського ГОСТ 28147-89), так і асиметричного (наприклад RSA). Таки чи безпечна квантовий зв'язок? На даний момент вона повністю безпечна, але ситуація незабаром може змінитися через появу квантового комп'ютера. Справа в тому, що в системах шифрування з відкритим ключем використовуються так звані односторонні функції, в яких за відомим аргументу знайти значення функції досить просто, а от зворотна операція вкрай складна. Наприклад, множення навіть дуже великих чисел - просте завдання для комп'ютера, а ось зворотна - розкладання на множники (факторизація) - вимагає багато разів більше обчислювального часу, ніж для вирішення вихідної задачі, причому складність цього завдання швидко зростає в міру збільшення числа. На використанні асиметрії множення і факторизации заснований, наприклад, широко поширений алгоритм шифрування RSA, і багато інших систем шифрування, які називаються "асиметричними". Їх головна перевага полягає в тому, що для їх використання не потрібно передавати ключі шифрування за спеціальним захищеному каналу (наприклад, флешкою з довіреною кур'єром), як у випадку з симетричними алгоритмами, де один і той же секретний ключ використовується і для шифрування і дешифрування. В асиметричних технологіях використовується два ключі - відкритий і закритий, перший можна передавати по мережах, і його можна використовувати тільки для того, щоб зашифрувати повідомлення, а для розшифровки потрібен закритий ключ, який зберігається у користувача. Закритий і відкритий ключ пов'язані між собою асиметричною функцією, і як вважається, відновити з відкритого ключа закритий за допомогою сучасних технологій практично неможливо (на це можуть знадобитися мільярди років). Але це зараз, в майбутньому ситуація може змінитися, якщо з'являться квантові комп'ютери. Ще в середині 1990-х років математик Пітер Шор розробив

квантовий алгоритм, який отримав його ім'я. Алгоритм дозволяє здійснювати факторизацию майже так само швидко, як множення. Квантові пристрої, на яких можна запустити алгоритм Шора, вже існують, але поки вони успішно факторізовано лише числа 15 і 21. З появою більш просунутих квантових машин все криптосистеми, засновані на цій асиметрії, стануть марними. Деякі вчені називають квантовий комп'ютер "інформаційної атомною бомбою", через яку доведеться прибрати більшу частину звичних нам сьогодні інформаційних і банківських сервісів: близько 50% інтернет-трафіку цих сервісів закодована алгоритмами з відкритим ключем. Причому той факт, що квантовий комп'ютер не створений зараз, не означає, що дані, якими ви обмінюєтеся зараз, в безпеки - можливо, вони будуть розшифровані в майбутньому. Наприклад, американське розвідувальне агентство NSA в своєму дата-центрі в Юті зберігає як мінімум кілька ексабайт нерозшифрованих даних. Як тільки з'являться нові методи дешифрування, вони можуть бути розшифровані. Але квантова ж фізика дає нам і захист від обчислювальних можливостей і квантового і майбутніх класичних комп'ютерів і обчислювальних алгоритмів - квантове розподіл ключів.

3.4 Оцінювання якості роботи запропонованого способу

Щоб оцінити якість запропонованого методу емуляції квантових логічних функцій і розрахунків за допомогою мов програмування високого рівня застосовувалась мануальна оцінка та рекомендації із [28]. Для запропонованого підходу методу емуляції квантових логічних функцій і розрахунківвиконувалось тестування виходячи з результатів проведених досліджень. Для виконання операції тестування було використано рекомендації. Тестування роботи створеногопродукту було виконано для різних логічних функцій і обрахунків.

Розроблена програма тестувалася на різних системах з різною конфігурацією. Кількість вірних результатів визначалася виходячи з відношення кількості вірних результатів емуляції квантових логічних функцій і

розрахунків до загальної кількості виконаних спроб. Перевірка функціонування програми була виконана при різних конфігураціях вхідних данних. Запропонований підхід по емуляції квантових логічних функцій і розрахунків є досить точним, але його точність зростає із зменшенням кількості емульованих частинок.

Результати проведених досліджень показують, що запропонований спосіб емуляції квантових логічних функцій і розрахунків працює з хорошими результатами, які у порівнянні із класичними алгоритмами є дещо кращими.

Підсумовуючиварто відзначити, що запропонований підхід емуляції квантових логічних функцій і розрахунків можна застосовувати у системах виділення та розпізнавання об'єктів.

4 РОЗРАХУНОК ЕКОНОМІЧНОЇ ДОЦІЛЬНОСТІ СТВОРЕННЯ ПРОГРАМИ ФОРМУВАННЯ ОЗНАК ОБ'ЄКТІВ

Виконання будь-якої науково-дослідної роботи завжди вимагає певних витрат. Ці витрати на виробництво та реалізацію продукту, повинні постійно зменшуватись, тому що у цьому полягає прогрес будь-якого виробництва. Якщо цього немає, то ніяка науково-технічна розробка не буде реалізована на практиці, адже така розробка не буде ефективніша ніж існуючі на ринку аналоги [27, 28].

На основі економічних розрахунків, можна довести економічну доцільність та ефективність впровадження результатів, що були отримані в результаті виконаних науково-дослідних робіт у виробництві.

4.1 Технологічний аудит розробки

Результатом магістерської кваліфікаційної роботи «Емуляція квантових логічних функцій і розрахунків за допомогою мов програмування високого рівня» є розробка програмного продукту. Для проведення технологічного аудиту залучено трьох незалежних експертів: Захарченко С.М. (к.т.н., доцент), Гарнага (к.т.н.,), Снігур А.В. (к.т.н., доцент). Оцінювання комерційного потенціалу здійснене за критеріями, що наведені в таблиці 4.1 [27][28].

Таблиця 4.1 - Критерії оцінювання комерційного потенціалу розробки бальна оцінка

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри-терій	0	1	2	3	4
Технічна здійсненність концепції:					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги (недоліки):					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
	Продовження таблиці 4.1				

3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат Аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витрачати значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування

Закінчення таблиці 4.1

10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років

12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів	Необхідно отримання великої кількості дозвільних документів, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання комерційного потенціалу експертами розробки зведено в таблицю 4.2.

Таблиця 4.2 - Результати оцінювання комерційного потенціалу розробки

Критерії	Прізвище, ініціали, посада експерта		
	1 – Захарченко	2 – Гарнага	3 – Снигур
	Бали, виставлені експертами:		
1	4	3	4
Ринкові переваги (недоліки):			
2	4	3	3
3	3	4	3
4	3	3	4
5	3	4	3
Ринкові перспективи			
6	3	4	3
7	4	3	3
Практична здійсненність			
8	4	3	3
9	3	4	3
10	3	3	4
11	3	3	3
12	3	3	3
Сума балів	$СБ_1$=40	$СБ_2$=40	$СБ_3$=40
Середньоарифметична сума балів $\overline{СБ}$	40		

За даними таблиці 4.2 можна зробити висновок, щодо рівня комерційного потенціалу розробки. Зважимо на результат й порівняємо його з рівнями комерційного потенціалу розробки, що представлено в таблиці 4.3.

Таблиця 4.3 – Рівні комерційного потенціалу розробки

Середньоарифметична сума балів $\overline{СБ}$, розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0 – 10	Низький
11 – 20	Нижче середнього
21 – 30	Середній
31 – 40	Вище середнього
41 – 48	Високий

Рівень комерційного потенціалу розробки, становить 40 балів, що відповідає рівню «Вище середнього».

В першому розділі магістерської роботи було розглянуто існуючі аналоги на ринку.

До якісних характеристик нової розробки можна віднести: функціональність, надійність, зручність використання, простоту використання (адаптованість, зручність установки, здатність до співіснування).

Продукт, який пропонується, є модифікацією продуктів, що вже існують на ринку. Потреби користувачів, які повинен задовольнити продукт – зручність у використанні, цілодобовий режим роботи, достатній ресурс роботи без необхідності додаткового обслуговування, швидкість обслуговування.

В даній розробці зацікавлені науковці в сфері компютерної електроніки.

Концепція підтверджена експертними висновками, програмний продукт знаходиться в розробці, є перші результати Q# та OpenQASM

Розробка технічно готова на 70%. Вона охоплює виконання всього обсягу робіт з проектування, також дає можливість впровадження нових та вдосконалення готових складових ресурсу. Потребує фінансових та трудових ресурсів, а також фахівців відповідної кваліфікації.

Комерціалізація розробки знаходиться на початковому етапі, розпочато процес виведення продукту на інформаційний ринок, проводиться підготовка презентації для інвесторів.

Розповсюдження інформації про нову розробку планується через технічні форуми, платформу GitHub та університетські форуми.

4.2 Прогнозуваннявитрат на виконання та впровадженнярезультатівнауковоїроботи

Прогнозування витрат на виконання науково-дослідної, дослідно-конструкторської та конструкторсько-технологічної робіт складається з таких етапів:

1) розрахунок витрат, які безпосередньо стосуються виконавців даного розділу роботи;

2) розрахунок загальних витрат на виконання даної роботи;

3) прогнозування загальних витрат на виконання та впровадження результатів даної роботи.

Виконаєморозрахунок витрат, які безпосередньо стосуються виконавця даного розділу роботи, приймаючи до уваги те, що для розробки програми було залучено одного розробника.

Основна заробітна плата розробника (дослідника) $З_о$:

$$З_о = \frac{М}{Т_р} \cdot t \, [грн], \qquad (4.1)$$

де М – місячний посадовий оклад розробника, 5000,00 грн.

$Т_р$ – число робочих днів в місяці; приблизно $Т_р$ = (22) дні;

t – число робочих днів роботи розробника (дослідника) – 66 днів.

$$З_о = \frac{5000}{22} \cdot 66 = 15000{,}00 \text{ (грн)}.$$

Додаткова заробітна плата $З_д$ розробника, розраховується як 10 % від суми основної заробітної, тобто:

$$З_д = (0{,}1...0{,}12) \cdot З_о \, [грн]. \qquad (4.2)$$

$$З_д = 0{,}10 \cdot 1500 = 1500{,}00 \text{ (грн)}.$$

Нарахування на заробітну плату $Н_{зп}$ розробника становлять 22 % і розраховуються за формулою:

$$Н_{зп} = (З_о + З_д) \cdot \frac{\beta}{100} \, [грн], \qquad (4.3)$$

де $З_о$ – основна заробітна плата розробників, грн;

$З_д$ – додаткова заробітна плата розробника, грн;

β – ставка єдиного соціального внеску, 22%.

$$Н_{зп} = (15000,00 + 1500,00) \cdot 0,22 = 3630,00 \text{ (грн)}.$$

Зроблені розрахунки зведено до таблиці 4.4.

Таблиця 4.4 – Амортизаційні відрахування

Найменування	Балансова вартість, грн	Термін використання, роки	Фактична трив. використання, міс.	Величина амортизаційних відрахувань, грн
Приміщення	200000	25	3	2000,00
Ноутбук	20000	5	3	1000,00
Всього				3000,00

У спрощеному вигляді амортизаційні відрахування розраховуємо за формулою:

$$А = (Ц \cdot Т)/(12 \cdot Тв) \text{ [грн]}, \quad (4.4)$$

де Ц – загальна балансова вартість обладнання, приміщення тощо, грн;

Т – фактична тривалість використання, міс;

$Т_В$ – термін використання обладнання, приміщень тощо, роки.

Під час розробки програмного продукту використовувались лише безкоштовні програмні засоби.

Крім того, за три місяці було сплачено 450 грн на послуги Інтернет (150 грн/міс).

Інші витрати $І_в$, як 50% від суми основної заробітної плати розробника тобто:

$$В_{ін} = 50\% \cdot З_о \textit{ [грн]}. \quad (4.5)$$

$$В_{ін} = 0,5 \cdot 15000,00 = 750,00 \text{ (грн).}$$

Сума всіх попередніх статей витрат дає витрати на виконання даної частини (розділу, етапу) роботи – В:

$$В = 15000+1500,00+ 3630,00+3000,00+450,00+750,00 = 24330,00 \text{ (грн).}$$

Загальна вартість всієї наукової роботи ($В_{заг}$) визначається за формулою:

$$B_{заг} = B/\alpha \text{ [грн],} \qquad (4.6)$$

де α – частка витрат, які безпосередньо здійснює виконавець даного етапу роботи, у відносних одиницях.

$$В_{заг} = 24330,00 /1 = 24330,00 \text{ (грн).}$$

Проведемо прогнозування загальних витрат ЗВ на виконання та впровадження результатів виконаної наукової роботи за формулою:

$$ЗВ = \frac{В_{заг}}{\beta} \text{ [грн],} \qquad (4.7)$$

де β – коефіцієнт, який характеризує етап (стадію) виконання даної роботи. Так, як розробка знаходиться на дослідного зразка, то $\beta \approx 0,9$.

$В_{заг}$ - загальна вартість всієї наукової роботи – 24330,00 грн.

$$ЗВ = \frac{24330}{0,7} = 34757,14 \text{ (грн).}$$

Отже, прогноз загальних витрат на виконання та впровадження результатів становить 34757,14 грн.

4.3 Прогнозуваннякомерційнихефективвідреалізаціїрезультатіврозробки

Спробуємо кількісно спрогнозувати вигоду яку можна отримати у майбутньому від впровадження результатів виконаної наукової роботи. Зрозуміло, що всі зроблені тут розрахунки будуть приблизними і не передбачають деталізації.

В умовах ринку, узагальнюючим позитивним результатом, який планує отримати підприємець від впровадження результатів нової розробки, є збільшення чистого прибутку.

Виконання даної наукової роботи та впровадження її результатів складає приблизно 3 місяці. Позитивні результати від впровадження розробки очікуються вже в перший рік впровадження.

Проведемо прогнозування позитивних результатів та кількісне їх оцінювання по роках.

Обчислимо збільшення чистого прибутку підприємства $\Delta П_i$ для кожного із років, протягом яких очікується отримання позитивних результатів від впровадження розробки, розраховується за формулою:

$$\Delta П_i = \sum_1^n (\Delta П_я \cdot N + П_я \cdot \Delta N)_i \ [грн], \qquad (4.8)$$

де $\Delta П_я$ – покращення основного якісного показника від впровадження результатів розробки у даному році;

N – основний кількісний показник, який визначає діяльність підприємства у даному році до впровадження результатів наукової розробки;

ΔN – покращення основного кількісного показника діяльності підприємства від впровадження результатів розробки;

$П_я$– основний якісний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки;

n – кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки.

Припустимо, що внаслідок впровадження результатів наукової розробки чистий прибуток збільшиться на 200 грн., а кількість оплачених рекламних послуг збільшиться: протягом першого року – на 100 од., протягом другого року – ще на 500 од., протягом третього року – ще на 700 од.

Орієнтовно: реалізація реклами до впровадження результатів наукової розробки складала 1 шт., а прибуток, що його отримувало підприємство на одиницю продукції до впровадження результатів наукової розробки – 100 грн.

Потрібно спрогнозувати збільшення чистого прибутку підприємства від впровадження результатів наукової розробки у кожному році відносно базового.

Збільшення чистого прибутку підприємства $\Delta П_1$ протягом першого року складе:

$$\Delta П_1=200\cdot1+(100+200)\cdot100=30200{,}00 \text{ (грн)}.$$

Обчислимо збільшення чистого прибутку підприємства $\Delta П_2$ протягом другого року:

$$\Delta П_2=200\cdot1+(100+200)\cdot(100+500)=180200{,}00 \text{ (грн)}.$$

Збільшення чистого прибутку підприємства $\Delta П_3$ протягом третього року становитиме:

$$\Delta П_3=200\cdot1+(100+200)\cdot(100+500+700)=390200{,}00 \text{ (грн)}.$$

Отже, комерційний ефект від впровадження розробки виражається у щорічному збільшенні чистого прибутку підприємства протягом трьох років.

4.4 Розрахунок ефективності вкладених інвестицій та періоду їх окупності.

Щоб оцінити доцільність фінансування проекту, необхідно провести розрахунки ефективності вкладених інвестицій.

Основними показниками є абсолютна і відносна ефективність вкладених інвестицій та термін їх окупності.

На першому етапі розрахуємо теперішню вартість інвестицій PV, що вкладаються в наукову розробку.

Такою вартістю ми можемо вважати прогнозовану величину загальних витрат ЗВ на виконання та впровадження результатів НДДКР. Будемо вважати, що ЗВ = PV=34757,14 (грн).

На другому етапі розраховуємо очікуване збільшення прибутку $\Delta П_i$, що його отримає підприємство від впровадження результатів наукової розробки, для кожного із років, починаючи з першого року впровадження. Таке збільшення прибутку було розраховане раніше.

Результати вкладених у наукову розробку інвестицій виявляться за перший рік після провадження у тому, що у першому році підприємство отримає збільшення чистого прибутку на 30200,00 грн відносно базового року, у другому році – збільшення чистого прибутку на 180200,00 грн (відносно базового року), у третьому році – збільшення чистого прибутку на 390200,00 грн (відносно базового року).

На третьому етапі для спрощення подальших розрахунків будуємо вісь часу, на яку наносимо всі платежі (інвестиції та прибутки), що мають місце під час виконання науково-дослідної роботи та впровадження її результатів.

Платежі показуються у ті терміни, коли вони здійснюються.

Рисунок, що характеризує рух платежів (інвестицій та додаткових прибутків) буде мати вигляд, наведений на рис. 4.1.

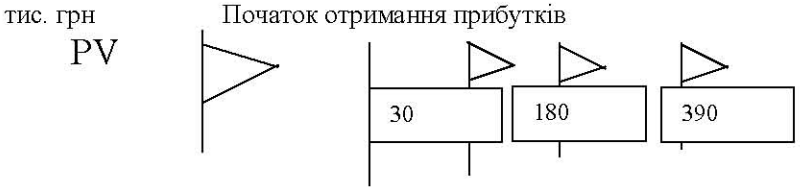

Рисунок 4.1 – Вісь часу з фіксацією платежів, що мають місце під час розробки та впровадження результатів НДДКР

На четвертому етапі розраховуємо абсолютну ефективність вкладених інвестицій $E_{абс}$ за формулою:

$$E_{абс} = (ПП - PV)\ [грн], \qquad (4.9)$$

де ПП – приведена вартість всіх чистих прибутків, що їх отримає підприємство (організація) від реалізації результатів наукової розробки, грн;

PV – теперішня вартість інвестицій PV = ЗВ, грн.

Приведена вартість всіх чистих прибутків ПП розраховується за формулою:

$$ПП = \sum_{1}^{\tau} \frac{\Delta П_i}{(1+\tau)^t}\ [грн], \qquad (4.10)$$

де $\Delta П_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн;

т – період часу, протягом якого виявляються результати впровадженої НДДКР, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні - 0,1;

t – період часу (в роках) від моменту отримання чистого прибутку до точки „0”;

$$ПП = \frac{30200}{(1+0{,}1)^1} + \frac{180200}{(1+0{,}1)^2} + \frac{390200}{(1+0{,}1)^3} = 469543{,}20\ (грн).$$

$$Е_{абс} = 469543{,}20 - 34757{,}14 = 434786{,}06\ (грн).$$

Оскільки $Е_{абс} > 0$, результат від проведення наукових досліджень щодо розробки програмного продукту та їх впровадження принесе прибуток, тобто є доцільним, але це ще не свідчить про те, що інвестор буде зацікавлений у фінансуванні даної програми.

На п'ятому етапірозраховують відносну (щорічну) ефективність вкладених в наукову розробку інвестицій $Е_в$ за формулою:

$$E_в = \sqrt[T_ж]{1 + \frac{E_{абс}}{PV}} - 1, \qquad (4.11)$$

де $Е_{абс}$ – абсолютна ефективність вкладених інвестицій, грн;

PV – теперішня вартість інвестицій PV = ЗВ, грн;

$Т_ж$ – життєвий цикл наукової розробки, роки.

$$E_В = \sqrt[3]{1 + \frac{434786{,}06}{34757{,}14}} - 1 = \sqrt[3]{13{,}5} - 1 = 1{,}38 \text{ або } 138\%$$

Порівняємо $Е_в$ з мінімальною (бар'єрною) ставкою дисконтування $\tau_{мін}$, яка визначає ту мінімальну дохідність, нижче за яку інвестиції вкладатися не будуть. Спрогнозуємо величину $\tau_{мін.}$ У загальному вигляді мінімальна (бар'єрна) ставка дисконтування $\tau_{мін}$ визначається за формулою:

$$\tau = d + f, \qquad (4.12)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; d = 0,2;

f – показник, що характеризує ризикованість вкладень; величина f = 0,1.

$$\tau = 0{,}2 + 0{,}1 = 0{,}3$$

Оскільки $E_в$ = 138% $> \tau_{мін}$ = 30%, то у інвестора є потенційна зацікавленість у фінансуванні даної наукової розробки.

На шостому етапірозраховуємо термін окупності вкладених у реалізацію наукового проекту інвестицій $T_{ок}$ за формулою:

$$T_{ок} = \frac{1}{E_в} \text{ [року].} \quad (4.13)$$

$$T_{ОК} = \frac{1}{1{,}38} = 0{,}7 \text{ (року).}$$

Оскільки термін окупності вкладених у реалізацію наукового проекту інвестицій менше трьох років ($T_{ок} < 3$ років), то фінансування нової розробки є доцільним.

4.5 Висновкиекономічногогрунтування

В даному розділі було здійснено оцінювання комерційного потенціалу розробки. Проведено технологічний аудит з залученням трьох експертів.

Аналіз експертних даних показав, що рівень комерційного потенціалу розробки є високим. Дослідження комерційного потенціалу розробки показав, що програмний продукт за своїми характеристиками випереджає аналогічні програмні продукти, що робить його конкурентоспроможним на ринку. Існуючі переваги нової розробки дозволять швидко її поширити на ринку.

Згідно із розрахунками всіх статей витрат на виконання науково-дослідної, дослідно-конструкторської та конструкторсько-технологічної роботи загальна вартість витрат на розробку і впровадження складає 34757,14 грн.

Абсолютна ефективність вкладених інвестицій в сумі 434786,06грн свідчить про отримання прибутку інвестором від впровадження програмного продукту.

Щорічна ефективність вкладених в наукову розробку інвестицій складає 138 %, що набагато вище за мінімальну бар'єрну ставку дисконтування, яка складає 30%. Це означає потенційну зацікавленість інвесторів у фінансуванні розробки.

Термін окупності складає 0,7 року, що також свідчить про доцільність фінансування.

ВИСНОВКИ

Головною цілю магістерській кваліфікаційні роботі була розробка програмних засобів для емуляції квантових логічних функцій і розрахунків з покращеними показниками ефективності обрахунків і зручності використання в навчальних цілях. Для цього було проведено такі етапи досліджень:

1) У магістерській роботі був проведений аналіз сучасного стану галузі і проведено пояснення базових концепцій які визначають розвиток квантових обчислень на сьогоднішній день.

2) У магістерській роботі вдосконалено метод поширення внутрішнього страну внутрішнього стану віртуальних кубітів за рахунок впровадження ряду прогресивних рішень, які якісніше емулюють квантову заплутаність.

3) У магістерській роботі розроблено алгоритм емуляції квантових розрахунків з використанням вдосконаленого підходу поширення внутрішнього стану віртуальних кубітів, проедставлені зразки використання базових логічних функцій.

4) У магістерській роботі виконані економічні розрахунки із обґрунтуванням доцільності виконання нової розробки по емуляції квантових логічних функцій і розрахунків, обраховані фінансові затрати на виготовлення програмного продукту та визначено комерційні, економічні переваги впровадження запропонованого рішення.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Берко С.Г., Гарнага В.А. Емуляція квантових логічних функцій і розрахунків за допомогою мов програмування високого рівня/ В.А. Гарнага,Берко С.Г. // Перспективи впровадження адаптивних технологій передачі мультимедійного контенту каналами радіозв'язку (2019), Україна, Одеса, 7 листопада.: збірник наукових праць.- Одеса, 2019.

2. Языки программирования для квантового компьютера [Електронний ресурс] // Блог компании Mail.ru Group. – 2018. – Режим доступу до ресурсу: https://habr.com/ru/company/mailru/blog/350208/.

3. Квантовые вычисления и язык Q# для начинающих [Електронний ресурс] // Блог компании Microsoft. – 2018. – Режим доступу до ресурсу: https://habr.com/ru/company/microsoft/blog/351622/.

4. Учимся программировать квантовый компьютер [Електронний ресурс] // Tproger. – 2018. – Режим доступу до ресурсу: https://tproger.ru/translations/quantum-computer-programming/.

5. Бернхард К. Квантові обчислення для справжніх айтішників / Кріс Бернхард. – Москва: Пітер Прес, 2018. – 240 с. – (Офсет). – (Бібліотека програміста).

6. Квантовые вычисления против классических: зачем нам столько цифр [Електронний ресурс] // Блог компании Сбербанк. – 2017. – Режим доступу до ресурсу: https://habr.com/ru/company/sberbank/blog/343308/.

7. Введение в квантовые вычисления [Електронний ресурс] // Блог компании Microsoft. – 2018. – Режим доступу до ресурсу: https://habr.com/ru/company/microsoft/blog/351624/.

8. Квантовые вычисления [Електронний ресурс] // Санкт-Петербурзькийдержавнийуніверситет. – 2019. – Режим доступу до ресурсу: https://ru.coursera.org/learn/kvantovyye-vychisleniya.

9. Анненков А. Квантовые вычисления [Електронний ресурс] / Андрій Анненков // КОММЕРСАНТЪ НАУКА. – 2018. – Режим доступу до ресурсу:

https://elementy.ru/nauchno-populyarnaya_biblioteka/435036/Kvantovye_vychisleniya.

10. 10) Digital Signal Processing: World Class Designs / [David Katz, Kenton Williston, Walt Kester and other]. – Newnes, 2009. – 460 p.

11. Test and Measurement / [Stuart Ball, G. M. S. De Silva, Jon Wilson and other]. – Newnes, 2008. – 891 p.

12. Садовничего В. А. Квантовые вычисления: за и против. / ВолодимирАатолійовичСадовничего. – Ізжевск: Удмуртский университет, 1999. – 212 с.

13. Белонучкин В. Е. Квантовые компьютеры: можно ли их сделать "большими" / В. Е. Белонучкин, Д. А. Заикин. – Москва: Іжевськ: Видавничийдім, 2000. – 169 с. – (Успіхифізичних наук).

14. Walt Kester Which ADC architecture is right for yourapplication / Walt Kester // Analog dialogue. – 2005. – Vol. 39, № 2. – P. 11–18.

15. Душкин Р. Кратчайшее введение в квантовые вычисления [Електронний ресурс] / Роман Душкин // Записки програміста. – 2017. – Режим доступу до ресурсу: https://eax.me/quantum-computing-intro/.

16. Ценцура К. Вселенськийкомп'ютер. Як квантоваперевага Google змінить наше життя [Електронний ресурс] / КостянтинЦенцура // НВ. – 2019. – Режим доступу до ресурсу: https://nv.ua/ukr/techno/it-industry/gosti-iz-budushchego-kak-google-sozdal-kvantovyy-kompyuter-i-zastavil-ibm-nervnichat-50049940.html.

17. Stakhov A.P. The Mathematics of Harmony. From Euclid to Contemporary Mathematics and Computer Science / International Publisher «World Scientific» (New Jersey, London, Singapore, Beijing, Shanghai, Hong Kong, Taipei, Chennai), 2009.- P. 748.

18. Як працюєквантовіобчислення? [Електронний ресурс] // wfoojjaec. – 2016. – Режим доступу до ресурсу: https://wfoojjaec.eu.org/uk/projects/news/2019-01-30-how-does-quantum-computing-work.html.

19. Experiencetheimpactofquantumsolutionstoday [Електронний ресурс] // Блог компании Microsoft. – 2019. – Режим доступу до ресурсу: https://www.microsoft.com/en-us/quantum/development-kit.

20. GettingStartedwithQuantumProgramming [Електронний ресурс] // Hackernoon. – 2019. – Режим доступу до ресурсу: https://hackernoon.com/an-interactive-tutorial-on-quantum-programming-327da388f859.

21. Harris C. G., Stephens M. J. Combined corner and edge detector // Proc. Fourth Alvey Vision Conference. – 1988. – P. 147–151.

22. B. Smith. SUSAN - A new approach to low level, 1997

23. David G. Lowe «Distinctiveimagefeaturesfromscale-invariantkeypoints» InternationalJournalofComputerVision, 60, 2 (2004), pp.91-110

24. Rosten E., Drummond T. Machine learning for high-speed corner detection // Proc. European Conference on Computer Vision. – 2006. – V. 1. – P. 430–443.

25. M. Awrangjeb. CPDA .// IEEE CVPR'2009, Vol.2, 2009. – pp. 142-149.

26. Рейсиг Д. Инструменты отладки и тестирования.— СПб.: Питер, 2008. 76 с.— ISBN 978-5-91180-904-1.

27. Кавецький В. В. Економічне обґрунтування інноваційних рішень: Практикум /В.В.Кавецький, В.О.Козловський, І.В.Причепа. – ВНТУ, 2013. – 110 с.

28. Адлер О.О. Методичні вказівки до підготовки та написання курсової роботи з дисципліни «Економічне обґрунтування інноваційних рішень» / Уклад. О.О.Адлер, І.В.Причепа, Н.М.Тарасюк. – Вінниця: ВНТУ, 2014. – 38 с.