Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА

Галузь знань 12 – Інформаційні технології

Спеціальність 123 –Комп'ютерна інженерія

на тему «Метод забезпечення безпеки програм згідно оборотної логіки»

КвРКІП. 202146.22.02.35 ПЗ

Виконав: студент 2 курсу, група КІ2м-22-2 Підпис Кучерук Д.В. Ініціали, прізвище

Керівник к. т.н., доцент Науковий ступінь, вчене звання Підпис Березька К.М. Ініціали, прізвище

До захисту допускаю:
Зав. кафедри КІІС, д.т.н., проф.
Т.О. Говорущенко
__ _____ 2024 р.

Хмельницький, 2024

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень МАГІСТР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма ОСВІТНЬО-НАУКОВА ПРОГРАМА «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Т.О.Говорушенко

"01" 09 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА

Кучеруку Дмитру Віталійовичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Метод забезпечення безпеки програм згідно оборотної логіки
Керівник проекту (роботи) К.М. Березька, кандидат т.н., доцент

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 01.01.2024 р. № 1

2. Строк подання студентом проекту (роботи) на кафедру 01.05.2024 р.

3. Вихідні дані до проекту (роботи) Завдання на дипломне проектування

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) Провести аналіз відомих математичних моделей із оборотної логіки, розглянути логічну та фізичну оборотність, допустимість послідовної логіки в оборотних обчислювальних системах.

Розробити набір обчислювальних логічних примітивів, щоб забезпечити підтримку фізичної оборотності для всіх оборотних логічних структур із шляхами зворотного зв'язку.

Розробити найефективніший підхід до запобігання атакам DPA, щоб ускладнити для зловмисника визначення необхідної інформації для визначення вхідних даних.

Запропонувати реалізацію оборотної логіки в CMOS та SRAM, розробити алгоритм синтезу двошинної адіабатичної логіки.

Представити адіабатичний S-BOX з двома шинами для пом'якшення атак DPA та провести експериментальні дослідження для перевірки ефективності запропонованого методу забезпечення безпеки програм згідно оборотної логіки.

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

6. Консультанти розділів кваліфікаційної роботи магістра

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Лисенко С.М., професор кафедри КІС		
Антиплагіат	Нічепорук А.О., доцент кафедри КІС		

7. Дата видачі завдання «01» 09 2023р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) кваліфікаційної роботи магістра	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напряму дослідження та узгодження тематики КвРМ з керівником	01.09.2023	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.10.2023	виконано
3	Робота над розділом 1 – аналіз відомих моделей, методів за темою; постановка задачі	01.11.2023	виконано
4	Робота над розділом 2 – розробка моделей для вирішення поставленої задачі	01.12.2023	виконано
5	Робота над науковою статтею	01.02.204	виконано
6	Робота над розділом 3 – розробка методів для вирішення поставленої задачі	15.02.2024	виконано
7	Робота над розділом 4 – проектування та розробка ПЗ для вирішення поставленої задачі, експериментальна частина	01.04.204	виконано
8	Оформлення пояснювальної записки згідно вимог	18.04.2024	виконано
9	Попередній захист ДРМ	29.04.2024	виконано
10	Захист ДРМ на засіданні ЕК	До 15.05.2024	

Студент Підпис Кучерук Д.В. Ініціали, прізвище

Керівник роботи Підпис Березька К.М. Ініціали, прізвище

РЕФЕРАТ

Тема кваліфікаційної роботи магістра: Метод забезпечення безпеки програм згідно оборотної логіки.

Автор роботи: Кучерук Дмитро Віталійович

Керівник роботи: к.т.н., доц., Березька Катерина Миколаївна

Пояснювальна записка: 88 с., 0 рис., 1 табл., 3 дод., 88 джерел.

ОБОРОТНА ЛОГІКА, АДІАБАТИЧНА ДИНАМІЧНА ДИФЕРЕНЦІАЛЬНА ЛОГІКА, CMOS, АНАЛІЗУ ДИФЕРЕНЦІАЛЬНОЇ ПОТУЖНОСТІ, SRAM, МОДЕЛЮВАННЯ.

Об’єктом дослідження є процес забезпечення безпеки програм згідно оборотної логіки.

Предметом дослідження є метод забезпечення безпеки програм згідно адіабатичної динамічної диференціальної логіки для додатків у безпечному дизайні для пом’якшення атак ДАП.

Метою кваліфікаційної роботи магістра є реалізація адіабатичної динамічної диференціальної логіки для додатків у безпечному дизайні для пом’якшення атак ДАП.

Для розв’язання поставлених задач використовувалися методи:

1) аналіз математичних моделей із оборотної логіки;
2) аналіз допустимості послідовної логіки в оборотних обчислювальних системах;
3) моделювання фундаментальних оборотних логічних структур в адіабатичній логіці;
4) реалізація та експериментальні дослідження удосконаленої адіабатичної схеми з двома шинами для пом’якшення ДАП.

Наукова новизна отриманих результатів:

− Запропоновано математичний доказ того, що послідовні оборотні логічні структури фізично можливі. Розроблено набір обчислювальних логічних

примітивів, щоб забезпечити підтримку фізичної оборотності для всіх оборотних логічних структур із шляхами зворотного зв’язку.

− Удосконалено методологію розробки адіабатичної динамічної диференціальної логіки для пом’якшення атак ДАП на захищені інтегровані чіпи. Представлено адіабатичну схему із подвійною шиною для реалізації в шифрі Rijndael для алгоритму AES для додатків із низьким енергоспоживанням і низькою частотою, таких як смарт-карти на 13,56 МГц.

На основі проведених досліджень розроблено та реалізовано метод забезпечення безпеки програм згідно оборотної логіки.

Практична значимість отриманих результатів полягає у розробленні методу забезпечення безпеки програм згідно адіабатичної динамічної диференціальної логіки для додатків у безпечному дизайні для пом’якшення атак ДАП. Він полягає в тому, щоб включити логіку на основі безпеки в саму апаратну реалізацію, щоб ускладнити для зловмисника визначення необхідної інформації для визначення вхідних даних.

У першому розділі проведено аналіз математичних моделей із оборотної логіки, розглянуті основи оборотної та адіабатичної логіки. Також, в роботі звернуто увагу на смарт-картки. Розглянуто атаку диференціального аналізу потужності (ДАП) та алгоритм Rijndael.

У другому розділі розглянуто допустимість послідовної логіки в оборотних обчислювальних системах. Представлено результат моделювання послідовної оборотної логічної структури, запропоноване покращення пристрою вихідної пам'яті CMOS та структури SRAM.

Третій розділ роботи включає розгляд методу оптимізації синтезованих оборотних логічних структур. Проведено синтез двошинної адіабатичної логіки. Представлено алгоритм для синтезу адіабатичних логічних структур у CMOS.

У четвертому розділі запропоновано методологію розробки адіабатичної динамічної диференціальної логіки для пом’якшення атак ДАП на захищені інтегровані чіпи. Представлено удосконалену адіабатичну схему із подвійною шиною.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ 5

ВСТУП 7

1 АНАЛІЗ МАТЕМАТИЧНИХ МОДЕЛЕЙ 12

1.1 Основи оборотної логіки та моделювання схем з низьким енергоспоживанням 12

1.2 Метод запобігання атакам зловмисників 27

1.3 Постановка задачі 33

1.4 Висновки 34

2 ДОПУСТИМІСТЬ ПОСЛІДОВНОЇ ЛОГІКИ В ОБЧИСЛЮВАЛЬНИХ СИСТЕМАХ 36

2.1 Послідовні обчислювальні структури та моделювання оборотних систем 36

2.2 Пропозиція щодо вдосконалення CMOS та SRAM 44

2.3 Висновки 47

3 КОМУТАЦІЯ ФУНКЦІЙ, ЩО ЗМЕНШУЄ ПОТУЖНІСТЬ ЕНЕРГІЇ 49

3.1 Моделювання оборотних логічних структур 49

3.2 Удосконалення алгоритму з використанням двошинної адіабатичної логіки. 61

3.3 Висновки 77

4 УДОСКОНАЛЕННЯ МЕТОДУ ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ ПРОГРАМ ЗГІДНО ОБОРОТНОЇ ЛОГІКИ 80

4.1 Реалізація адіабатичної схеми 80

4.2 Порівняння контрольних показників 83

4.3 Висновки 86

ВИСНОВКИ 87

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ 89

ДОДАТОК А Копія публікації у виданні, що індексується в наукометричній базі Google Scholar 97

ДОДАТОК Б Сертифікат учасника Міжнародної конференції 103

ДОДАТОК В Презентація до магістерської роботи 104

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

AES – стандарт відомий під назвою Rijndael — симетричний алгоритм блочного шифрування

ДАП – диференціальний аналіз потужності

CMOS – комплементарний металооксидний напівпровідник.

SRAM – статична оперативна пам'ять з довільним доступом.

УПШ – універсальний програмований шлюз.

ЛВЗРР – логіка відновлення заряду на двох рівнях.

PMOS – технологія виробництва напівпровідникових елементів p-типу.

NMOS – технологія логіки на n-типу транзисторах.

HSPICE – галузевий «золотий стандарт» для точного моделювання схем і пропозицій, сертифікований на ливарному виробництві моделі MOS-пристроїв.

NAND – тип флеш-пам'яті, із звичайними транзисторними затворами.

PVC – полівінолхлорид.

TCP/IP – мережева модель передачі даних, що представлені у цифровому вигляді.

NFC – технологія бездротового зв'язку малого радіуса дії «за один дотик».

ISO – Міжнародна організація стандартизації.

IEC – Міжнародна електротехнічна комісія.

ОП – оперативна пам’ять.

ПП – постійна пам’ять.

ФЕОП – криптоконтролер, що розширює функціональні можливості карт пам’яті або мікропроцесорних карт, кодуючи та декодуючи збережені дані.

РІ – сигнал радіочастотної ідентифікації, який надсилають безконтактні смарт-картки.

ІС – інформаційна система.

РЛ – розширена логіка.

ДДЛ – динамічна та диференціальна логіка.

ХДДЛ – хвильова динамічна диференціальна логіка.

ЗКДДЛ – зменшена комплементарна динамічна та диференціальна логіка.

ЛЗДМт – логіка захищеного диференціального мультиплексора з використанням транзисторів проходу.

КТЛ – комплементарна транзисторна логіка.

АЛ – адіабатичних логіка.

КОАДЛ – квазістатична однофазна адіабатична динамічна логіка.

РДЛ – резонансна динамічна логіка.

ОС – операційна система.

НІСТ – Національний інститут стандартів і технологій.

DRAM – Динамі́чна операти́вна па́м'ять.

КВП – контрольний вентиль Переса.

ІК – інтегрований шлюз Кубіт.

AIDA. VHDL – поведінкові мови.

ДНК - дезоксирибонуклеи́новая кислота́.

ЛЕВЗ - логіка ефективного відновлення заряду.

ЕМАМ – евристичний метод алгоритму мінімізації.

ВАДДЛ – методологія проектування високопродуктивної адіабатичної динамічної диференціальної логіки для пом’якшення атак ДАП.

АДДЛЗ – адіабатична динамічна диференціальна логіка, зміщена за тілом.

ІВМ – американська електронна корпорація, один із найбільших світових виробників усіх видів комп'ютерів і програмного забезпечення.

TSMC – тайванська компанія, що займається вивченням та виробництвом напівпровідникових виробів.

РТМ – файл карти, який використовується програмним забезпеченням картографування MapPoint, розроблений корпорацією Майкрософт

ВСТУП

Актуальність роботи. Виробництво економічно ефективних безпечних інтегрованих мікросхем, таких як смарт-карти, вимагає від розробників апаратного забезпечення врахування компромісів у розмірі, безпеці та енергоспоживанні. Для створення успішних проектів, орієнтованих на безпеку, апаратне забезпечення низького рівня повинно містити вбудовані механізми захисту, які доповнюють криптографічні алгоритми, такі як AES, запобігаючи атакам на бокових каналах, таким як диференціальний аналіз потужності (ДАП). Динамічна логіка затьмарює вихідні сигнали та роботу схеми, знижуючи ефективність атаки ДАП.

Принципи квантової механіки керують фізичними обмеженнями обчислювальних схем і систем. Ці системи розсіюють енергію через стирання бітів у своїх взаємопов’язаних примітивних структурах, що є важливим фактором, оскільки щільність транзисторів зростає. Збільшення ентропії в цих середовищах безпосередньо пов’язане з ймовірністю того, що квантова частинка займе будь-який із своїх станів. Щоб створити ідеальний універсальний комп’ютер, який розсіює довільно низьку енергію, має бути реалізована обертова логіка, оскільки закони фізики вказують на оборотність у часі.

Оборотні логічні структури є задовільними для проектування та реалізації в обчислювальних структурах та організації, коли ці правила проектування забезпечують оборотність логічної структури [1]. Таким чином, універсальна обчислювальна машина може бути реалізована для того, щоб ідеально моделювати кожну кінцево реалізовану фізичну систему, оскільки кожен електрон у квантовому комп’ютері представлений постійним унітарним оператором у гамільтоновому просторі [2].

Основний принцип оборотного обчислення полягає в тому, що бієктивний пристрій з однаковою кількістю вхідних і вихідних ліній не матиме розсіювання тепла. Електродинаміка системи дозволяє передбачити всі майбутні стани на

основі відомих минулих станів, і система досягає кожного можливого стану. Є дві окремі, але однаково важливі парадигми оборотної логіки:

1) логічна оборотність, яка використовує принципи, які керують оборотною логічною структурою, щоб визначити логічні обчислення, необхідні для здійсненних проектів;

2) фізична оборотність, яка передбачає розробку фізичної структури, вхідні значення якої можуть однозначно визначатися виходом кожного обчислювального циклу, і розсіювання енергії якої не перевищує бар'єр Ландауера kTln(2) джоулів на обчислювальний цикл.

Різниця між цими двома парадигмами є важливою, оскільки логічно-обертова структура все ще може перевищувати бар'єр Ландауера. Наприклад, CMOS – це інвертор, що працює при кімнатній температурі (298 K), VDD якого становить 1 В і має вихідну ємність 100 пФ, буде розсіювати $5 \cdot 10^{-11}$ Джоулів на перехід стану, що в $1{,}75 \cdot 10^{10}$ разів більше, ніж kTln(2), навіть якщо інвертори логічно оборотні.

Реалізацією оборотної логіки в CMOS, де струм (що протікає через схему) контролюється, щоб мінімізувати розсіювання енергії через перемикання є – адіабатична логіка. Існують значні дослідження щодо проектування та аналізу локально оптимальних адіабатичних елементів для пом'якшення атак бічних каналів. Однак жодна з цих робіт не розглядала використання адіабатичної логіки в реалізації гнучких і програмованих політик апаратної безпеки. Адіабатична логіка також не застосовувалася в додатках апаратної безпеки, таких як надійні системи голосування та стандарти шифрування даних.

Адіабатичну теорему вперше представили Борн і Фок [3]. Вони описують фізичну систему такою, що залишається у своєму миттєвому власному стані, якщо збурення діє досить повільно і якщо існує розрив між власним значенням і рештою спектра гамільтоніана. Тому, уповільнюючи зміну умов, система сама адаптується до нової конфігурації, змінюючи щільність ймовірності. Це означає, що якщо система починається у власному стані початкового гамільтоніана, вона закінчиться у відповідному власному стані кінцевого гамільтоніана [4].

В кваліфікаційній роботі я звернуся до двох основних джерел теоретичних дебатів в адіабатичній і оборотній логіці. В роботі розглянуто питання про те, чи можна маніпулювати схемами перемикання потоку електронів оборотно за допомогою логічних структур CMOS. Представлені результати симуляції прикладу адіабатичної логіки, де бінарна комутаційна мережа розсіює менше kTln(2) джоулів енергії за подію комутації. Також, розглянута допустимість послідовної логіки в оборотних обчислювальних системах. Представлені математичні викладки того, що послідовні оборотні логічні структури фізично можливі. Розроблено набір обчислювальних логічних примітивів, щоб забезпечити підтримку фізичної оборотності для всіх оборотних логічних структур із шляхами зворотного зв'язку.

Розвитком підходу до проектування оборотної логіки займалися Дойч [5], Морісон [6], Перес [7], Фейнман [8] та інші. Численні статистичні методи були використані для моделювання фізичних структур оборотної логіки, найвидатніші з яких – Максвелла-Больцмана, Фермі-Дирака та Бозе-Ейнштейна [9].

Метою магістерської роботи є реалізація адіабатичної динамічної диференціальної логіки для додатків у безпечному дизайні для пом'якшення атак ДАП.

Для досягнення мети роботи необхідно вирішити наступні завдання:

1. Провести аналіз відомих математичних моделей із оборотної логіки, розглянути важливі парадигми оборотної логіки – логічну та фізичну оборотність, допустимість послідовної логіки в оборотних обчислювальних системах.

2. Розробити набір обчислювальних логічних примітивів, щоб забезпечити підтримку фізичної оборотності для всіх оборотних логічних структур із шляхами зворотного зв'язку.

3. Розробити ефективний підхід до запобігання атакам ДАП, щоб ускладнити для зловмисника визначення необхідної інформації для визначення вхідних даних.

4. Запропонувати реалізацію оборотної логіки в CMOS та SRAM, розробити алгоритм синтезу двошинної адіабатичної логіки.

5. Представити удосконалену адіабатичну схему з двома шинами для пом’якшення атак ДАП та провести експериментальні дослідження для перевірки ефективності запропонованого вдосконалення методу забезпечення безпеки програм згідно оборотної логіки.

Об’єктом дослідження є процес забезпечення безпеки програм згідно оборотної логіки.

Предметом дослідження є метод забезпечення безпеки програм згідно адіабатичної динамічної диференціальної логіки для додатків у безпечному дизайні для пом’якшення атак ДАП.

Для розв’язання поставлених задач використовувалися методи:

1) аналіз математичних моделей із оборотної логіки;
2) аналіз допустимості послідовної логіки в оборотних обчислювальних системах;
3) моделювання фундаментальних оборотних логічних структур в адіабатичній логіці;
4) реалізація та експериментальні дослідження удосконаленої адіабатичної схеми з двома шинами для пом’якшення ДАП.

Наукова новизна одержаних результатів полягає в наступному:

1. Запропоновано математичне обгрунтування того, що послідовні оборотні логічні структури фізично можливі. Розроблено набір обчислювальних логічних примітивів, щоб забезпечити підтримку фізичної оборотності для всіх оборотних логічних структур із шляхами зворотного зв’язку;

2. Удосконалено методологію розробки адіабатичної динамічної диференціальної логіки для пом’якшення атак ДАП на захищені інтегровані чіпи. Представлено адіабатичну схему із подвійною шиною для реалізації в шифрі Rijndael для алгоритму AES для додатків із низьким енергоспоживанням і низькою частотою, таких як смарт-карти на 13,56 МГц.

Практична цінність дипломної роботи полягає в удосконалені методу забезпечення безпеки програм згідно адіабатичної динамічної диференціальної логіки для додатків у безпечному дизайні для пом’якшення атак ДАП. Він полягає

в тому, щоб включити логіку на основі безпеки в саму апаратну реалізацію, щоб ускладнити для зловмисника визначення необхідної інформації для визначення вхідних даних.

За темою магістерської роботи опубліковано статтю на тему «Метод забезпечення безпеки програм згідно оборотної логіки» в матеріалах конференції X International Scientific and Practical Conference Science, Education, Innovation Topical Issues and Modern Aspects, 2024, Tallin, Estonia, що індексуються в наукометричній базі Google Scholar [10, 88], (Додаток А) та отриманий сертифікат участі у конференції з кількісттю годин дистанційної роботи – 12 годин (0,4 ECTS credits) (Додаток Б).

1 АНАЛІЗ МАТЕМАТИЧНИХ МОДЕЛЕЙ

1.1 Основи оборотної логіки та моделювання схем з низьким енергоспоживанням

Математичні моделі за своєю суттю є абстрактними об’єктами. Однак, щоб створити математичну модель, яка точно відображає природу структури, вона повинна врахувати взаємодію частин об’єкта та природу його обчислення. Багато з представлених робіт із оборотної логіки [11] є – уявними експериментами – щодо пружних зіткнень класичних більярдних куль, які представляють ідеальні зіткнення квантових частинок, або перевороту квантованих спінів для представлення бінарних логічних операцій. Результатом цих експериментів є розробка теоретичної математичної моделі для квантових обчислень для систем з низьким енергоспоживанням, яка застосовна для будь-якої нової технології.

У системі квантових частинок неможлива одноразова передача тепла від тіла з нижчою температурою до тіла з вищою температурою без іншої пов’язаної зміни, що відбулася в той самий час [12]. У системі, де відбувається N перетворень, загальна зміна значення еквівалентності є сумою значень еквівалентності, яка рівна швидкості утворення тепла, поділеній на температурну функцію.

Отже, сума всіх теплових перетворень у циклічному процесі, такому як двигун Карно, повинна бути невід’ємною.

Формулу $\sum_{i=1}^{n} \frac{Q_i}{T_i} = \int \frac{dQ}{T}$ було визначено як ентропію системи, а різниця між початковою та кінцевою ентропією системи [13] становить $\int \frac{dQ}{T} = S - S_0$.

Оборотна система — це унікальна система, у якій кожна частинка досягає всіх можливих станів. Оскільки ці перетворення компенсують одне одного, зміна ентропії дорівнює нулю, тобто тепло не розсіюється в системі. Ентропія системи прямо пропорційна логарифму від енергії, об’єму та кількості частинок у системі, а також газовій константі [14]. Ентропія системи визначається за допомогою

рівняння $\int \frac{dQ}{T} = k \ln(W)$, а зміна ентропії в системі знаходиться шляхом порівняння початкового і кінцевого станів системи.

Оскільки ентропія безпосередньо пов'язана з кількістю частинок у системі, розподіл енергетичних елементів має бути скінченним цілим числом. Електромагнітна енергія може випромінюватися лише в дискретних квантованих кількостях, і загальна кількість можливих станів має бути скінченною.

Квантовий стан [15] усіх квантових взаємодій пов'язаний з імпульсом частинки: $ih\frac{\partial}{\partial t}\psi(x,t) = -\frac{h}{2m}\nabla^2\psi(x,t) + V(x)\psi(x,t)$. Це співвідношення було використано для представлення фізичної перспективи квантової електродинаміки шляхом використання просторово-часових діаграм для інтерпретації взаємодії електронів, коли вони дуже близькі один до одного [11]. Ці представлення дозволяють розрахувати будь-який квантовий процес, що включає передачу енергії [16]. Стан S_i в точці *i* простору-часу задається функцією $S_i = F_i(S_i \dots S_k)$.

Отже, для того, щоб симулювати час, функція F, яка повинна мати можливість передбачати майбутній стан на основі минулих станів, а також знати всі минулі та майбутні стани. Зокрема, у оборотному процесі буде представлено кожну минулу конфігурацію електронів, що кожен можливий стан буде досягнутий квантами, а їх взаємодія буде такою, що попередній стан буде однозначно визначатися.

Розглянемо універсальну обчислювальну машину, що є одиничним автоматичним пристроєм, робота якого повністю визначається його взаємопов'язаними примітивними структурами. Вона виводить або логічний «0», або «1», і здатна обчислювати розрахункову послідовність [17]. Спочатку стверджувалося, що обчислювальні пристрої не можуть виконувати оборотні операції, що призводило до розсіювання енергії та збільшення ентропії [18]. Теоретичне мінімальне розсіювання енергії необоротного обчислювального пристрою було розраховано шляхом зв'язку кількості станів, досягнутих квантовими частинками на вході та виході структури, а також їх ймовірностей до зміни ентропії. Сценарій, коли виникає мінімальна кількість розсіюваного тепла в

необоротній системі, при ідентичності всіх вхідних та вихідних ймовірностей та підстановка цих значень у рівняння ентропії дає $k_B TIn(2)$, який відомий як бар'єр Ландауера. Це означає, що неможливо розробити обчислювальну систему, яка повністю складається з оборотних систем, оскільки вони неминуче вимагали б обчислення без однозначного зворотного. У результаті будь-який двійковий пристрій з одним ступенем свободи буде розсіювати енергію, пропорційну значенню kT. Ландауер визначив стан, коли частинка перебувала в лівій частині, як «0», а в правій — як «1». У цій конкретній системі він визначив стан «1» як стан рівноваги, коли для досягнення цього стану не потрібна сила, тоді як сила потрібна, щоб частинка набрала енергію, щоб перевищити бар'єр та досягти стану «0». Для того, щоб утримувати частинку в цьому стані «0», необхідно застосувати гальмівну силу. Він стверджує, що ця система не є оборотною в часі, оскільки не існує залежного від часу рівняння сили, яке можна було б використовувати для визначення того, чи була частинка у стані «1» на попередньому часовому етапі, або чи була вона у стані «0». Якщо сила, необхідна для переведення частинки в стан «0», більше не прикладається в момент часу t, а час, необхідний для повернення в стан «1», дорівнює Δt, тоді ми більше не можемо інвертувати місце розташування частинки після часу t+Δt. Це пояснюється тим, що для частинки однаково можливо перебувати в стані «1» вже в момент часу t, а потім просто залишатися там протягом Δt. Крім того, необхідна сила затримки часу додає ще один потенційний стан для частинки, який перевищує двійкову природу обчислювальної системи. Традиційна обчислювальна система поширює сигнали через структуру незалежно від вхідних даних і є лише функцією фізичної схеми.

Згодом Ландауер представив три класи обчислювальних структур для більш ретельного визначення незворотного процесу. Першою системою була кріотронна система, яка являла собою теоретичні пристрої, які не розсіюють енергію під час зберігання інформації, але розсіюють енергію під час перемикання зі стану в стан. Вони відрізняються від попереднього пристрою, оскільки ні стани «0», ні «1» не існують у локальних мінімумах, і, таким чином, не є рівнодійними станами, але ця система більш репрезентативна для роботи комп'ютера. Логічна структура може

бути розроблена з використанням пристроїв першого класу, але можуть бути побудовані обчислювальні структури, які містять або тільки кріотрони, або тільки магнітні сердечники.

Другий тип системи складається з пристроїв у стаціонарному стані, які розсіюють енергію, утримуючи інформацію. Вони мають переходи від бажаного стану до іншого стабільного стаціонарного стану та є послідовними за своєю природою. До них належать пристрої, як засувки, тригери та структури пам'яті.

Ландауер назвав третю систему «всеохоплюючим» пристроєм, де інформація, що зберігається в системі, залежить від дисперсії в часі поширення сигналу через структуру [19]. Це пов'язано з тим, що ємність пам'яті було збільшено за рахунок використання малих бістабільних об'ємів для кожного біта пам'яті. Однак їх не можна було зробити надто маленькими, оскільки це призвело б до збільшення квантового тунелювання та термочутливості.

Оскільки вхідний стан не може бути однозначно визначений шляхом вимірювання вихідного стану, було сказано, що необоротний пристрій втрачає інформацію, що призводить до збільшення ентропії. У двійковому обчислювальному пристрої, де можуть бути отримані всі потенційні вхідні та вихідні стани, різниця ентропії визначається як $kln(2^{N_{вих}}) - kln(2^{N_{вх}})$.

Де розв'язок для *Q=ktln2.* Тут результат дає мінімальну кількість утворення тепла Q за фіксований цикл обчислення. Співвідношення між енергією, що розсіюється на біт енергії, відоме як бар'єр Ландауера, і прийнято як теоретична нижня межа обчислень для незворотних пристроїв. При кімнатній температурі (25^0C) значення бар'єру Ландауера становить $2{,}805*10^{-21}$ Дж за одиницю часу. Ландауер показав, що ця втрата інформації буде результатом двох додаткових факторів: неповного переходу з одного стану в інший через надмірні тактові частоти та розпаду збереженої інформації через температурні коливання.

В свою чергу Беннетт стверджував, що просту, надійну машину Тьюрінга можна зробити логічно оборотною без втрати інформації [20]. Він зазначив, що твердження про те, що було непрактично повернути назад кроки початкового

обчислення машини Тьюрінга [21], не було повним, оскільки оборотну обчислювальну структуру можна створити, щоб стерти свою історію

Однак це спричинило іншу проблему. Якщо виконується детермінований, оборотний обчислення, а потім виходи негайно стираються, щоб досягти початкового стану шляхом зворотного обчислення, тоді бажаний результат втрачається, роблячи обчислювальну структуру повністю марною. Беннетт вирішив цю проблему, дозволивши операцію «Копіювати». Додатковою перевагою є те, що в обчислювальній структурі не зберігається історія, але пристрій є повністю оборотним і детермінованим.

Щоб продемонструвати це формально, розглянемо звичайну машину Тьюрінга S і оборотну машину Тьюрінга R. R можна використовувати для імітації всіх обчислень, можливих у S, з додатковим обмеженням, що в кінці обчислення структура повинна мати бажані вихідні та початкові вхідні розрахунки. R виконуватиме свої обчислення наступним чином. По-перше, вона виконує обчислення подібно до необоротної структури, за винятком того, що кожен проміжний результат зберігається. Збереження цих результатів запобігає стиранню бітів, що виключає збільшення ентропії, викликане незворотною операцією. По-друге, подібно до будь-якого іншого комп’ютера, пристрій повинен друкувати вихідні обчислення. Останній етап вимагає виконання обчислень у зворотному порядку, що гарантує, що система досягне свого початкового стану, задовольняючи фізичну вимогу оборотності. Так машина S складається з головки читання/запису, нескінченної стрічки T, поділеної на квадрати, кожен з яких служить структурою пам’яті, і контролера A. Структура використовує засоби читання-запису-зміщення для керування проміжними кроками переходу. для розрахунків, результати яких заносяться на стрічку.

Машина Тьюрінга функція чотирьох змінних: $S = S(K, \sum, \delta, s)$, де K — множина станів; Σ — алфавіт; δ — множина інструкцій; s — початковий стан машини.

Отже, якщо ми визначимо дві четвірки змінних, що складаються з n стрічок: $\alpha \equiv A[t_1,\ldots t_n] \rightarrow [t`_1,\cdots t`_n]A`$ і $\beta \equiv B[u_1,\cdots u_n] \rightarrow \left[u`_{1`,\ldots}u`_n\right]B`$, де A і B – контролери для α і β відповідно.

То можемо визначити три унікальні властивості, необхідні для реверсивної n-стрічки Машини Тьюрінга. Машина R зроблена оборотною з S шляхом додавання перехідних станів, які нагадують інверсії початкових переходів. Наприклад, інверсія Тьюринга «читання-запис-зсув» вимагає «зсуву-читання-запису» для досягнення оборотності. Зокрема, прості формули переходу використовуються в S, де будь-яка стрічка підлягає операції читання-запису або операції зсуву, але не обох операцій. Це можливо, якщо α і β визначають обернене відображення S. Це можливо тоді і тільки тоді, коли обернене відображення отримано шляхом заміни початкового керуючого стану кінцевими символами стрічки читання та зміни знаків усіх зрушень, що означає, що A=B' і B=A'. Області α і β можуть перекриватися тоді і тільки тоді, коли A=B та A'=B' [22].

Отже, якщо S є універсальною машиною Тьюринга, то R стає універсальною оборотною машиною Тьюринга. Насправді, будь-яке обчислювальне завдання, можливо виконати в логічно оборотному комп’ютері без надмірного збільшення складності машини, кількості кроків, небажаного виведення або тимчасової ємності зберігання.

Щоб продемонструвати різницю в мінімальному розсіюванні енергії в оборотних і необоротних обчисленнях, давайте розглянемо реалізацію універсального програмованого шлюзу (УПШ) [23], який складається з інтегрованого затвору Кубіта. Затвор I не є однозначними, оскільки користувач не може визначити вхідну комбінацію, коли на виході є логічний «0». Використовуючи таблицю істинності «ТА» і рівняння ентропії, ми визначаємо, що мінімальний приріст ентропії цього пристрою дорівнює $k_B\left(-\frac{3}{4}\cdot In\left(\frac{3}{4}\right)+\right.$ $\left.+\frac{1}{4}\cdot In\left(\frac{1}{4}\right)\right)-k_B\left(4\cdot\frac{1}{4}\cdot In\left(\frac{1}{4}\right)\right)$, що дає тепловиділення $\frac{3}{4}k_B N\,In(3)$, яке в 1,2 рази перевищує бар'єр Ландауера. УПШ має однакову кількість входів і виходів і

є біективним, тобто $\int_{N_{\text{вх}}}^{N_{\text{вих}}} \frac{dQ}{T} = k_B(8 * In(8)) - k_B\big(8 * In(8)\big) = 0,$ що означає – пристрій фізично оборотний.

У [8] Фейнман поставив два дуже важливі питання, які керують нашим підходом до проектування оборотної логіки: чи можна змоделювати фізичні процеси за допомогою універсального комп'ютера, і які фізичні процеси ми будемо імітувати? Перше питання стосується питання локальних взаємозв'язків. Основна проблема з'єднання полягає в тому, що дроти є основним джерелом розсіювання енергії в обчислювальних структурах через зміни напруги та внутрішнього опору дроту [24]. Другий передбачає розгляд того, чим обчислювальна техніка відрізняється від фізичного закону. В обчислювальній структурі існує кінцева кількість вхідних і вихідних комбінацій, а також кінцева кількість логічних обчислень. Навіть розмір комп'ютера обмежений. Це відрізняється від фізичних процесів, де простір можна виміряти. Оскільки фізичні знання завжди неповні, мета будь-якої математичної моделі оборотної логічної структури полягає в тому, щоб розробити теоретичну структуру, яка перевершує експеримент в даний час. Наприклад, розглянемо рівняння для квантової взаємодії в просторі-часі (1.7). Це дозволяє описати стільникову автоматизацію шляхом обчислення даної точки з точок у попередні моменти, що дозволяє обчислювати наступні значення. Але що, якщо функція F залежить від усіх точок у минулому? Виявляється, єдиний тип обчислювальної системи, який може успішно імітувати цю природу фізики, — це оборотна система. Численні статистичні методи були використані для моделювання цих фізичних структур – Максвелла-Больцмана, Фермі-Дірака та Бозе-Ейнштейна [9]. У комп'ютері загальна ймовірність представлена кількістю вхідних і вихідних станів, 2^N. Реверсивна обчислювальна система є імовірнісною обчислювальною системою.

Фейнман показав систему, що складається з чотирьох математичних операторів для представлення створення та знищення цих частинок. Матриці два на два представляють дві можливі бази, які є незайнятими або зайнятими. Розглянемо матрицю для анігіляції, показану в рівняннях нижче, і результати в

рівнянні $\frac{1}{2}(\sigma_x - i\sigma_y)$. Це випадок, коли частинка спочатку займає простір, а потім анігілює через квантову взаємодію. Якщо він займає місце, то стає незайнятим.

Теорія, представлена Фейнманом для систем Бозе-Ейнштейна, була доведена та розширена в [5] Дойчем. Оскільки класична фізика припускає неперервну систему, її неможливо змоделювати за допомогою універсальної машини Тюрінга. Однак квантова фізика має дискретний характер. Дойск модифікував принцип Черча-Тюрінга [25] фізично, щоб стверджувати: кожна кінцево реалізована фізична система може бути ідеально змодельована за допомогою універсальної моделі обчислювальної машини, що працює кінцевими засобами. Це означає, що закони фізики допускають існування фізичних моделей для арифметики. До цього моменту найбільш прогресивна робота з квантово-механічного автомата була в [26], де квантова клітинна автоматизація використовувалася для пізнання та передбачення властивостей рівнянь руху. Дойч спирався на це, розробивши загальну, повністю квантову модель для обчислень. Такий комп'ютер побудований на основі всіх звичайних операцій Тьюрінга, а також восьми додаткових операцій. Вони відомі як унітарні перетворення, які представляють квантову взаємодію у двовимірному гільбертовому просторі. Дано ірраціональний множник α з π, перші чотири перетворення гільбертового простору показані як $V_0 = \begin{pmatrix} \cos \propto & \sin \propto \\ -\sin \propto & \cos \propto \end{pmatrix}, V_1 =$

$= \begin{pmatrix} \cos \propto & \mathrm{i} \sin \propto \\ i \sin \propto & \cos \propto \end{pmatrix}, V_2 = \begin{pmatrix} e^{i\propto} & 0 \\ 0 & 1 \end{pmatrix}, V_3 = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\propto} \end{pmatrix}.$

Оператори від V_4 до V_7 є зворотними, знайдених за допомогою рівняння $A^{-1} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \frac{1}{ad-bc}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix}.$

Ця система також може використовувати два зручні генератори, які відповідають поворотам обертання на 90 градусів, які можуть бути використані для створення будь-якого з оригінальних восьми перетворень простору Гільберта. Вони відомі як унітарні перетворення Лі, і їх представлення є $V_8 = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}$ та $V_9 = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & i \\ i & 1 \end{pmatrix}.$

У [7] Перес розробив метод побудови локально оборотного квантово-механічного гамільтоніана, динамічна еволюція якого може бути представлена серією логічних операцій. Це досягається шляхом забезпечення того, що кожна локальна операція є оборотною і коди коригування помилок вбудовані в сам гамільтоніан. Це було зроблено, щоб усунути занепокоєння, що оборотна логіка неможлива за наявності шуму [27], який спричинив би втрату бітової інформації та, як наслідок, розсіювання тепла. Щоб досягти цього, Перес розрізнив локальну оборотність і глобальну оборотність. Прикладом глобальної оборотної системи може бути генератор випадкових чисел $x_n \to x_{n+1} = ax_n (\mathrm{mod} b)$.

Значно складніше отримати x_n з x_{n+1}, ніж навпаки, оскільки потрібно значно більше операцій. А локально оборотна система складається лише з кількох бітів, а це означає, що потрібне зворотне обчислення така ж невелика кількість бітів. Як приклад він навів «Універсальний реверсивний шлюз», який зараз широко відомий як шлюз Переса, як реверсивна логічна структура 3*3: P=S, Q=A⊕B та R=AB⊕C.

Цей шлюз може бути отриманий в термінах обертових операцій і виконання над ним операції прямого добутку. Прямий добуток матриць включає дві задані матриці $a\ i\ b$, які перетворюватимуть вектори x і y, тому $\varepsilon = ax$ та $\mu = by$. Тому a⊕b це матриця, яка перетворює належним чином упорядкований набір бінарних добутків $x_i y_j$, [28] тому $\varepsilon_i \mu_j = \sum_{r,s} a_{jr} b_{js} x_r y_s$.

Саді Карно продемонстрував, що ідеальний тепловий двигун, кроки якого є оборотними, матиме однакову енергоефективність, $(T_H - T_L)/T_H$, і що не існує теплового двигуна, який був би ефективнішим, ніж оборотний двигун [4]. Це пояснюється тим, що двигун мав би створювати роботу з нічого, що порушувало б другий закон термодинаміки Ньютона. Будь-який реверсивний двигун Карно повинен пройти чотири чіткі стадії: ізотермічне розширення, де обсяг збільшується без зміни температури; адіабатичне розширення, де об’єм збільшується, коли температура змінюється від T_H до T_L; ізотермічного стиснення, де об'єм зменшується без зміни температури; і адіабатичного розширення, коли об’єм зменшується, а температура змінюється від T_L до T_H.

Адіабатична теорема стверджує, що «фізична система залишається у своєму миттєвому власному стані, якщо дане збурення діє на неї досить повільно і якщо існує розрив між власним значенням і рештою спектра гамільтоніана» [29]. Оскільки схеми CMOS працюють на тактових циклах, адіабатичний логічний дизайн призводить до калібрувально-інваріантної фази Беррі. Зазвичай, коли хвилі піддаються варіаціям, які самозгортаються, початковий і кінцевий стани системи будуть відрізнятися. Щоб запобігти цьому, адіабатичні системи розроблені оборотно, щоб система завжди могла досягти свого початкового стану, незалежно від кількості циклів, в яких вона працює.

Основною перешкодою для використання адіабатичної логіки в комп'ютерному проектуванні є те, що повна адіабатичність означає абсолютно нульову швидкість генерації ентропії. Це вимагало б нескінченного ступеня ізоляції системи від неконтрольованого зовнішнього середовища. На практиці процес є адіабатичним у тій мірі, якою їх генерація ентропії наближається до нуля. Для цього використовується термін «квазіадіабатичний».

Таким чином, метою адіабатичного логічного проектування є використання принципів оборотної логіки, щоб мінімізувати розсіювання енергії в схемах CMOS. У будь-якому адіабатичному контурі необхідно вирішити дві проблеми:

1) реалізація повинна призвести до енергоефективної конструкції комбінованого джерела живлення та тактового генератора;

2) оборотні логічні функції вимагають більших логічних витрат, щоб задовольнити біективну вимогу [30].

Таким чином, енергію, що розсіюється під час перемикання ланцюга, необхідно контролювати та переробляти, а не розсіювати в навколишнє середовище.

Зменшення розсіювання енергії досягається за рахунок використання функції наростання замість швидшого перемикання, досягнутого в ступінчастих функціях. Таким чином, транзистори можуть використовуватися в адіабатичній роботі, незважаючи на те, що вони продемонстровані як пристрої з втратами [30], і це досягається шляхом застосування двох правил. По-перше, транзистор завжди

включений, коли через нього протікає значний струм. По-друге, коли є значна різниця між джерелом і напругою стоку, транзистор повинен бути вимкнений. У [31] було показано, що адіабатичні схеми забезпечують зниження розсіювання енергії на 60% при 20 МГц і на 35% менше енергії при 100 МГц, а в [32] продемонстровано реверсивні двошинні CMOS-транзистори.

Проблема адіабатичних ланцюгів, сконструйованих із подвійними ланцюгами, полягає в тому, що вони не дозволяють оборотності між кількома етапами послідовної конвеєрної логіки. Цю проблему було вирішено за допомогою логіки відновлення заряду на двох рівнях (ЛВЗРР) [33].

Справді адіабатичні схеми мають бути фізично однозначними, тобто вихідні сигнали можуть бути розміщені на виходах, а унікальні вхідні сигнали можуть бути відтворені на вхідних проводах. Для досягнення цієї мети в [34] використовувався підхід «подвійних рейок», в якому три фундаментальні оборотні логічні структури були розроблені та виготовлені за технологією 0,35 мкм, де $V_{tp} = 0.6V$ і $V_{tn} = -0.6V$. Схеми не мають входів джерела живлення, тобто вся енергія вихідних сигналів походить від вхідних сигналів. Цей метод вдосконалено на ЛВЗРР і ЛЕВЗ шляхом значного зменшення накладних витрат, необхідних для виконання оцінки та розрядки, а також покращення розповсюдження сигналу, що дозволяє покращити каскадування пристроїв.

Багато ранніх реалізацій адіабатичних CMOS-схем використовували діоди на шляху повернення заряду, щоб забезпечити захист від електростатичного розряду або ізоляцію пристрою. Діоди діють як односпрямований струмовий бар'єр, що означає, що вони фізично незворотні. Тому при кожній роботі пристрою, що використовує діод, відбувається мінімальне розсіювання енергії через втрату інформації. Це пояснюється тим, що діод у реверсивній роботі працюватиме як «перемикач Максвелла».

Дж. Максвелл запропонував квантову систему, у якій швидші та повільніші частинки були розділені на два відсіки. Мініатюрний перемикач розміщувався між двома камерами. Коли швидша частинка наблизилася до дверей, перемикач відкрив їх, дозволяючи частинці увійти в іншу камеру. Результатом є те, що

температура однієї камери буде підвищуватися і знижуватися в іншій камері, але це буде зроблено без витрат праці, яка б знизила ентропію. Протиріччя з перемикачем Максвелла полягало в тому, що для того, щоб перемикач знав, які частинки швидкі, а які повільні, він мав би виміряти швидкість частинки, але акт отримання цієї інформації вимагав енергії та збільшення ентропії системи. [35]. Це означає, що діод, який не розсіює енергію, був би змушений генерувати енергію з нічого, тобто він функціонував би так само, як вічний двигун. Діоди обмежують здатність квазіадіабатичного пристрою наближатися до нульової ентропії. Тому діоди не допускаються в адіабатичних конструкціях.

Застосування прямого зміщення до транзисторів зменшує порогову напругу та підвищує продуктивність пристрою. Застосування зсуву тіла вперед покращує поведінку відкату Vt і дозволяє використовувати коротші шлюзи. Таким чином, схема прямого зміщення корпусу може використовуватися в деяких важливих аспектах для розширення масштабування технології bulk-Si CMOS [36]. Зміщення прямого корпусу використовувалося в CMOS для покращення розсіювання потужності та здатності перемикання подвійних рейкових адіабатичних інверторів шляхом дозволу роботи під порогом [37]. Крім того, зміщення корпусу в транзисторах покращує вразливість схеми CMOS проти атак ДАП [38]. Оскільки порогова напруга підтягуючих транзисторів, що використовуються для відновлення заряду, змінюється пороговою напругою, саме відновлення погіршується, збільшуючи різницю між піковим і середнім енергоспоживанням і робить схему більш уразливою до атак аналізу потужності.

Тепер розглянемо детальніше основи адіабатичного проектування двох рейок. Для реалізації адіабатичної логіки в CMOS необхідно дотримуватися двох правил:

1) транзистор ніколи не можна вмикати, коли на ньому є напруга. Це означає, що якщо бажана напруга на стоці та витоку різна, то транзистор потрібно вимкнути. Якщо це правило порушується, тоді енергія розсіюється і інформація втрачається;

2) друге правило полягає в тому, що відмінна від нуля напруга ніколи не повинна прикладатися до транзистора під час будь-якого переходу. Якщо це відбувається, то внутрішній опір транзистора є відносно малим, що призводить до дуже високого стрибка потужності та, як наслідок, розсіювання енергії.

Для досягнення цих проектних цілей CMOS-транзистори використовуються як перемикачі, і вони вмикаються лише тоді, коли ми хочемо, щоб напруга джерела та стоку були однаковими. Крім того, вхідними сигналами необхідно керувати за допомогою динамічного перемикання замість звичайних прямокутних форм сигналу, і синхронізувати їх так, щоб два вхідні сигнали не перемикалися одночасно, оскільки це порушило б друге правило.

У нашій методології проектування ми використовуємо адіабатику з двома шинами, представлену Ван Рентергемом і Де Вос [34]. У цьому методі кожен комутатор має додатковий транзистор PMOS і NMOS, який включений для підвищення надійності схеми.

Остаточна конструкція з подвійною шиною не має входів джерела живлення V_{dd}. Цей вибір конструкції є прийнятним, оскільки вхідні сигнали контролюються так, щоб зміна напруги стоку транзисторних перемикачів булла така, щоб увімкнути транзистор можна було б лише тоді, коли на ньому немає напруги.

Як приклад ефективності адіабатичного дизайну з подвійною шиною, представимо шлюз Тоффолі [1]. Шлюз Тофолі може мати високий стрибок покращеної миттєвої потужності, яку отримує адіабатичний затвор Тоффолі порівняно зі звичайними чотирма транзисторним затвором NAND. Шлюз NAND порушує адіабатичні та оборотні правила двома способами:

1) він не односторонній, тому логічно необоротний;

2) оскільки перемикання відбувається швидше, ніж адіабатична реалізація затвора Тоффолі, відмінна від нуля напруга прикладається до транзистора під час кожного переходу.

Тому внутрішні опори транзисторів зменшуються під час перемикання.

Отже, три основні цілі проектування оборотної логіки такі;

1) мінімізація квантової вартості (кількості оборотних обчислень 1*1 і 2*2, необхідних для створення логічного виводу) зменшить обчислювальну складність пристрою;
2) мінімізація затримки покращить пропускну здатність пристрою;
3) зменшення допоміжних входів, виходів - входів і виходів, які не реалізовані в конструкції шлюзів і служать лише для підтримки реверсивності пристрою - покращить проектний простір, необхідний для реалізації логіки.

Споживання електроенергії є досить важливим для смарт-карток. Існує три різні типи смарт-карток, які використовуються залежно від конкретного застосування: пам’яті смарт-карти: мікропроцесорні смарт-карти та криптоконтролери. Ці мікросхеми розрізняються на основі їх пам’яті, інтерфейсів пристрою зчитування та центральних процесорів. Смарт-карти пам’яті спеціально розроблені для того, щоб містити енергонезалежну електрично стираючу програмовану постійну пам’ять, а також спеціальну логіку для інтерфейсу введення/виведення запитуваних даних. Мікропроцесорні смарт-карти містять оперативну пам’ять (ОП), постійну пам’ять (ПП) і центральний процесор, який може виконувати певні операції на самому чіпі, надаючи йому більшу функціональність, ніж карта пам’яті. Криптоконтролери розширюють функціональні можливості карт пам’яті або мікропроцесорних карт, кодуючи та декодуючи збережені дані. Вони виконують обчислення ключів необхідної довжини за допомогою ефективних криптографічних процедур. Криптоконтролери використовують Ферро-електричну ОП (ФЕОП), оскільки запис даних у пам’ять може виконуватися набагато частіше зі зниженим енергоспоживанням, що є перевагою для безконтактних смарт-карт. Оскільки безконтактні смарт-картки надсилають сигнал PI, можливість надсилати швидший сигнал із меншим енергоспоживанням запобігає певним типам атак, роблячи сигнал більш безпечним. Стандарт ISO/IEC 7810 використовується для визначення фізичних характеристик ідентифікаційних карток.

Споживання електроенергії також є важливим фактором у стандартах кодування SIM-карт. Це навіть більш критично для мобільних пристроїв, ніж для звичайних портативних комп'ютерів. Оскільки, мобільні пристрої матимуть ще менше місця для акумуляторів і тому, корисність послуг, які вони надають, значно зменшиться, то люди не зможуть покладатися на їх роботу протягом довших періодів часу. Стандартом для запуску програм на основі Java на SIM-картах, який вирішив усі ці проблеми, є стандарт Java Card. Метою технології Java Card є підвищення безпеки SIM-карти, а також покращення переносимості SIM-карт між різними мобільними телефонами. Вони досягають цього шляхом визначення обчислювального середовища Java Card Platform, подібного до JVM у звичайних комп'ютерах, а також бібліотеки часу виконання. Дані в Java Card інкапсульовані в самій програмі, і програми запускаються в окремому апаратному просторі, ніж операційна система. Основна відмінність середовища Java Card VM від звичайних Java VM полягає в тому, що програми запускаються окремо, щоб обмежити доступ між програмами. Це важливий момент, оскільки певні програми повинні мати доступ лише до певних можливостей мобільного телефону. Як і більшість інших смарт-карт, стандарт Java Card використовує AES і алгоритм асиметричного ключа. Співпроцесор Java Card є прискорювачем віртуальної машини Java Card і реалізований апаратним способом. Для безконтактної картки IC необхідно розробити співпроцесор Java Card з низьким енергоспоживанням.

Мова Java Card, яка виконується на SIM-картах, є підмножиною стандартної мови Java і є повною програмою Turing. Мова Java Card досить портативна, тому її можна запускати на сумісній з Java системі та виконувати ті самі функції. Через обмежений простір Java Card не підтримує багато функцій і типів даних Java, таких як char, багатовимірні масиви або потоки. Подібним чином байт-код Java Card є підмножиною байт-коду Java 2, який використовується у віртуальній машині Java на звичайному комп'ютері. Подібно до мови Java Card, ці оптимізації виконуються через обмежені ресурси та простір.

1.2 Запобігання атакам зловмисників

Розширена логіка (РЛ) була запропонована в [39] і служить методологією динамічної та диференціальної логіки (ДДЛ). Ця реалізація використовує як комплементарні, так і некомплементарні сигнали в порядку шляхом попередньої зарядки вихідних вузлів, що дозволяє ІС генерувати диференціальні виходи до того, як атака зможе їх оцінити. Їхній метод був реалізований за технологією 0,18 мкм при 1,8 В, і було визначено, що нормалізована зміна потужності була в 116 разів меншою, ніж стандартна CMOS (SCMOS). Основним недоліком РЛ є те, що через погіршення сигналу важко каскадувати комірки. Можна розробити додаткові схеми для роботи з РЛ для покращення якості сигналу, але в результаті диференціальна потужність стає достатньою для успішної атаки ДАП [40].

Хвильова динамічна диференціальна логіка (ХДДЛ) була представлена в [41], яка використовує захищені складені вентилі та поєднує їх зі стандартними вентилями CMOS під час процесу автоматизації проектування, щоб імітувати поведінку вентилів РЛ зі зниженим енергоспоживанням. Цей метод дав зменшення коливань потужності на 52% за допомогою ХДДЛ порівняно з методом РЛ, але виграв майже вдвічі більше з точки зору середнього енергоспоживання, площі та часу перемикання.

Зменшена комплементарна динамічна та диференціальна логіка (ЗКДДЛ) була представлена в [42] як 42,29% покращення порівняно з ХДДЛ з точки зору енергоспоживання. Вони обидва використовують комплементарні логічні структури, але мають різну комутаційну ємність через властиву їм фізичну структуру. Побудована бібліотека ЗКДДЛ складається з комірок XOR і MUX у поєднанні з вентилем «ТА/АБО», реалізованим за допомогою логіки ХДДЛ, і двох буферів SCMOS із затримками поширення, що дорівнюють 0,3 нс і 0,5 нс.

Логіка захищеного диференціального мультиплексора з використанням транзисторів проходу (ЛЗДМт) була представлена в [43] і використовує мережі оцінки та попереднього розряду для впровадження комплементарної транзисторної логіки (КТЛ) у динамічну поведінку CMOS. Оцінна мережа

складається з PMOS-транзисторів, а мережа попереднього розряду складається з NMOS-транзисторів. Основний недолік використання лише КТЛ полягає в тому, що він задовольняє лише диференціальну вимогу для безпечної логіки ІС, але не задовольняє динамічну вимогу одного перемикання за цикл. Впроваджуючи мережу перед розрядом у ЛЗДМт, динамічні вимоги задовольняються, що дозволяє використовувати в дизайні Secure IC.

Попередні методи продемонстрували значне покращення диференціальної потужності порівняно зі стандартними схемами CMOS (SCMOS). Метод ЛЗДМт продемонстрував значне покращення завдяки використанню мережі оцінювання та розрядки, а також логіки подвійної шини. Однак усі вони покладаються на звичайні методи логіки з втратами для отримання необхідних динамічних логічних виходів.

Адіабатичний підхід був застосований у [44] для пом’якшення атак ДАП за допомогою логічного підходу AND/NAND. Їм вдалося отримати покращення на 47% у порівнянні з підходом RCCDL, але їх переміг ЛЗДМт. Інший недолік запропонованої ними конструкції полягає в тому, що вони використовують діоди у своїх елементах. Діоди фізично необоротні і розсіюють енергію кожного такту. Тому запропонований у [44] підхід не є справді адіабатичним або оборотним. Крім того, у [45] було представлено порівняння логічних структур «ТА» і NAND з 2 входами на частоті 13,56 МГц для всіх адіабатичних логік АЛ, 2N-2DL, ADCL та подвійної шини. Недоліком цих проектів є те, що логіка І та І-НЕ не може бути досягнута оборотно лише з двома входами, тому все ще існують великі диференціальні входи для певних входів через необхідне розсіювання для незворотних операцій. Це означає, що вхідні дані не є справді динамічними, і зловмисник усе одно може визначити входи однозначно з виходів. У [46] було запропоновано однофазну схему безпечного буфера для пом’якшення атак ДАП. Вони спробували обійти правило діодів, включивши два діоди, один на шляху оцінки, а інший на шляху розряду. Це стало проблемою, оскільки вони намагалися тактувати дизайн, оскільки вони продовжували додавати діоди для кожного тактового сигналу та кожного вихідного сигналу.

Багатофазні адіабатичні логічні структури були використані в [47] для покращення струму витоку та потужності, необхідних для виконання атаки ДАП. Дослідження в цій статті обмежувалося оцінкою фазованих інверторів і не розглядало динамічну логіку чи клітинки смарт-карт. У [48] адіабатичну логіку було використано для проектування пам'яті з адресацією вмісту в смарт-карті з метою продовження терміну служби батареї до 18 місяців. У [49] симетрична адіабатична логіка розподілу заряду була використана для розробки 8-розрядної схеми AES і покращила диференціальне енергоспоживання порівняно з раніше використовуваними адіабатичними підходами на коефіцієнт 53,17. А в [60] квазістатична частково адіабатична логіка з діодами була використана для представлення архітектури радіочастотної ідентифікації з 64-бітною тестовою схемою PI-міток у смарт-картці, яка потребувала менше схем, оскільки вони змогли усунути регулятор напруги та двополуперіодні схеми випрямляча.

Дроти та буфери CMOS розсіюють енергію, незважаючи на те, що обидва вони логічно оборотні. Звичайні CMOS-інвертори логічно оборотні, але вони не виробляють вхідний сигнал, якщо ви помістите еквівалентний вихідний сигнал на вихідний штифт. Поширення тактового сигналу вимагає ще більше енергії. Тому в будь-якій адіабатичній схемі CMOS необхідно вирішити дві проблеми. По-перше, реалізація повинна призвести до енергоефективної конструкції комбінованого джерела живлення та тактового генератора. По-друге, оборотні логічні функції вимагають більших логічних витрат, щоб задовольнити біективну вимогу [50]. Таким чином, енергію, що розсіюється під час перемикання ланцюга, необхідно контролювати та переробляти, а не розсіювати в навколишнє середовище.

Адіабатичні ланцюги використовують функції зміни, щоб мінімізувати енергію, що розсіюється через енергію, наскільки це можливо. По-перше, транзистор завжди включений, коли через нього протікає значний струм. По-друге, коли існує значна різниця між напругою витоку і стоку, транзистор повинен бути вимкнений. У [51] було показано, що адіабатичні схеми зменшують розсіювання енергії на 60% при 20 МГц і на 35% менше енергії при 100 МГц, а

реверсивні двошинні CMOS-транзистори були продемонстровані для роботи з низькою потужністю в [52].

Адіабатичні схеми CMOS вимагають здатності бути фізично однозначними, тобто вихідні сигнали можуть бути розміщені на виходах, а унікальні вхідні сигнали можуть бути відтворені на вхідних проводах. Для досягнення цієї мети в [34] був використаний підхід з подвійною рейкою, в якому три основні реверсивні логічні структури були розроблені та виготовлені за технологією 0,35 мкм, де Vtp=0,6В і Vtn=-0,6В. Схеми не мають входів джерела живлення, тобто вся енергія вихідних сигналів походить від вхідних сигналів.

Адіабатична динамічна логіка була вперше запропонована в [53] як метод адіабатичної реалізації динамічної логіки CMOS, щоб отримати порядок величини у зниженні потужності. Вони представили інвертор, що складається з NMOS-транзистора паралельно з діодом прямого зміщення з оцінкою та фазою попереднього заряду. Вони зменшили споживання електроенергії з 26 мкВт до 1,7 мкВт, використовуючи адіабатичний підхід на 100 МГц. Однак ці схеми не є справді адіабатичними, оскільки діоди не є оборотними за своєю природою [30].

Підхід із попереднім заряджанням і оцінкою з [54] було реалізовано в [55] як динамічна адіабатична MOS із хвильовою конвеєрною системою, що дозволяла 73% рециркуляції енергії на 200 МГц при виготовленні за технологією 0,25 мкм при напрузі живлення 2,5 В. Питання накладних витрат і тактування було розглянуто в [56] з реалізацією однофазного годинника. Вони отримали скорочення на 55% за допомогою технології 0,18 мкм із компромісом подвоєння часу роботи, хоча робота була значно скорочена порівняно з багатофазними схемами тактування. Це було ще більше вдосконалено в [57] за допомогою квазістатичної однофазної адіабатичної динамічної логіки (КОАДЛ). Завдяки використанню синусоїдального тактового сигналу, а також ступінчастої функції вони отримали 87% покращення споживання електроенергії порівняно зі статичною CMOS та 75% порівняно з однофазною адіабатичною схемою синхронізації. У [58] була використана адіабатична логіка, щоб продемонструвати, що родини адіабатичної логіки, такі як ЛВЗРР і КОАДЛ,

споживають постійну потужність під час етапу попереднього заряджання та оцінювання, що дозволяє пом’якшити атаки аналізу потужності.

У [59] було представлено резонансну динамічну логіку для використання індуктивно-ємнісної мережі для накопичення та відведення енергії в адіабатичній системі. Завдяки використанню цих мереж на частоті 500 МГц вони змогли досягти більш ніж 50% зниження розсіювання енергії.

Реалізація РДЛ в апаратному забезпеченні була запропонована в [60] для архітектури MINOS і, зокрема, спрямована на атаки на дані керування, які перезаписують адресу повернення в покажчиках пам’яті. Архітектура, представлена в [61], розглядала як контрольні, так і неконтрольні атаки, створюючи виняток, якщо тег індексу додається до незаплямованого вказівника з арифметичною інструкцією вказівника, і мала на меті покращити захист від переповнення буфера, наданий запобіжний стек [62] і стек-привід [63]. Це також дозволило багатозернові механізми для керування зберіганням тегів, що зменшило витрати пам’яті до менш ніж 2%. Цей підхід покращив хибні спрацьовування для винятків безпеки, оскільки багато програм використовують зв’язані перевірки для перевірки вхідних даних. Однак багато хибних негативів все ще існувало для звичайних атак.

Цю проблему було вирішено в [64] шляхом очищення тегу, коли зіпсовані дані порівнюються з незіпсованими, і не очищаючи вказівники на арифметику вказівників. РДЛ також було реалізовано для запобігання витоку інформації шляхом відстеження явного та неявного потоку інформації.

Головним недоліком впровадження РДЛ до цього моменту було те, що він не був гнучким і мав жорстко закодовану політику безпеки. MINOS не зміг протистояти будь-якій атаці на дані, яка не стосувалася конкретно механізму контролю архітектури. Архітектура в [64] створює багато помилкових позитивних і негативних результатів, оскільки передбачає, що порівняння здійснюють перевірку через перевірку меж. Програмні системи можуть також використовувати методи перевірки, такі як використання операції по модулю над

індексами хеш-таблиць або виконання логічної операції І над хеш-таблицями UID.

Розглянемо шифрування та алгоритм Rijndael. Алгоритми шифрування з відкритим ключем дозволяють передавати повідомлення між трансивером і вбудованим пристроєм за допомогою радіочастотної ідентифікації (РІ) за допомогою двох різних ключів для шифрування та дешифрування. Діффі та Хеллман розробили шифрування з відкритим ключем, щоб зменшити залежність від безпечних каналів розподілу ключів [65]. Вони досягли цього, вимагаючи від користувача отримати як відкритий, так і закритий ключ для шифрування та дешифрування повідомлень. Лише користувач володіє закритим ключем, тобто лише дійсний одержувач може розшифрувати повідомлення. Алгоритм Rijndael [66] був обраний у 2000 році Національним інститутом стандартів і технологій (НІСТ) як Advanced Encryption Standard [67]. Алгоритм Rijndael — це циклічний симетричний блочний шифр, який шифрує та розшифровує дані у 128-бітних блоках із ключем 128, 192 або 256 бітів, щоб забезпечити ефективний захист переданих і збережених даних від криптоаналітичних атак [68]. Застосовуючи чотири перетворення до звичайного тексту - початкове додавання ключа раунду, стандартні раунди та остаточний раунд, дані шифруються за допомогою відкритого ключа та розшифровуються за допомогою закритого ключа ефективним чином.

Алгоритм Rijndael [68] є прийнятим стандартом НІСТ для Advanced Encryption Standard (AES). Будь-яка архітектура Rijnadel повинна мати можливість реалізувати ітерований блоковий шифр зі змінною довжиною ключа/блоку з використанням початкових даних/додавання ключа, ряду стандартних раундів і остаточного раунду. Початкові входи в 128-бітну реалізацію алгоритму Rijndael складаються з блоку вхідного тексту, що складається з шістнадцяти 8-бітових станів, і ключа шифру, що складається з шістнадцяти 8-бітових блоків. Операція БайтСаб виконує мультиплікативне обернення поля Галуа $GF(2^8)$ з подальшим афінним перетворенням. Дешифрування виконує цей процес у зворотному порядку. Результатом є те, що

операція БайтСаб діє як 8-бітна таблиця пошуку, де вхідне значення однозначно корелює з вихідним значенням, і навпаки. Для 128-бітного шифрування відкритий текст розбивається на фрагменти розміром 16 байт. Чотири старші біти відносяться до «рядка» в таблиці удосконаленої схеми, а чотири молодші біти відносяться до «стовпця» в таблиці схеми. Результат використовується для заміни старого байта в ключі. Схема вважається найважливішим аспектом шифру AES щодо швидкості, розміру та пом’якшення атак ДАП.

1.3 Постановка задачі

Виробництво економічно ефективних безпечних інтегрованих мікросхем вимагає від розробників апаратного забезпечення врахування компромісів у розмірі, безпеці та енергоспоживанні. Для створення успішних проектів, орієнтованих на безпеку, апаратне забезпечення низького рівня повинно містити вбудовані механізми захисту, які доповнюють криптографічні алгоритми, такі як AES і Triple DES, запобігаючи атакам на бокових каналах, таким як диференціальний аналіз потужності (ДАП). Динамічна логіка затьмарює вихідні сигнали та роботу схеми, знижуючи ефективність атаки ДАП. Отже, метою роботи є реалізація адіабатичної динамічної диференціальної логіки для додатків у безпечному дизайні для пом’якшення атак ДАП.

Для досягнення мети роботи необхідно вирішити наступні завдання:

1. Провести аналіз відомих математичних моделей із оборотної логіки, розглянути важливі парадигми оборотної логіки – логічну та фізичну оборотність, розглянути допустимість послідовної логіки в оборотних обчислювальних системах.

2. Розробити набір обчислювальних логічних примітивів, щоб забезпечити підтримку фізичної оборотності для всіх оборотних логічних структур із шляхами зворотного зв’язку.

3. Розробити найефективніший підхід до запобігання атакам ДАП, щоб ускладнити для зловмисника визначення необхідної інформації для визначення вхідних даних.

4. Запропонувати реалізацію оборотної логіки в CMOS та SRAM, розробити алгоритм синтезу двошинної адіабатичної логіки.

5. Представити удосконалену адіабатичну схему з двома шинами для пом’якшення атак ДАП та провести експериментальні дослідження для перевірки ефективності запропонованого вдосконалення методу забезпечення безпеки програм згідно оборотної логіки.

Для досягнення поставлених завдань використані основні положення:

1) адіабатичної та оборотної логіки;
2) математичного моделювання та матричних перетворень для квантових обчислень для систем з низьким енергоспоживанням для доказу того, що послідовні оборотні логічні структури фізично можливі;
3) забезпечення безпеки програм згідно адіабатичної динамічної диференціальної логіки для додатків у безпечному дизайні для пом’якшення атак ДАП.

1.4 Висновок

В першому розділі роботи були розглянуті два основні джерела теоретичних дебатів в адіабатичній і оборотній логіці. Розглянуто питання про те, чи можна маніпулювати схемами перемикання потоку електронів оборотно за допомогою логічних структур CMOS. Представлені результати симуляції прикладу адіабатичної логіки, де бінарна комутаційна мережа розсіює менше kTln(2) джоулів енергії за подію комутації. Також, розглянуто допустимість послідовної логіки в оборотних обчислювальних системах. Запропоновано математичне обгрунтування того, що послідовні оборотні логічні структури фізично можливі.

Також, в роботі приділено увагу на старт-картки. Смарт-картками називають невеликі інтегровані схеми, вбудовані в пластик або жетони, які використовуються для автентифікації, ідентифікації та зберігання персональних даних. Вони використовуються військовими, у банкоматах, SIM-картках мобільних телефонів, школами для відстеження відвідування уроків і зберігання сертифікатів для безпечного перегляду веб-сторінок. Вони також використовуються на міжнародному рівні як альтернатива кредитним і дебетовим карткам Europay, MasterCard і Visa. Вони залежать від конкретної програми, тому їх розмір і програмне забезпечення можуть бути мінімізовані. Їх можна запрограмувати для запобігання крадіжці шляхом запобігання негайному повторному використанню, що робить їх ефективнішими, ніж картки з магнітними смужками. Технологія смарт-карт просувається в бік програм вищої сумісності та кількох інтерфейсів, таких як TCP/IP, NFC і безконтактні чіпи.

Також, в розділі приділено увагу використанню енергоспоживання для отримання компрометуючої інформації відоме як атака диференціального аналізу потужності (ДАП). Зловмисник аналізує інформацію, отриману з деталей практичної реалізації безпечних алгоритмів. Більшість сучасних обчислювальних систем використовують технологію CMOS, і динамічне споживання енергії вентилем CMOS пропорційне його вхідним сигналам. Таким чином, аналіз вихідної потужності дозволяє зловмиснику визначити кореляцію між даними та ключем, оскільки перемикання в вентилях CMOS залежить від цих вхідних даних. Коли зловмиснику відомий відкритий текст і круглий підключ, він може визначити вхідні дані для логічної функції та вивести їх вихід за допомогою таблиці пошуку. Алгоритми відкритого ключа можна аналізувати за допомогою ДАП шляхом співвіднесення значень кандидатів для проміжних обчислень із вимірюванням енергоспоживання. Для операцій модульного піднесення до степеня можна перевірити припущення бітів експоненти, перевіривши, чи співвідносяться прогнозовані проміжні значення з фактичним обчисленням.

Отже, з метою забезпечення безпеки програм, перспективним і актуальним є удосконалення та розроблення методів згідно оборотної логіки.

2 ДОПУСТИМІСТЬ ПОСЛІДОВНОЇ ЛОГІКИ В ОБОРОТНИХ ОБЧИСЛЮВАЛЬНИХ СИСТЕМАХ

2.1 Послідовні обчислювальні структури та моделювання оборотних систем

Послідовні обчислювальні структури — це ефективні недорогі пристрої, які використовують шляхи зворотного зв’язку для повторного використання підпрограм, щоб пристрій міг оновити інформацію, що зберігається, перш ніж вона буде втрачена. Існує дискусія щодо того, чи викликає зворотний зв'язок стирання бітів. Це завадить комп’ютеру відстежити всі свої обчислення, унеможливлюючи розробку квантової машини Тьюринга зі зворотним зв’язком. У багатьох текстах і статтях рішуче стверджується, що зворотній зв'язок неприпустимий у структурі квантових обчислень ні за яких обставин. Інші кажуть, що це «загальне неправильне сприйняття» [30]. Стаття, що підтримує обидві точки зору, була опублікована протягом останнього часу [69]. Встановлення законів і показників для цих пристроїв має важливе значення для визначення природних меж для будь-якої обчислювальної технології.

Для того, щоб квантова оборотна обчислювальна машина задовольняла обмеження, встановлені Беннетом, пристрій має бути повністю оборотним у часі, що означає, що – враховуючи залежний від часу набір виходів – схема зможе відтворювати вхідні дані, керуючи ланцюгом в зворотному порядку. Фредкін був першим, хто підтримав ідею послідовних оборотних схем [69]. У його підході існує миттєвий комбінаційний елемент і провід зворотного зв’язку, який має вбудований елемент затримки, щоб забезпечити збереження ітераційного характеру оборотної структури в часі. З цією метою він представив систему, що складається з вентиля Фредкіна та інвертора на шляху зворотного зв’язку, щоб продемонструвати, що будь-які обчислення, які можуть бути виконані звичайною мережею, також можуть бути виконані відповідною консервативною логічною мережею, за умови, що є зовнішня подача констант і зовнішній злив для сміття. Тоффолі представив структуру для згущення мережі з ітераціями в часі в

реверсивну послідовну мережу шляхом багаторазового використання тієї самої операції та введення елемента затримки, щоб забезпечити збереження залежності від часу на етапах [1]. Він дійшов висновку, що будь-які операції, які можуть бути обчислені довільним скінченним автоматом, також можуть бути обчислені за допомогою оборотного скінченного автомата. Також, Дойч представив 4-бітну універсальну логічну структуру [70], ідентифікуючи початковий стан шляху зворотного зв'язку як вихідний біт, і стверджував, що він є послідовним та оборотним за своєю природою, «незважаючи на одиничний дріт, який петляє назад». Однак більшість досліджень оборотного логічного синтезу базуються на уявленні про те, що послідовні оборотні логічні структури неможливі. Вони стверджують, що «зворотний зв'язок не дозволяється» і що графічні представлення оборотних логічних структур мають бути спрямованими ациклічними графами. Вони стверджують, що цикли не дозволені, оскільки схеми, що містять цикли, можуть стати нестабільними, що призведе до втрати інформації, забороняючи їх використання. Ті, хто підтримує послідовні схеми в оборотній логіці, будуть стверджувати, що оскільки схема повністю складається з оборотних примітивів, то вона повинна бути оборотною. Однак і це не зовсім так. Причина відмови від послідовних схем, яка часто згадується, походить із [71], де показано приклад інвертора зі зворотним зв'язком без будь-яких затримок, а також розгортання та розведення на вході та виході. Інвертори є логічно реверсивними, але без контрольованого шляху зворотного зв'язку вхідний сигнал інвертора зміниться одразу після зміни вихідного сигналу. Це порушує властивості оборотної машини Тьюрінга, встановлені Беннетом, оскільки вхідні дані змінюються раніше, ніж інтервал часу, необхідний інвертору для оновлення сигналу. Це створює умови змагання, і оскільки пам'ять реверсивної машини Тьюрінга оновлюється на кожному такті, результат змінюється набагато швидше, ніж годинник оновлює пам'ять, що призводить до втрати інформації. При цьому порушується оборотність. Таким чином, можна синтезувати послідовну логічну структуру, чий комбінаційний логічний блок є логічно оборотним, але пристрій не є фізично оборотним.

В літературі існують два відомі погляди: одна сторона стверджує, що послідовні схеми допустимі за будь-яких обставин; інша наполягає на тому, що це взагалі неприпустимо. Спираючись на роботу [72] я непогоджуюсь з твердженням про те, що зворотний зв’язок не допускається в оборотних логічних структурах за будь-яких обставин. Використовуючи елемент затримки, представлений Тоффолі та Фредкіном, і розглядаючи початковий стан контуру зворотного зв’язку як вихідний біт «0», як запропоновано Дойчом, було математично продемонстровано, що послідовна оборотна логічна схема може бути розроблена. Її операції можуть бути повністю протилежними, що дозволяє використовувати їх у реверсивних машинах Тьюрінга. Враховуючи це, я вдосконалив модель для врахування різниці між найгіршою затримкою в ланцюзі та затримкою на шляху зворотного зв’язку, щоб гарантувати, що сигнал на залежному від зворотного зв’язку вході надходить одночасно з іншими входами.

Отже, будемо визначати послідовну оборотну логічну структуру як будь-який логічний пристрій, де один або більше входів у структурі залежать від логічних обчислень одного чи кількох виходів, а решта структури є оборотною. Залежний від зворотного зв’язку вхід – це вхід, який отримує дріт зворотного зв’язку. Вихід, що створює зворотний зв'язок, - це вихід, з якого походить дріт зворотного зв'язку. А шлях зворотного зв’язку – це провід, який починається на виході, що створює зворотний зв’язок, і закінчується на вході, що залежить від зворотного зв’язку. Щоб належним чином застосувати принципи квантової механіки до цих структур, я повинен визначити зв’язки між шляхами зворотного зв’язку, вихідними сигналами, що створюють зворотний зв’язок, і вхідними сигналами, залежними від зворотного зв’язку. Оскільки послідовна реверсивна логічна структура повинна мати однакову кількість вхідних і вихідних ліній, розгортання та розведення не допускаються. Це означає, що послідовна реверсивна логічна структура повинна мати однакову кількість n залежних від зворотного зв'язку входів і зворотних виходів. Для того, щоб послідовна реверсивна логічна структура мала більше вхідних сигналів, що залежать від

зворотного зв’язку, ніж вихідних сигналів, що створюють зворотний зв’язок, принаймні один із цих виходів має бути продубльований, що вимагає розбиття.

Для того, щоб структура мала більше вихідних сигналів (які генерують зворотний зв’язок) ніж входів, що залежать від зворотного зв’язку, необхідно, щоб принаймні один із цих входів залежав від двох виходів. Це потребує наявності вентилятора.

Оскільки між кожним входом і виходом існує один шлях, кількість шляхів зворотного зв’язку має дорівнювати n. У початковому циклі тактування шляху зворотного зв’язку діє так само, як вихідний біт, оскільки значення квантового біта буде створено на виході. Вихідний біт – це вентиль, який на кожному кроці обчислення видає на вихід значення «0» або «1» [70]. Вихідні біти визначаються як оборотні, оскільки існує бієкція між виробленим значенням на вході вентиля та виробленим виходом. Тому в початковому такті зворотний зв'язок є фізично оборотним. Для кожного наступного тактового циклу шлях зворотного зв’язку діє як провід, виробляючи значення з виходу, що залежить від зворотного зв’язку, на вхід, що створює зворотний зв’язок. Оскільки дроти є оборотними за своєю природою, шлях зворотного зв’язку є оборотним для кожного наступного тактового циклу, що робить шлях зворотного зв’язку повністю оборотним.

Тепер розглянемо математичну модель процесу. Під час початкового тактового циклу будь-якого послідовного оборотного логічного пристрою втрата одного вхідного стану на d залежних входів призведе до втрати 2^d потенційних вхідних станів. Крім того, кожен допоміжний вхід зменшить загальну кількість можливих станів на 2^a, оскільки виходи, можливі, коли a є протилежним значенням, ніколи не будуть досягнуті. Це призводить до втрати потенційних вхідних станів 2^{d+a}. Оскільки пристрій є оборотним, а вихідний біт, що створює значення для допоміжних входів, є оборотним, то бієкція призведе до втрати 2^{p+a} потенційних вихідних станів (тут p – кількість вихідних сигналів, що генерують зворотний зв’язок). Оскільки вхідний і вихідний стани однозначні, початкове значення кожного залежного входу можна однозначно визначити. Отже, оскільки

d = p, загальна кількість можливих вхідних станів і вихідних станів еквівалентні і дорівнюють величині 2^{N-d+a}.

Протягом кожного наступного тактового циклу кожен вихідний сигнал, що створює зворотний зв'язок, може бути в послідовному оборотному логічному пристрої, де зворотний зв'язок створюватиме «0» або «1», або підтримуватиме одне й те саме значення для кожного тактового циклу. Отже, потенційна кількість станів вихідного сигналу, що створює зворотний зв'язок, S_i, дорівнює 2 у першому випадку та 1 у другому випадку. Це означає, що кількість втрачених потенційних вихідних станів для всіх вихідних сигналів, що створюють зворотний зв'язок, становить $2^{\sum_{i=1}^{p} 2-s_i}$. Втрачені потенційні стани виникають лише на виходах, що створюють зворотний зв'язок, які потім виробляються на залежних від зворотного зв'язку входах для обчислення в наступному тактовому циклі. Решта пристрою є оборотною, що означає, що загальна кількість можливих станів входу та станів виходу для наступного такту еквівалентна і дорівнює $2^{N-a\sum_{i=1}^{p} 2-s_i}$.

Це означає, що для кожного такту можна обчислити загальну кількість вхідних і вихідних станів. Крім того, бієкція між вхідними станами та вихідними станами пристрою дозволяє визначити попередні вхідні стани та наступні вихідні стани. Також, оскільки допоміжні входи реалізовані з оборотними вихідними бітами, то використання допоміжних входів не призводить до збільшення ентропії. Таким чином, вхідний стан кожного тактового циклу послідовного оборотного логічного пристрою може бути однозначно визначений шляхом спостереження за вихідними значеннями.

Я зробив висновок про минулі стани послідовних оборотних логічних структур, що означає, що я маю достатньо інформації, щоб відстежити попередні обчислення, що дозволяє використовувати їх на завершальній стадії оборотної машини Тьюринга. Однак, все одно потрібно визначити, чи пристрій фізично оборотний. Для початкового тактового циклу можлива кількість вхідних станів дорівнює $W_0 = 2^{N-d-a}$, а загальна кількість можливих вихідних станів дорівнює

$W_f = 2^{N-p-a}$. Оскільки пристрій є біективним, ймовірність досягнення кожного вхідного стану дорівнює $\frac{1}{2^{N-d-a}}$, а ймовірність кожного вихідного стану дорівнює $\frac{1}{2^{N-p-a}}$.

Використовуючи співвідношення ймовірностей для зміни ентропії в обчислювальній системі, отримуємо наступне:

$$\int \frac{dQ}{T} = k\left(2^{N-p-a} * \frac{1}{2^{N-p-a}} * \mathrm{In}\left(\frac{1}{2^{N-p-a}}.\right)\right) -$$
$$-k\left(2^{N-d-a} * \frac{1}{2^{N-d-a}} * In\left(\frac{1}{2^{N-d-a}}\right)\right), \quad (2.1)$$
$$\int \frac{dQ}{T} = (d-p)k_B * In(2) = 0$$

де Q – кількість тепла;

T – температура;

p – кількість вихідних сигналів, що генерують зворотний зв'язок;

d – кількість залежних входів;

a – кількість можливих входів;

N – кількість виходів;

k_B – коефіцієнт відповідності.

Оскільки я знаю, що кількість входів, залежних від зворотного зв'язку, дорівнює кількості виходів, що створюють зворотний зв'язок, рівняння оцінюється як $\int \frac{dQ}{T} = 0$, що робить початковий тактовий цикл фізично оборотним. Зміна ентропії для кожного наступного такту в послідовній оборотній логічній структурі визначається встановленням кількості можливих вхідних станів на $W_0 = 2^{N-a-\sum_{i=1}^{d} 2-S_i}$ і кількості можливих вихідних станів $W_f = 2^{N-a-\sum_{k=1}^{p} 2-S_i}$.

Це дає рівняння для визначення статистичної ймовірності кожного вхідного та вихідного стану для кожного наступного тактового циклу: $\int \frac{dQ}{T} = k_B In(2)\left(\sum_{i=1}^{d} 2 - S_i - \sum_{k=1}^{p} 2 - S_k\right)$.

Відомо, що кількість шляхів зворотного зв'язку дорівнює як d, так і p, і що кожен шлях зворотного зв'язку є оборотним. Таким чином, значення і для кожного вхідного сигналу, що залежить від зворотного зв’язку, має бути еквівалентним значенню k відповідного вихідного сигналу, що створює зворотний зв’язок. Це означає: $\sum_{i=1}^{d} 2 - s_i = \sum_{k=1}^{p} 2 - s_k$. Отже, рівняння ентропії зводиться до $\int \frac{dQ}{T} = 0$. Це означає, що послідовні оборотні логічні структури, як я їх визначив, фізично оборотні в усіх випадках.

Отже, розглянемо вентиль Фредкіна з послідовним контуром зворотного зв’язку. Цей пристрій має 3 входи та 3 виходи, з одним входом, що залежить від зворотного зв’язку, і одним виходом, що створює зворотний зв’язок. Крім того, три входи встановлюються на «0», тому пристрій має 0 допоміжних входів без вихідних бітів. Я визначаю кількість потенційних станів введення як 2^{N-d-a}. Оскільки: $N = 3$, $d = 1$ і $a = 0$, то отримуємо $2^{3-1-0} = 4$. Отже є лише 4 можливі стани введення та виведення у першому тактового циклу, пристрою. Оскільки існує бієкція, коли вимірюється вихід пристрою в кінці першого тактового циклу, початкове значення залежного входу може бути однозначно визначено.

Наприкінці початкового тактового циклу вхід, що створює зворотний зв’язок, може бути «0» або «1» у будь-якому випадку. Таким чином, вихідне значення s_1 дорівнює 2. Можлива кількість вхідних станів визначається шляхом підстановки $N = 3$, $d = 1$, $a = 0$ і $s_1 = 2$, що дає 8 можливих станів для кожного наступного такту. Тому $\sum_{i=1}^{1} 2 - s_i = \sum_{k=1}^{1} 2 - s_k$ для всіх наступних тактів. Отримаємо: $dQ/\mathrm{T} = k_B \ln(2) \left(\sum_{i=1}^{d} 2 - s_i - \sum_{k=1}^{p} 2 - s_k\right)$.

Оскільки приріст ентропії для кожного такту дорівнює 0, представлений структурний пристрій є повністю оборотним. Покажемо, що вхідні значення можна відтворити на основі вхідних даних. Далі розглянемо набір входів, протягом 12 тактів. Значення є входами для сигналів A та B. При першому часовому циклі входи A та B приймають значення <0 0>, при другому часовому циклі входи A та B приймають значення <0 1>, при третьому часовому циклі входи A та B приймають значення <1 1>, при четвертому часовому циклі входи A

та B приймають значення <1 0>, при п’ятому – <1 1>, при шостому – <1 0>, при сьомому – <1 0>, при восьмому – <0 1>, при дев’ятому – <0 0>, при десятому – <0 1>, при одинадцятому – <1 0>, при дванадцятому – <1 1>.

При t=0 вхід, залежний від зворотного зв’язку, розглядається як біт сигналу зі значенням «0». Тому результати першого введення дають <0 0 0>, а значення «0» повертається на вхід. Це дає три входи <010> для t=2. Це дає <010> для виводу. Для t=3 початкові вхідні значення <110>, що дає вихід <101>. Значення «1» розміщується на шляху зворотного зв’язку, що дає початкове значення <101> для виходів. Вихідні значення, отримані на початкових виходах, і значення, надане на шляху зворотного зв’язку *f* наступні: входи P, Q та зворотній шлях *f* протягом першого часового такту приймають значення <0 0 0>, протягом другого часового такту – <0 1 0>, протягом третього часового такту – <1 0 1>, протягом четвертого часового такту – <1 1 0>, п’ятого – <1 0 1>, шостого – <1 1 0>, сьомого – <1 0 0>, восьмого – <0 1 0>, дев’ятого – <0 0 0>, десятого – <0 1 0>, одинадцятого – <1 0 0>, дванадцятого – <1 0 1>.

Щоб показати оборотність, розмістимо значення t = 4 на виходах пристрою та запустимо у зворотному напрямку. Отримаємо наступні вхідні дані: при першому часовому циклі входи A та B приймають значення <1 0>, при другому часовому циклі входи A та B приймають значення <1 0>, при третьому часовому циклі входи A та B приймають значення <0 1>, при четвертому часовому циклі входи A та B приймають значення <0 0>, при п’ятому – <0 1>, при шостому – <1 0>, при сьомому – <1 0>, при восьмому – <1 1>, при дев’ятому – <1 1>, при десятому – <1 0>, при одинадцятому – <0 1>, при дванадцятому – <0 0>.

При t=0 вхід, залежний від зворотного зв’язку, розглядається як біт сигналу зі значенням «0». Таким чином, результатом першого введення є <1 1 0>, а значення «0» повертається на вхід. Це те, що очікувалося, оскільки при початковому вході в схему остання ітерація <1 1> і «0» було значенням на вході. Виконуючи зворотні обчислення, показано, що значення A і B є зворотними. Таким чином, оборотність досягнута.

2.2 Пропозиція щодо покращення CMOS та SRAM

Для моделювання представимо комбінаційну реверсивну логічну структуру, яка використовує мережу PMOS/NMOS, керовану системним входом +Vo, -Vo та 0V. Це вихідний біт, який вперше було заявлено як оборотний у [70], й ідентичний експерименту, проведеному в [75]. Різниця полягає в тому, що замість мережі RC використовується мережа CMOS. Дана схема працюватиме під пороговою напругою. Вона виконує два оборотні обчислення: «Копіювати S до M» і «Стирати за допомогою копії». Ці операції порівнювалися з незворотною операцією «Стерти без копіювання». Енергія, яку утримує біт ($E_{\text{біт}}$), визначається зарядом вихідного конденсатора та визначається за допомогою рівняння $E_{\text{біт}} = \frac{1}{2} * C_I V_0^2$, де $C_I -$ вихідна ємність.

При даному моделюванні по-перше, перемикач переміщується з 0V на +V0 і вимірюється для заданого часу наростання для схеми. Кожен часовий відрізок для всіх операцій у цьому моделюванні ідентичний. Це імітує «Копіювати S до M ’1’».

Після іншого відрізку часу перемикач переміщується в положення 0V, що імітує «Стерти «1».

Після іншого відрізку часу перемикач переміщується з 0V на -V0, що імітує «Копіювати S до M ’0’». Знову, перемикач повертається на 0V, щоб імітувати «Стерти 0». Нарешті, перемикач переміщується в положення петлі, дозволяючи вимірювати наступний тактовий цикл на основі сигналу з вихідного сигналу зворотного зв’язку.

Для того, щоб схема CMOS працювала поза межами шуму та зберігала надійність, я повинен порівняти енергію, що розсіюється схемою з енергією, що зберігається в конденсаторі. Енергія, що зберігається в конденсаторі, представлена (2.6). У всіх симуляціях використано вихідну ємність $C_I = 5fF$. Схема з напругою живлення 2мВ буде зберігати $1*10^{-20}$Джоулів.

Представлені результати моделювання були отримані з використанням малопотужної моделі прогнозної технології, розробленої в [31]. Моделювання проводилося для схем з напругою живлення в діапазоні від 1 В до 500В. Було введено ряд обмежень на поняття успішної роботи схеми. Оскільки потрібно було показати, що схема працює належним чином поза межами шуму, то «успішною операцією» вважатимемо таку, коли вихідна напруга досягла 95% напруги живлення під час усіх операцій копіювання та зберігала не більше 5% напруги після операцій стирання.

Перша успішна змодельована схема була з +Vo=1,4 мВ і часом перемикання 60 мкс. Вдосконалена CMOS має напругу живлення у 28 разів більшу, ніж схема RC суб-Ландауера в [75], і час перемикання в 10,67 разів швидший при досягненні тієї самої операції. Ця схема має напругу живлення 1 мВ і час перемикання 250 мкс, що означає, що напруга живлення в 20 разів вища, а час перемикання в 2,56 рази швидше, ніж експеримент RC, представлений Снайдером.

Даний пристрій використовує переваги рециркуляції енергії в CMOS, щоб створити локально оборотну схему з відповідними кодами виправлення помилок. Енергія переробляється, керуючи потоком струму через ланцюг.

Під час операції копіювання «1» у ланцюзі підтримується позитивний струм. Під час операцій «Стерти 1» і «Копіювати 0» існує негативний струм, а «Стерти 0» має позитивний струм. Це результат належного підключення пристрою пам’яті до відповідного джерела напруги з метою поступового підвищення напруги. Це дозволяє системі належним чином вловлювати всю розсіювану енергію.

Щоб продемонструвати, чому це важливий фактор як для шуму, так і для реверсивної роботи, було змодельовано роботу схеми CMOS, яка не враховує відповідні коди виправлення помилок. Ця схема виконує операцію «Стерти без копіювання». Операція зберігає значення в пам’яті, а потім стирає, розсіюючи значення в пам’яті. Це як логічно, так і фізично необоротно. Це пов’язано з тим, що значення пам’яті після «Стерти без копіювання» ґрунтується, незалежно від попереднього значення. Це фізично необоротно, оскільки пам'ять не підключено

до відповідного джерела напруги, що скасовує будь-які коди виправлення помилок. Це означає, що конденсатор потрібно розрядити, щоб стерти біт. Таким чином, копія біта не робиться, а енергія розсіюється в навколишнє середовище. Це означає, що порушується і фізична оборотність.

Кількість енергії, що розсіюється під час цієї операції, еквівалентна $\frac{1}{2}CV^2$. Це значення дорівнює $5*10^{-21}$ Дж, що в 1,77 разів перевищує бар'єр Ландауера.

Для удосконалення SRAM, пропоную наступне: перші два входи мають бути сигналами читання та запису, а наступні два входи мають бути допоміжними входами, де обидва утримуються на 0. Причому вентиль Фейнмана додається до виходу для відновлення значення запису.

Тоді реверсивна комірка SRAM буде забезпечувати таку ж вихідну функціональність як і звичайна комірка CMOS 6T SRAM [11].

Щоб реалізувати масив комірок SRAM, необхідний декодер 2-до-4. Таким декодером може бути реверсивний вентиль 4*4 RD. Логічна конфігурація, якого наступна: перший вхідний сигнал «0» на виході дає $\overline{(\mathrm{X0}\oplus\mathrm{X1})\oplus(\mathrm{X0X1})} = \overline{\mathrm{X0X1}}$; другий вхідний сигнал X1 дає $\mathrm{X1}\oplus(\mathrm{X0X1}) = \overline{\mathrm{X0}}\mathrm{X1}$; третій вхідний сигнал X0 дає на виході $\mathrm{X0}\oplus(\mathrm{X0X1}) = \mathrm{X0}\overline{\mathrm{X1}}$; четвертий вхідний сигнал «0» на виході дає X0X1.

Налаштований шлюз RD використовується для трансляції вхідної адреси та вибору відповідних комірок SRAM для читання/запису в масиві 4x2 SRAM. Далі однопортову SRAM потрібно буде модифікувати для створення оборотної комірки з подвійним портом SRAM (двопортовий SRAM).

Після цього реверсивний двопортовий SRAM має бути налаштований у вигляді реалізації n-розрядного синхронного двопортового масиву SRAM. Витрати цієї конструкції становлять $42n + \sum_{i=1}^{n}(i-1)$, а затримка становить (19+n).

Логічна конфігурація реверсивної комірки DRAM має наступну логіку керування, що використовує модифіковані затвори Переса. Конструкція має використовувати два вентиля RD, які слугують декодерами 2-до-4 для сигналів

вибору рядків і стовпців. Тут сигнал запису проходить через контрольний вентиль Переса (КВП), що складається з двох вентилів Переса.

У представленій конструкції SRAM мають використовуватися два вентилі Фредкіна, кожен з яких несе квантову вартість і затримку в 5 одиниць. Оскільки реверсивний пристрій SRAM не використовує третій вихід жодного вентиля Фредкіна, то обидва були виходами для відходів Для цієї конструкції, з огляду квантової фізики, краще використовувати вентиль RMUX.

2.3 Висновок

Послідовні обчислювальні структури дозволяють використовувати повторне використання підпрограм, що дозволяє скорочувати, прості та ефективні конструкції обчислювальних пристроїв. Реверсивна логіка дозволяє розробити універсальну обчислювальну машину, де кожна вхідна характеристика однозначно визначається результатами, що друкуються вихідним станом у кожному тактовому циклі, і система досягає всіх можливих станів, що призводить до відсутності розсіювання тепла.

У цьому розділі продемонстровано, що існує бієкція між вхідними сигналами, що залежать від зворотного зв’язку, і виходами, що створюють зворотний зв’язок, у послідовному оборотному логічному пристрої. Потім, використовуючи рівняння ентропії, обчислено загальну кількість можливих вхідних станів і вихідних станів для початкового такту, а також кожного наступного такту. У результаті, пристрій виявився однозначним на кожному такті.

Розроблено набір обчислювальних логічних примітивів, щоб забезпечити підтримку фізичної оборотності для всіх оборотних логічних структур із шляхами зворотного зв’язку.

Використовуючи рівняння Больцмана, можна зробити висновок, що послідовна оборотна логічна структура є фізично оборотною для кожного такту. Таким чином, я підтримую точку зору, що зворотний зв’язок дозволений у конструкціях послідовних обчислювальних структур на основі оборотної логіки.

Було запропоновано модифікацію реверсивної комірки SRAM за допомогою воріт RMUX для покращення квантових характеристик пристрою. Також, представлені результати симуляції прикладу адіабатичної логіки, де бінарна комутаційна мережа розсіює менше kTln(2) джоулів енергії за подію комутації. Показано, що енергія біта більша, ніж kTln(2), але результуюча дисипація менша, ніж kTln(2).

3 КОМУТАЦІЯ ФУНКЦІЙ, ЩО ЗМЕНШУЄ ПОТУЖНІСТЬ ЕНЕРГІЇ

3.1 Моделювання оборотних логічних структур

Розглянемо модель поведінки для фундаментальних інтегрованих шлюзів Кубіт (ІК) для проектування локально оборотних логічних структур. Моделювання шлюзів ІК, на відміну від шлюзів Контроль-V або шлюзів Тоффолі, дозволяє створити більш надійну модель, яка точніше відображає теоретичну оборотну обчислювальну структуру. Цей метод є розширенням існуючої мови програмування та методу моделювання, що дозволяє проектувати, моделювати та перевіряти оборотні логічні структури.

Кубіт – це бібліотека, що є надійною багатозначною логічною системою для визначення заданих детермінованих квантових станів і узагальнення недетермінованих станів.

Метою оборотного проектування є:

1) мінімізація квантової вартості – кількості оборотних обчислень 1*1 і 2*2, необхідних для створення логічного виводу – зменшить обчислювальну складність пристрою;
2) мінімізація затримки – логічної глибини пристрою – покращить пропускну здатність пристрою;
3) зменшення допоміжних входів і сміттєвих виходів – входів і виходів, які не реалізовані в конструкції шлюзу і служать лише для підтримки оборотності пристрою – покращить простір дизайну, необхідний для реалізації логіки.

Представимо матричні операції, необхідні для повної реалізації локально оборотної машини Тьюринга. У [73] Ді Вінченцо надав доказ того, що універсальну квантову схему можна повністю сконструювати, використовуючи лише 2x2 оборотні вентилі. Це стало важливою спробою після того, як Шор представив свої алгоритми для знаходження дискретних логарифмів і розкладання цілих чисел [76]. Оскільки закони фізики допускають лише унітарні

перетворення, детерміновані обчислення можна виконати на квантовому комп'ютері тоді і тільки тоді, коли вони оборотні.

Представлення станів у двовимірному гільбертовому просторі знайдено за допомогою комплексної проективної лінії, яка є геометричною сферою, відомою як сфера Блоха, яка містить точки на краю сфери, які відповідають взаємно ортогональним векторам стану. Точки на сфері представляють стани системи, а полюси сфери представляють стани «розкручування вгору» і «розкручування вниз». Важливо відзначити, що гранична умова на гільбертовому просторі полягає в тому, що тільки стани на поверхні сфери є «чистими» станами, тоді як будь-який стан поза поверхнею є змішаним станом [77]. Тому будь-який оборотний обчислювальний дизайн обмежений операціями, які досягають станів на поверхні сфери Блоха, в іншому випадку буде втрата інформації. Представлення стану спіна електрона на блоховській сфері задається $|\varphi >= \cos(\theta/2)\big|0> + e^{i\varphi}\sin(\theta/2)\big|1>$, де $\varphi, \theta -$ кути сферичної системи координат.

Це може бути представлено обертанням через x та y, а також уже представленими матрицями заперечення та тотожності.

Матриця «ЗАПЕРЕЧЕННЯ», яка може бути застосована до однокубітового сигналу для інвертування результату, так що $\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix} * \begin{bmatrix}a_0\\ a_1\end{bmatrix} = \begin{bmatrix}a_1\\ a_0\end{bmatrix}$.

Використовуючи унітарні оператори, вперше представлені Дойшем, ми можемо точно представити оператор «ЗАПЕРЕЧЕННЯ» в просторі Гільберта, використовуючи два оператори Кубіт в послідовності для представлення їхніх половинних спінів, де $\varphi = \pi/2$. Таким чином, матриця «ЗАПЕРЕЧЕННЯ» правильно представлена її матрицею квадратних коренів, $\frac{1+i}{2}\begin{pmatrix}1 & -i\\ -i & 1\end{pmatrix}$.

$$R_y(\theta) = \begin{pmatrix}\cos\left(\frac{\theta}{2}\right) & \sin\left(\frac{\theta}{2}\right)\\ -\sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right)\end{pmatrix} R_z(\varphi) = \begin{pmatrix}e^{i\varphi} & 0\\ 0 & e^{-i\varphi}\end{pmatrix} \qquad (3.1)$$

Цей оператор відомий як вентиль V – або квадратний корінь із заперечення – і послідовне виконання двох вентилів V призведе до матриці «ЗАПЕРЕЧЕННЯ»

$$\frac{1+i}{2}\begin{pmatrix}1 & -i\\ -i & 1\end{pmatrix} * \frac{1+i}{2}\begin{pmatrix}1 & -i\\ -i & 1\end{pmatrix} = \begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}. \quad (3.2)$$

Подібно до вентиля V, виконання двох операцій V+ створить матрицю «НЕ». Оскільки вентилі V і V+ є ермітовими спряженими, результатом є те, що вони створюватимуть одиничну матрицю, яка виконуватиметься послідовно.

Це дозволяє розглядати вентилі 2x2 Контроль-V і Контроль-V+ для використання в оборотних квантових обчисленнях. Коли керуючий сигнал дорівнює нулю, вхідне значення узгоджується на виході. Коли керуючий сигнал дорівнює «1», тоді виконується унітарна V-операція, як і з вентилями V і V+, показаними раніше. Коли результуюча матриця для вентиля Контроль-V множиться сама на себе, результатом є те, що вона створює матрицю С «ЗАПЕРЕЧЕННЯ». Тому два послідовних вентиля Контроль-V або Контроль-V+ еквівалентні вентилю Фейнмана. Як і вентилі V і V+, коли відбувається операція V/V+, ми отримуємо одиничну матрицю. Таким чином, було продемонстровано, що двобітових квантових вентилів достатньо для синтезу будь-якої унітарної операції в будь-якому розмірі гільбертового простору [77].

Наступним етапом проектування оборотної логіки є розробка бібліотеки дворозрядних квантових вентилів, які дозволяють мінімізувати побудову локально оборотних логічних структур. Однак ці операції можна об’єднати в інтегровані шлюзи Кубіт, оскільки їх унітарна еволюція є правильним представленням квантово-механічної операції простору-часу [78], оскільки набір квантових ациклічних вентилів може моделювати квантові машини Тьюрінга [79]. Цей вентиль реалізовано за допомогою вентиля Фейнмана з вентилем Контроль-V або Контроль V+. Вихід XOR вентиля Фейнмана використовується як керуючий сигнал для вентиля Контроль-V або V+, з яким він поєднаний.

Квантова вартість інтегрованого шлюза Кубіт дорівнює 1, а його затримка в найгіршому випадку дорівнює 1.

Для успішного моделювання квантових взаємодій, а також розробки системи, яка виконує бажані оборотні обчислення, мають бути адаптовані правила оборотної логіки.

Представимо набір правил дизайну та можливої реалізації.

По-перше, визначаємо кожен шлюз ІК як два вхідні сигнали та два вихідні сигнали. Це дозволяє створити проміжні значення сигналу, пов'язані з вхідними/вихідними сигналами фундаментальних шлюзів.

Друге правило, яке ми реалізуємо, – це те, що лінії контролю є станами бази на основі. Для того, щоб оборотна логічна структура належним чином працювала, будь-яка лінія управління повинна отримувати "0" або "1".

По-третє, існують невідомі цінності. Це пов’язано з тим, що лінія управління в «ЗАПЕРЕЧЕННІ» або ІК шлюзів може не отримувати "V", "v", "P" або "p" і все ще належним чином функціонувати. Це правило підкреслює хороші звички дизайну і існують невідомі значення і можуть бути результатом вихідного значення з логічної структури.

Застосуємо модель операцій двох фундаментальних оборотних логічних структур для імітації “Обмін”-затворів. Для цього використаємо два інтегровані шлюзи Кубіт. Дана поведінкова модель є вигідною. Вона точно імітує квантове обчислення операції заміни з меншою квантовою вартістю та затримкою, ніж будь-яка інша поведінкова модель.

Реалізація поведінкової моделі додає переваги порівняно із іншими поведінковими мовами. Вона дозволяє перевантажувати оператора та переосмислювати функцію, процедуру та оператори. Ця робота використовує різноманітні функції та процедури для моделювання та моделювання кожної з цілями, які є специфічними для завдання. Генерування файлів журналу моделювання та написання файлів вимагають, щоб рядок використовувався для запису у файл. Таким чином, виклики функцій для перетворення типу Кубіт у рядок для запису у файл під час тестового режиму використовувались для запису

в журнал моделювання, оскільки нічого не потрібно повертати. Процедури також корисні, оскільки вони дозволяють використовувати сигнали, локальні змінні та заяви про затримку для моделювання. Це переводить моделювання на квантовий рівень не лише для логічних розрахунків, а для визначення затримки поширення сигналу через розроблену оборотну логічну структуру.

Також, користувач може використовувати бібліотеку VHDL Кубіт для створення оборотних макетів логічної структури, які можуть бути застосовані в розробці більших оборотних логічних структур. За допомогою цього схематичного коду VHDL можна автоматично генерувати (за допомогою Xilinx ISE 13.2) і вимагати мінімальних змін у коді, щоб вони стали функціональними.

VHDL також вигідний тим, що він дозволяє автоматичну генерацію тестових балів для надійного тестування. Використовуючи Xilinx 13.2, була можливість генерувати файли TestBench, хоча ці файли потребували модифікації після кожного разу, коли був створений новий файл TestBench. Рішення було графічним інструментом TestBench, написаним мовою програмування C#. Інструмент призначений для читання поточного робочого каталогу для файлів *.vhd та файлів групового TestBench та VHDL у поданні дерева. Результати тестування показали, що програма змогла належним чином генерувати тестовий набір, який генерує 2^n тестові входи, щоб офіційно перевірити перевірку оборотного блоку. Ці тестові палички також могли бути використані в Xilinx 12.4 з симулятором ISIM M.81D та дотримуватися належного синтаксису VHDL.

Продемонструємо, що використання буферів у зонах тактування шляхів, де не має квантової взаємодії, може запобігти умовам, які порушують оборотність. Враховуючи те, що більшість досліджень оборотного логічного синтезу зосереджуються на зниженні квантової вартості та затримки схеми, відмітимо, що введення буферів збільшує обидва ці показники. Фейнман представив буфер 2x2, що складається з двох вентилів «ЗАПЕРЕЧЕННЯ» [13], який відновлює керований сигнал до початкового значення. Буфер 1x1 також може бути розроблений з використанням двох інверторів. Відсутність буфера 1x1, вартість і затримка якого дорівнює одиниці, стає проблематичною, оскільки кожна зона

тактування має часовий зріз, еквівалентний 1. Це означає, що без буфера 1x1 із затримкою в одиницю повні оборотні операції в інтегрованій бібліотеці Кубіт неможливі. Тому буфер 1x1 був представлений для використання в бібліотеці ІК. Буфер 1x1 логічно еквівалентний вентилям V і V+, розміщеним послідовно на одному дроті.

Покажемо, що буфер 1x1, реалізований у інтегрованій бібліотеці Кубіт, має вартість і затримку рівну одиниці.

Розглянемо реалізацію шлюзу «ЗАПЕРЕЧЕННЯ» в інтегрованій бібліотеці Кубіт. Ідентична реалізація може бути досягнута шляхом розміщення двох вентилів V або V+ послідовно. Два послідовних вентиля V+ забезпечують наступну унітарну операцію:

$$\frac{1}{i+1}\begin{pmatrix}1 & i\\ i & 1\end{pmatrix} * \frac{1}{i+1}\begin{pmatrix}1 & i\\ i & 1\end{pmatrix} = \frac{1}{2i}\begin{pmatrix}1+i^2 & i+i\\ i+i & 1+i^2\end{pmatrix} = \begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix} \qquad (3.3)$$

Це операція «ЗАПЕРЕЧЕННЯ». Це означає, що робота V і V або V+ і V+ разом послідовно становить вартість рівну одиниці. Це пояснюється тим, що ці операції представляють повне обертання, досягаючи стану «0» або «1». Коли V і V+ розташовані послідовно, вони дають таку унітарну операцію:

$$\frac{1+i}{2}\begin{pmatrix}1 & -i\\ -i & 1\end{pmatrix} * \frac{1}{i+1}\begin{pmatrix}1 & i\\ i & 1\end{pmatrix} = \frac{1}{2}\begin{pmatrix}1-i^2 & i-i\\ i-i & 1-i^2\end{pmatrix} = \begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix} \qquad (3.4)$$

Це матриця ідентичності. Затвори буфера не представляють обертання, але все одно досягають стану «0» або «1», як і «ЗАПЕРЕЧЕННЯ». А послідовне розміщення V і V+ за вартістю еквівалентно розміщенню V і V або V і V+ послідовно, що утворює вартість і затримку в одиницю. Таким чином, реверсивний буфер 1x1 ІК також несе вартість і затримку в одиницю.

В результаті, завдяки використанню цього методу буферизації, багато оборотних логічних структур, які використовують інтегровану бібліотеку Кубіт,

тепер будуть логічно правильними. Крім того, оскільки було зроблене припущення, що сигнали виходять з кожних затворів у момент найгіршої затримки, то усі розрахунки затримки тепер правильні. Єдиний негативний момент полягає в тому, що представлений дизайн призводить до вищої вартості, ніж було пораховано раніше.

Відзначимо, що Ді Вінченцо продемонстрував раніше, що всі оборотні логічні структури 3x3 можуть бути досягнуті за допомогою шести або менше інтегрованих Кубіт вентилів 1x1 або 2x2. Таким чином, оборотна логічна структура 3x3 має m 1x1 елементів і n 2x2 елементів, так що $m + n \leq 6$. Це означає, що структура має $m + n$ тактові зони. У зоні тактування з вентилем 2x2 потрібен лише один буфер для належного пом'якшення зони тактування, оскільки є лише один дріт, який не працює лише через квантову взаємодію. У зоні тактування з вентилем 1x1 потрібні два буфери. Таким чином, у будь-якій оборотній логічній структурі 3x3 не потрібно більше $m+2n$ буферів для забезпечення належної роботи схеми.

Представимо алгоритм оптимізації.

На першому етапі:

1) ми перемикаємо вентилі в синтезованій схемі з їх еквівалентом у представленій бібліотеці;
2) всі шлюзи Тоффолі, Фредкіна, Переса і "Обміну" замінюються наступними схемами:
   2.1) зі стану кубіта «0» виходить логічний 0;
   2.2) зі стану кубіта «1» виходить логічна 1;
   2.3) зі стану кубіта «v» вводиться «0» та застосовується трансформація V;
   2.4) зі стану кубіта «V» вводиться «1» та застосовується трансформація V;
   2.5) зі стану кубіта «p» вводиться «0» та застосовується трансформація V+;

2.6) зі стану кубіта «P» вводиться «1» та застосовується трансформація V+;

2.7) при стані кубіта «U» кубіт знаходиться в невідомому стані;

3) вдбувається перевірка двох шлюзів, які примикають один до одного, щоб перевірити, чи існує менша реалізація, яка може досягти того самого логічного обчислення з меншою вартістю.

Основна перевага цієї методики полягає в тому, що шляхом заміни двох або трьох суміжних вентилів меншою реалізацією досягається зниження квантової вартості та затримки.

Інше правило зменшення, яке реалізується на цьому етапі, полягає в тому, що два послідовних вентиля Тоффолі мають однакові керуючі сигнали. Алгоритм зазвичай замінює ці вентилі двома 5-кубітними вентилями Тоффолі. Однак це спричинило б витрати в 2 рази більше, ніж необхідно. Тут важливо зазначити, що якщо в послідовності є третій вентиль Тоффолі, який відповідає цим критеріям, алгоритм повинен додати 5-кубітний вентиль, оскільки значення Q було відновлено до початкового значення.

Третє правило зменшення, яке реалізується на цьому етапі, полягає в тому, що два послідовних вентиля Тоффолі мають одну спільну лінію керування та спільну залежну лінію. Алгоритм зазвичай замінює ці вентилі двома 5-кубітними вентилями Тоффолі. Однак це спричинило б витрати на 1 більше, ніж необхідно. Знову ж таки, якщо в послідовності є третій вентиль Тоффолі, який відповідає цим критеріям, алгоритм повинен додати 5-кубітний вентиль, оскільки значення Q було відновлено до початкового значення.

Четверте правило редукції, яке реалізується на цьому етапі, полягає в тому, що за Тоффолі слідує Фейнманівська лінія, яка має одну спільну лінію керування та спільну залежну лінію. Алгоритм зазвичай замінює ці вентилі на 5-кубітні затвори Тоффолі та 1-кубітні ворота Фейнмана. Однак це спричинило б витрати на 1 більше, ніж необхідно.

Отриманий граф розбивається на вузли наступним чином: задано оборотну схему з n входами та n виходами з вартістю c, отриманий граф матиме 2n+c-1

вузлів і n+2c ребер. Вузли від 0 до n-1 представляють входи. Вузли від n до n+c-1 представляють кожен квантовий вентиль у порядку їх затримки. Вузли від n+c до 2n+c-1 представляють виходи пристрою. Ребра мають напрямок, і оскільки алгоритм працюватиме від виходів до входів, напрямок йде від виходів до входів.

Кожен вузол складається з наступного: ціле число, що відноситься до типу затвору (наприклад, якщо тип затвору дорівнює «0», то це вхід; якщо тип затвору дорівнює «1», то це вихід; якщо тип затвору дорівнює «2», то це вентиль Фейнмана («ЗАПЕРЕЧЕННЯ»); якщо тип затвору дорівнює «3», то це елемент керування V; якщо тип затвору дорівнює «4» - це Контроль-V+; якщо тип затвору дорівнює «5», то це “ОБМІН”; а якщо тип затвору дорівнює «6», то це інвертор), два логічних значення Контроль1 і Контроль2 визначають, чи є кожен вхід лінією керування чи залежною лінією («Хибне значення» = залежність, «Істинне значення» = контроль), два цілих числа, що вказують, де знаходиться наступний вузол (вузол «1» вказує, куди йде верхній рядок, а вузол «2» вказує, куди йде нижній рядок), Тип 1 і Тип 2 – вказівки, чи є наступний шлюз контрольним верхнім (1), контрольним нижнім (2), залежним верхнім (3) і залежним нижнім (4) – і логічне значення включає вказівку про те, чи слід включити затвор в кінцеву схему.

Використовуючи значення вузла «1» і вузла «2», а також значення Контроль1 і Контроль2, переводимо схему, у граф послідовності. Отриманий граф є спрямованим ациклічним графом, де є *n* вузлів без вхідних ребер і *n* вузлів без вихідних ребер.

Список суміжності складається з пари цілих чисел для кожного вузла. Кожне значення списку буде цілим числом від 0 до 6. Значення 0 означає відсутність краю. Якщо верхній рядок є контрольним, значення буде 1. Якщо нижній рядок є контрольним, значення буде 2. Якщо верхній рядок є залежним, значення буде 3, а якщо нижня лінія є залежною лінією, тоді значення буде 4. Наприклад, розглянемо вузол 4. Верхня лінія є контрольною лінією, а край походить від вузла 6, тому значення (4,6) буде 1. Крім того, залежна лінія походить від вузла 5, тому значення вузла (4,5) у списку суміжності

дорівнюватиме 4. Іноді і контрольна, і залежна лінія походять з одного вузла. Наприклад, розглянемо вузол 11: верхній рядок є залежним, оскільки значення XOR залежить від контрольного рядка, отже, значення для (11,12) дорівнює 3. Крім того, контрольний рядок, який є нижнім, також походить від 12. Отже, значення також буде 2. Отже, значення вузла (11,12) дорівнює (3,2).

Існує кілька важливих властивостей вузлів, на які слід звернути увагу. Вузли від 0 до n-1 складатимуться лише з однієї 5 або однієї 6. Оскільки кожен вхід відповідає кожному виходу, ми можемо гарантувати, що оборотний граф матиме n значень 5 або 6 у списку суміжності. Наприклад: (0,10) дорівнює 5, (1,4) дорівнює 5, (2,4) дорівнює 6 і (3,5) дорівнює 6. Оскільки є 4 входи, ця частина матриці будується правильно. Вузли від n+c до 2n+c-1 складатимуться лише з одного значення від 1 до 4, оскільки вони походять або від контрольного значення залежного значення вузла. На виходах (9,14) дорівнює 1, (9,16) дорівнює 4, (12,13) дорівнює 1 і (12,15) дорівнює 4. Оскільки є 4 виходи, ця частина матриці побудована правильно.

Самі вузли мають властивості, які відображають природу оборотної схеми. Кожен вхідний вузол має одне вхідне ребро, а кожен вихідний вузол має одне вихідне ребро, обидва з яких є оборотними за своєю природою [14]. Кожен край інвертора має один вхідний і один вихідний фронт, який також є оборотним за своєю природою [11]. Кожен інший вузол має два вхідних ребра і два вихідних ребра, які також є оборотними. Таким чином, загальна кількість вузлів має лінійну залежність від квантової вартості та кількості вхідних даних, так що $n_{\text{загал}} = \text{c} - n_{\text{обміну}} + 2n$.

Далі, використовуючи набір правил, обходимо вузли, які зустрічає кожен процес, і на основі типу вузла визначаємо, чи потрібно вузол видаляти. Якщо вузол видаляється, два вхідні дроти з’єднуються з вихідними. Якщо вузол є «ЗАПЕРЕЧЕННЯМ», ми перевіряємо, чи є наступний шлюз іншим «ЗАПЕРЕЧЕННЯМ» чи шлюзом Контроль-V/V+. Якщо обидва вхідні рядки «ЗАПЕРЕЧЕННЯ» ідентичні вихідним лініям вентиля Контроль-V, тоді їх можна об’єднати в вентиль ІК. Якщо вентиль «ЗАПЕРЕЧЕННЯ» має ту саму лінію

керування, що й вентиль «ЗАПЕРЕЧЕННЯ», а вхідні лінії такі ж, як вихідні лінії попереднього «ЗАПЕРЕЧЕННЯ», тоді вони створюють ідентичність, і обидва можуть бути виключені зі схеми. Існує подібний метод для структур Контроль-V/V+, який ґрунтується на знаннях про те, що два Контроль-V у серії створюють «ЗАПЕРЕЧЕННЯ» (зменшуючи вартість з 2 до 1), а Контроль-V і Контроль-V+ у серії з тією самою лінією керування та лінії вводу/виводу створюють ідентичність (зменшуючи вартість з 2 до 0).

Після проходження контуру ми використовуємо два правила, щоб визначити, чи потрібно видалити ребро чи вузол:

1) якщо жодне вхідне ребро не позначено, тоді відповідний вентиль не потрібен;

2) якщо вузол має один позначений вхідний рядок, який є лінією керування, і відповідний рядок також є лінією керування, тоді шлюз не потрібний, і потрібно встановити значення «Наступний шлюз» попереднього вузла на значення шлюзу наступного вузла. Після того як набір правил запущено в схемі, його квантову вартість було зменшено з 10 до 8.

Наступним кроком ми оптимізуємо схему, виконавши пошук з потрібних виходів. Розглянемо реалізацію схеми, де користувач бажає мати виходи P і R. Отже, Q і S будуть сміттєвими виходами. Тому ми виконаємо пошук у глибину. Граф буде реалізований у вигляді двох матриць суміжності, де кожна матриця A0 представляє контрольні ребра, а матриця A1 представляє залежну матрицю. Перевага подання списку суміжності полягає в тому, що пошук у глибину може виконуватися за час O(V+E). Крім того, за допомогою паралелізму та відповідного захисту критичної секції – де кожен вузол є критичною секцією – алгоритм може виконувати цей пошук з кожного бажаного виходу одночасно за час O(V+E). Ми визначаємо, що загальна кількість ребер прямо пропорційна квантовій вартості. Таким чином, час виконання оборотного алгоритму становить O (C). Після запуску алгоритму пошуку ми використовуємо набір правил для перевірки кожного вузла, щоб визначити, чи слід його видаляти. Використовуючи операцію видалення вузлів, при умові, що вузол потрібно видалити, ми

об’єднуємо вхідні рядки та вихідні рядки вузла та видаляємо вузол, оскільки було доведено, що він лише логічно сприяє виведенню сміття.

Користувач вводить оборотну схему, R(N,C), де N — кількість входів/виходів, а C — вартість схеми. Алгоритм перетворює дану схему в інтегровану бібліотеку Кубіт і генерує нову схему. Створення списку суміжності займає O(C) часу. Далі алгоритм запитує у користувача потрібні логічні виходи та зберігає їх в U. Для кожного вузла в U алгоритм генерує одночасний потік, а потім запускає Реверсивний алгоритм у кожному потоці. Це дозволяє виконати цей крок за час O(C).

Пропонований алгоритм був написаний на C. Програма згенерувала схему на VHDL у запропонованій бібліотеці Контроль-V/V+/ІК на основі схем, представлених у [19]. У цьому розділі ми спочатку покажемо покрокову реалізацію нашого алгоритму на основі попередньо представленої синтезованої схеми, а потім порівняємо наші результати з наскрізним набором раніше існуючих синтезованих схем із попередніх алгоритмів. Розглянемо оборотну схему, отриману з позитивного дерева Давіо в [37]. Схема складається з 5 шлюзів Тоффолі та 2 шлюзів Фейнмана.

АЛГОРИТМ ІК Синтез:

Алгоритм (Схема R(N,C))

Перетворюємо R на R’, де R’ знаходиться в бібліотеці ІК

Створюємо список суміжності L з R’

Для кожного вузла $i \in L$

Перевірити злиття (L)

Отримати вказані користувачем логічні входи U

Створити паралельний потік для кожного вузла $i \in L$

Одночасно запускаємо *Реверсивний алгоритм (L, U[i])* для кожного потоку *i*.

Для кожного вузла $i \in L$

Якщо значення для перевірки видалення вузла «істинне», то видалити вузол

Видалити вузол (i)

Видалення вузла (номер попереднього вузла 1, номер попереднього вузла 2, і, номер наступного вузла 1, номер наступного вузла 2) Згенеруйте нову схему R з L і поверніть R.

Перевірка алгоритму та порівняння з попередніми результатами: Еквівалентна схема в цій бібліотеці має початкову вартість 27. Виходи U і V були позначені як виходи сміття. Для реалізації запропонованого алгоритму входи P, Q, R, S і T позначині як потрібні логічні виходи. Після запуску алгоритму вартість схеми була зменшена з вартості 27 до вартості 24, що є покращенням на 11,11%. Отже, отримані результати показують, що було досягнуто значного зниження квантової вартості в порівнянні з раніше синтезованими схемами.

3.2 Удосконалення алгоритму з використанням двошинної адіабатичної логіки.

Оборотна логіка є багатообіцяючою парадигмою обчислювального дизайну, яка представляє метод побудови комп'ютерів, які виробляють довільно низьке розсіювання тепла [20]. Основний принцип оборотних обчислень полягає в тому, що біективний пристрій з однаковою кількістю вхідних і вихідних ліній створює обчислювальне середовище, де електродинаміка системи дозволяє передбачати всі майбутні стани на основі відомих минулих станів, і система досягає всіх можливих станів, що призводить до відсутності розсіювання тепла [1][69]. Оборотна логіка має важливе значення в майбутніх реалізаціях CMOS [70], квантових обчислень [2], оптичних обчислень [80] і ДНК-обчислень [81], оскільки ці структури потрібні для подолання бар'єру kTln(2) для розсіювання енергії [82].

Адіабатична логіка — це реалізація оборотної логіки в CMOS, де струм, що протікає через схему, контролюється таким чином, що розсіювання енергії через перемикання та розсіювання конденсатора мінімізується [33]. Це досягається шляхом переробки енергії контуру, а не розсіювання її в навколишнє середовище. Це вигідно для реалізації CMOS, оскільки вхідні та вихідні заряди зберігаються окремо. Реалізація адіабатичної логіки CMOS була використана для покращення

енергоспоживання порівняно з транзисторною логікою [32]. Адіабатична логіка вимагає використання змінних функцій замість швидшого перемикання, досягнутого ступінчастими функціями. Також було продемонстровано, що адіабатична логіка подвійної шини, реалізована за оборотними принципами, значно зменшує диференціальне споживання електроенергії [83]. Таким чином, адіабатична логіка з подвійною шиною показує значні перспективи як методологія проектування в програмах, де безпека є основним проектним показником, а більш висока робоча частота не є бажаною. Одним із таких застосувань є розробка смарт-карт, де пом'якшення атак диференціального аналізу потужності є дуже бажаним, а робоча частота становить 13,56 МГц відповідно до стандарту ISO 14443.

У будь-якій адіабатичній схемі CMOS необхідно вирішити дві проблеми. По-перше, реалізація повинна призвести до енергоефективної конструкції комбінованого джерела живлення та тактового генератора. По-друге, оборотні логічні функції вимагають більших логічних витрат, що відповідають бієктивній вимозі [30]. Таким чином, енергію, що розсіюється під час перемикання ланцюга, необхідно контролювати та переробляти, а не розсіювати в навколишнє середовище.

Зменшення розсіювання енергії досягається за рахунок використання функції наростання замість більш швидкого перемикання, досягнутого в ступінчастих функціях. Таким чином, транзистори можуть використовуватися в адіабатичній роботі, незважаючи на те, що вони продемонстровані як пристрої з втратами [30], і це досягається шляхом застосування двох правил:

1) транзистор завжди включений, коли через нього протікає значний струм;
2) коли є значна різниця між джерелом і напругою стоку, транзистор повинен бути вимкнений.

У [31] було показано, що адіабатичні схеми забезпечують зниження розсіювання енергії на 60% при 20 МГц і на 35% менше енергії при 100 МГц, а в [32] продемонстровано реверсивні двошинні CMOS-транзистори.

Адіабатичні схеми CMOS вимагають здатності бути фізично однозначними, тобто вихідні сигнали можуть бути розміщені на виходах, а унікальні вхідні сигнали можуть бути відтворені на вхідних проводах. Для досягнення цієї мети в [34] був використаний підхід з подвійною рейкою, в якому три основні оборотні логічні структури були розроблені та виготовлені за технологією 0,35 мкм, де $V_{tp} = 0.6V$ і $V_{tn} = -0.6V$. Схеми не мають входів джерела живлення, тобто вся енергія вихідних сигналів походить від вхідних сигналів. Цей метод удосконалив ЛВЗРР і ЛЕВЗ, значно зменшивши накладні витрати, необхідні для виконання оцінки та розрядки, а також покращивши розповсюдження сигналу, дозволивши покращити каскадування пристроїв.

Квантові принципи адіабатичності та оборотності зосереджені на розсіюванні нульової енергії в ідеальних умовах. Термінологія для «адіабатичного контуру» відрізняється тим, що використання лінійних функцій намагається мінімізувати енергію, що розсіюється через енергію. Дроти та буфери CMOS розсіюють енергію, незважаючи на те, що обидва вони логічно оборотні. Звичайні CMOS-інвертори логічно оборотні, але вони не виробляють вхідний сигнал, якщо ви помістите еквівалентний вихідний сигнал на вихідний штифт. Поширення тактового сигналу вимагає ще більше енергії. Це тому, що перемикання в CMOS вимагає певної кількості енергії. Ступінь, до якої ця енергія перемикання обмежує розсіювання енергії, є спірною [58, 75, 82]. Що не суперечить, так це те, що швидший час перемикання та більша робоча частота потребують більшого розсіювання енергії.

Однією з мотивацій для цього алгоритму синтезу є мінімізація кількості перемикачів і шляхів, необхідних для реалізації адіабатичної логіки двох рейок у схемах безпеки. Одним із недоліків методології подвійної шини є те, що вона вимагає більше транзисторів і більше входів і виходів. Це збільшує довжину дроту, що призводить до більшого розсіювання енергії.

Крім того, чим більша схема, тим більша буферність потрібна, що також збільшує розсіювання енергії. Деякі методології адіабатичного проектування, такі як ЛВЗРР [32], вирішують цю проблему. Багато інших схем використовували аж

чотирифазні годинники для керування поширенням сигналу по тракту даних, що вимагало більших витрат і обмежень сигналів. Для вирішення цієї проблеми шляхом мінімізації необхідних накладних витрат була представлена логіка ефективного відновлення заряду (ЛЕВЗ) [84]. Цей алгоритм розроблено з урахуванням цих недоліків. Зменшуючи кількість перемикачів, було зменшено розмір ланцюга, довжину дроту та споживану потужність кінцевого ланцюга.

Схема, яка була удосконалена за допомогою запропонованого алгоритму. є схемою 16 на 16 і складається з 8 входів і 8 інвертованих входів, а також 8 виходів, інверсія яких представлена на інших 8 виходах. Форма сигналу представляє всі 256 можливих вихідних комбінацій. Адіабатичний контур із подвійною шиною споживає середню потужність 1,1850*10^{-6} Вт.

Найбільший сплеск потужності становив 26,4*10-6 Вт, а найменший — 18,6*10^{-6} Вт, що становить 30% різниці в диференціальній потужності. Удосконалена схема також є прикладом, коли методологія проектування з подвійною шиною є перевагою порівняно зі звичайними та одношинними реалізаціями. Оскільки використовується транзисторна логіка, схеми є оборотними. Розміщення сигналів на вихідних виводах дозволяє однозначно визначати вхідні сигнали. Більшість реалізацій шифру Rijndael вимагають схеми для шифрування та схеми для дешифрування. Схема, синтезована запропонованим алгоритмом, вимагає 4806 комутаторів. Замість того, щоб робити другу схему для дешифрування, ми можемо просто розмістити вентиль Фредкіна 3 на 3, який використовується для мультиплексування, на кожному вході та виході, щоб диктувати потік схеми. Кожен вентиль Фредкіна несе вартість 8 комутаторів, тобто удосконалена схема 16 на 16 потребує 32 вентилів Фредкіна, що становить додаткову вартість 256 комутаторів. Цей підхід зменшує вартість шифрування/дешифрування удосконаленої схеми з 9612 комутаторів до 5062 комутаторів.

Більшість оборотних алгоритмів логічного синтезу корелюють необхідні логічні переходи з бібліотекою синтезу, що складається з логічних елементів.

Зменшення, якого досягається завдяки безпосереднього відображення в технології реалізації замість вставки бібліотечних еквівалентних схем.

Перший етап алгоритму включає зчитування бажаних вихідних значень і визначення горизонтального зсуву від матриці перестановок до унітарної матриці. Оскільки всі оборотні та адіабатичні логічні структури є логічно та фізично однозначними, жоден рядок чи стовпець не матиме більше одного значення «1». Матриця перестановок фундаментального елемента «ЗАПЕРЕЧЕННЯ» ліворуч і унітарна матриця праворуч має вигляд

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ -1 \end{bmatrix} \tag{3.5}$$

Шлюз «ЗАПЕРЕЧЕННЯ» є оборотною логічною структурою 2x2, що означає, що вона має чотири можливі вихідні комбінації. Горизонтальні стовпці представляють вхідні комбінації, вертикальні стовпці – вихідні комбінації, а «1» представляє відповідність між введенням і виходом. Наприклад, четвертий стовпець і третій рядок матриці перестановок «ЗАПЕРЕЧЕННЯ» є «1», що означає, що вхід «11» корелює з виходом «10». В унітарній матриці четвертий рядок має значення «1» у четвертому стовпці. Матриця 1x4 праворуч представляє обчислені горизонтальні зсуви. Ми визначаємо горизонтальний зсув як різницю між очікуваним розташуванням в унітарній матриці та фактичним розташуванням у матриці перестановок. Рівняння для кожного зсуву дорівнює $Off_i = Perm_i - -Unit_i$. Наприклад, матриця перестановок «ЗАПЕРЕЧЕННЯ» має «1» у четвертому рядку третього стовпця, а унітарна матриця має «1» у четвертому стовпці. Використовуючи рівняння, ми визначили зміщення як $Off_4 = Perm_4 - -Unit_4 = 3 - 4 = -1$.

Визначимо ребра та перемикачі для початкової схеми, створюючи ребро для кожного вузла, де зсув є ненульовим значенням, і порівнюючи двійкові представлення початкового та кінцевого вузлів ребра на графі. Якщо порівняння

показує, що всі значення, крім одного, ідентичні, тоді ці значення представляють перемикання на шляху. Значення, яке відрізняється, представляє початковий і кінцевий вузол на шляху. Якщо порівняння показує, що кілька значень різні, і лише одне значення є однаковим, тоді алгоритм створить край для кожного біта різниці від цільового значення до вихідного значення, а інші вихідні значення представляють перемикачі. Якщо порівняння показує, що немає ідентичних значень, тоді алгоритм створить ребро від початкового вузла до вузла призначення для кожного вузла, а решта вузлів представляють перемикачі.

Визначимо початкові ребра для воріт Тоффолі. У цьому випадку кількість ідентичних значень дорівнює 2. Отже, значення перемикача A=1 і C=1, які з часом будуть замінені адіабатичними перемикачами «ввімкнено» для «1». Перше ребро має початок {111} і призначення {110}. Це означає, що початок ребра є для C, а призначення ребра R'. Подання перемикачів краю {1-}. Подібним чином друге ребро має початок {110} і пункт призначення {111}. Це означає, що початок ребра є для C', а призначення ребра – R.

Існує унікальний випадок для оптимізації оборотних логічних схем, де значення двох входів міняються місцями. Це відбувається в вентилі "Обміну" без необхідності перемикачів або в вентилі Фредкіна з керуючим сигналом, що забезпечує контрольований "Обмін". На цьому етапі алгоритму перевіряють Sw_Cond (Додаток А). Sw_Cond існує, коли є рівно два біти, що відрізняються у вихідному та кінцевому вузлах, і ребро від вузла призначення має місце призначення до початкового вузла. У вентилі "Обмін" є рівно два біти різниці, і два біти всередині початкового вузла відрізняються. Це вказує на обмін, де перемикання не потрібні. Для воріт Фредкіна існує рівно два біти різниці, і два біти всередині початкового вузла, також, відрізняються. Вони спільно використовують ідентичні біти, A = 1, що означає, що обмін відбувається лише тоді, коли виконується ця умова. Тому, коли A=0, сигнали передаються до нормального вихідного значення.

У цьому алгоритмі всі випадки Sw_Cond будуть відкладені та не синтезовані до останнього етапу. Це пов'язано з тим, що всі інші випадки

передбачають визначення, коли вхідний сигнал буде поширюватися на відповідний вихід або його інверсію. У випадку Sw_Cond (Додаток А) сигнал буде поширюватися на зовсім інший вихід. Наприклад, у вентилі “Обмін” умова Sw_Cond визначає, що вхід A передається на вихід Q, вхід A’ – на Q’, вхід B – на P, а вхід B’ – на вихід P’. Це спрощує наступні два етапи алгоритму та зменшує загальну вартість. Наприклад, вартість шлюзу “Обмін” зменшується з вартості 8 перемикачів до ідеальної вартості нульових перемикачів, а шлюз Фредкіна зменшується з вартості 16 до ідеальної вартості 8.

Ребра, створені першою частиною алгоритму, є біективними, а також мають властивість, що кожне ребро, яке походить із вхідного вузла i, закінчується на тому самому вихідному вузлі j. Ця властивість дозволяє використовувати мінімізацію логічних функцій, щоб мінімізувати кількість вихідного вузла до вузла призначення.

Інверсійний вентиль має найвищий потенціал для оптимізації. Початковий результат полягає в тому, що між чотирма парами вузлів утворюється вісім ребер. Поєднання A і P’ має комбінації перемикачів {-0} і {-1}. Оскільки ця комбінація країв може бути зменшена до {--}, є можливість для оптимізації. Насправді кожне з цих ребер може бути зведено до {--}, що означає, що перемикачі не потрібні. Комбінація вузлів вводу-виводу для A→P’ і A’→P має мінтерми {1-} і {0-}, які скорочуються до {--}. Комбінація вузлів вводу-виводу для B→Q’ і B’→Q має мінтерми {-1} і {-0}, які зменшуються до {--}. Тому кількість перемикачів, необхідних у конструкції інверсного затвора, дорівнює нулю.

Останнім етапом алгоритму є знаходження ребер від вхідного вузла до ідентичного вихідного вузла. До цього моменту ребра йдуть від вхідного вузла до перевернутого вихідного вузла. Наприклад, усі ребра з вузла 0 з’являться на вихідному вузлі 1, оскільки вузол 0 відповідає входу A, а вихідний вузол 1 відповідає виходу P’. Нам потрібно створити ребра, які йдуть від вхідного вузла 0 до вихідного вузла 0. Якщо вихідний вузол i є звичайним вузлом, а i+1 є перевернутим вузлом для реалізації з подвійною шиною, тоді значення вихідного вузла i ніколи не повинно дорівнювати значенню вихідного вузла i+1.

Використовуючи це правило, ми визначаємо межі, знову запускаючи мінімізацію EMAM. Результатом цього етапу є те, що схема стає фізично та логічно оборотною, і задовольняються обидва обмеження для адіабатичного впровадження.

Як приклад, покажемо весь синтез для шлюзу Тоффолі, де логічні виходи представляють $P = A, Q = B$ і $R = AB \oplus C$. Алгоритм визначає, що існують лише два ребра, що зменшує складність вхідних даних, необхідних для частини евристичного алгоритму EMAM. Кожне ребро має лише один біт відмінності від вхідного та вихідного вузлів. Отже, ребро від {111}→{110} має початок C і пункт призначення R', а перемикачі A=1 і B=1. Крім того, ребро від {110}→{111} має початок C' і пункт призначення R, а перемикачі A=1 і B=1. Оскільки кожна комбінація вузлів входу-виводу має лише один край, метод EMAM не потрібен. На останньому етапі алгоритму, оскільки мінтерми кожного ребра дорівнюють {11-}, це означає, що мінтерми, які створюють протилежне ребро, є {00-}, {01-} і {10-}. Використовуючи метод EMAM, основними імплікантами є {0--} і {-0-}. Таким чином, у схемі використовуються два ребра від кожної комбінації вузлів вводу-виводу, s кожен з одним перемикачем.

$$\widetilde{\mathrm{A}} = \begin{pmatrix} 10000000 \\ 01000000 \\ 00100000 \\ 00010000 \\ 00001000 \\ 00000010 \\ 00000100 \\ 00000001 \end{pmatrix}, \widetilde{\mathrm{B}} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ -1 \\ 0 \end{pmatrix} \tag{3.6}$$

Умову обміну розглянемо на прикладі шлюзу Фредкіна, де логічні виходи представляють $P = A, Q = A`B \oplus AC$ і $R = AB \oplus A`C$. Початкову матрицю перестановок $\widetilde{\mathrm{A}}$ для затвору Фредкіна та матрицю горизонтального зміщення $\widetilde{\mathrm{B}}$ представлено в (3.1). Алгоритм визначає, що існують лише два ребра, і обидва відповідають умові Sw_Cond (Додаток А). Загальний біт A=1, отже, фронт від

B->R, B'->R', C->Q і C'->Q' має перемикач, еквівалентний A=1. Використовуючи метод мінімізації EMAM, визначаємо, що A=0 є перемикачем, необхідним для переходу до початкового краю. Таким чином, для схеми потрібно всього 8 перемикачів.

Тепер, формально визначаємо запропонований алгоритм для синтезу двошинних адіабатичних кіл. Алгоритм був написаний на C++ з використанням компілятора gcc версії 3.4.6. Граф G і матриця горизонтального зсуву h[n^2] є глобальними змінними. Другий алгоритм називається Cr_Initial_Edg (Додаток А). Третій і останній алгоритм — EMAM_Reduction. Потік синтезу поданий у Додатоку А.

У Додатку Б показано кількість транзисторів, необхідних для синтезу фундаментальних оборотних логічних структур. У таблиці Додатку Б показано бажану послідовність виводу, представлену користувачем у першому стовпці. У другому стовпці показано згенеровані горизонтальні зміщення на основі цих виходів. Третій стовпець показує початкову кількість перемикачів, згенерованих частиною Cr_Initial_Edg алгоритму. Четвертий стовпець показує кількість перемикачів, що залишилися після виконання EMAM_Reduction. П’ятий стовпець показує остаточну кількість перемикачів після того, як EMAM_Reduction використовує «don’t cares» для отримання протилежних фронтів, щоб гарантувати, що схема відповідає як логічним вимогам для оборотної логіки, так і фізичним вимогам для адіабатичної реалізації. Представлений алгоритм перевершив попередні найкращі результати на 27%.

Тепер застосуємо техніку прямого зміщення тіла [85], до звичайного інвертора та інверторів ЛВЗРР, ЛЕВЗ, щоб продемонструвати ефективність зміщення тіла в динамічній диференціальній логіці. Змодельовані інвертори – це звичайний інвертор без зміщення тіла, звичайний інвертор із зміщенням тіла, інвертори ЛВЗРР із зміщенням тіла та без нього та інвертори ЛЕВЗ із зміщенням тіла та без нього. Щоб показати поліпшення при підпороговій роботі, встановлено напругу живлення ланцюгів 0,5В, а вхідну частоту – 13,56МГц.

Розглянемо порівняння інвертора A та B (звичайного). Інвертори A і B тепер назвемо IHBa та IHBb. Виконання аналізу перехідних процесів IHBa на 1 МГц дало оптимізований час наростання та спаду, коли розмір транзистора становив $w_P / w_N = 540nm/324nm$. Ми отримали зміщення тіла $V_b = 0.14V$ шляхом визначення мінімальної різниці в піковій потужності на вихідних перемикачах для IHBb. Результатом цього є те, що IHBb покращив середню потужність $P_{Peak_{fall}}$ і диференціальну потужність порівняно з IHBa, причому компроміс збільшився $P_{Peak_{rise}}$. Середня потужність IHBb становить $5{,}5678 * 10^{-10}$ Вт, що на 4,58% більше, ніж IHBa. Крім того, значення IHBb становить $P_{Peak_{fall}}$ = $2{,}8299 * 10^{-8}$-8, що є покращенням на 7,08%. Значення IHBb $P_{Peak_{rise}}$ становить $2{,}7697 * 10^{-8}$ Вт, що є збільшенням на 28,35%. Однак це збільшення є корисним для конструкції, оскільки це призводить до значного зменшення P_{diff}, де має значення $6{,}0202 * 10^{-9}$ Вт. Це покращення диференціальної потужності IHBa на 43,28%.

Миттєва потужність IHBa показує чітку різницю в стрибках потужності, коли вихідний сигнал зростає та коли вихідний сигнал падає, що полегшує зловмиснику визначення вхідного сигналу сигналу на основі вихідного сигналу та форми миттєвої потужності. Скачки потужності для IHBb дуже схожі, що зменшує ефективність простої атаки ДАП. Конструкція IHBb покращує диференціальну потужність традиційного інвертора на коефіцієнт 957,51.

Розглянемо порівняння інвертора C та інвертора D (ЛВЗРР). Тут ми проводимо таке ж моделювання конструкції для інверторів C і D, які тепер назвемо IHBc і IHBd. Виконання аналізу перехідних процесів IHBb на 1 МГц дало оптимізований час наростання та спаду, коли розмір транзистора становив $w_P / w_N = 540nm/300nm$. Аналіз показує покращення часу наростання та спаду, а також роботи для IHBd.

При частоті 1 МГц інвертор ЛВЗРР IHBc має середнє енергоспоживання $9{,}1178 * 10^{-11}$ Вт, що на 84,37% більше, ніж IHBa. Значення IHBc становить $P_{Peak_{fall}}$ = $7{,}1045 * 10^{-10}$ Вт, покращення на 76,67% порівняно з IHBa. Крім того,

значення IHBc становить $4{,}0357 * 10^{-9}$ Вт, що є покращенням на 79,66% порівняно з IHBa. Виявилося, що інвертор IHBd має зміщення в середній точці ($V_b = 0.25V$), оскільки форми сигналів напруги зберігаються в середній точці, коли перемикання не відбувається в реалізації ЛВЗРР.

Значення IHBd становить $P_{Peak_{fall}} = 5{,}6502 * 10^{-10}$, покращення на 23,75% порівняно з IHBc і 81,44% порівняно з IHBa. Значення IHBd становить $P_{Peak_{rise}} = 3{,}3563 * 10^{-9}$, покращення на 16,83% порівняно з IHBc і 83,08% порівняно з IHBa. Диференціальна потужність IHBd є покращенням на 15,28% порівняно з IHBc і на 73,07% порівняно з IHBa. IHBc має значно меншу диференціальну потужність, але IHBf має найменшу диференціальну потужність, що демонструє його ефективність. Конструкція IHBd покращує диференціальну потужність звичайного інвертора на коефіцієнт 2065,2.

Тепер розглянемо порівняння інвертора E та інвертора F (ЛЕВЗ). Ми продовжуємо той самий аналіз конструкції для інверторів E і F, які тепер називаються IHBc і IHBd, які є інверторами ЛЕВЗ. Він демонструє покращення часу наростання та спаду, а також роботи для IHBf.

При 1 МГц інвертор ЛЕВЗ IHBe має середнє споживання електроенергії $9{,}1178*10^{-11}$ Вт, що є покращенням на 90,27% порівняно з IHBa. Значення IHBe для $P_{Peak_{fall}}$ становить $1{,}1649*10^{-9}$ Вт, покращення на 96,17% порівняно з IHBa. Крім того, значення IHBc для $P_{Peak_{rise}}$ становить $6{,}3389*10^{-14}$ Вт, що є покращенням на 99,99% порівняно з IHBa.

Виявлено, що інвертор IHBf має зміщення в середній точці, оскільки форми сигналів напруги зберігаються в середній точці, коли перемикання не відбувається в реалізації ЛВЗРР. Середнє споживання електроенергії інверторами IHBf становить $3{,}0568*10^{-11}$ Вт, що на 94,76% більше, ніж IHBa. Значення IHBf для $P_{Peak_{fall}}$ становить $3{,}4382*10^{-10}$, покращення на 70,48% порівняно з IHBe та 98,87% порівняно з IHBa. Значення IHBf для $P_{Peak_{rise}}$ становить $3{,}3563*10^{-9}$, покращення на 16,83% порівняно з IHBc і 83,08% порівняно з IHBa.

Диференціальна потужність IHBd становить 1,3083*10-10^{-14} Вт. Покращення на 79,36% порівняно з IHBe та на 99,99%% порівняно з IHBa.

Конструкція IHBf покращує диференціальну потужність звичайного інвертора на коефіцієнт 16772,09. IHBe має значно меншу диференціальну потужність, але все ще IHBf має найменшу диференціальну потужність, що демонструє його ефективність.

Метод основного зміщення, який використовувався для покращення середньої, пікової та диференціальної потужності звичайних і адіабатичних інверторів, використовувався для досягнення подібних покращень в CMOS-адіабатичному затворі Тоффолі. Середня потужність зміщеного затвора Тоффолі становить 2,1865*10^{-10} Вт, а пікова потужність — 1,9348*10^{-8}Вт. Найбільша диференціальна потужність у піках становить 2,29*10^{-9} Вт.

Виходи середнього рівня затвора Тоффолі для не зміщеного тіла показують погіршені сигнали, тоді як виходи зі зміщеним тілом працюють належним чином. Затвори Тоффолі без зміщення тіла мають середню потужність 4,4622*10^{-10}Вт, а пікову потужність — 2,0126*10^{-8} Вт. Найбільша диференціальна потужність у позитивних піках становить 2,6101*10^{-9}Вт. Таким чином, зміщений затвор Toffoli має покращення на 50,9% у середній потужності, на 3,86% у піковій потужності та на 12,25% у диференціальній потужності.

Різниця в потужності враховується в погіршеному вихідному сигналі затвору Тоффолі без зміщення тіла. Оскільки немає зміщення тіла, вихідний сигнал для R і R’ не може повернутися до середньої точки 0,25 В. Чим більше погіршується сигнал, тим більше порушується оборотність схеми. В результаті збільшується розсіювання енергії та енергоспоживання пристрою.

Той самий метод зміщення тіла використовувався для покращення середньої, пікової та диференціальної потужності CMOS-зміщеного затвора Фредкіна. Середня потужність становить 1,8829*10^{-10}Вт, а пікова — 1,2077*10^{-8}Вт. Найбільша диференціальна потужність у позитивних екстремумах — 3,0310*10^{-9}Вт.

Затвор Фредкіна без зміщення має середню потужність 5,3726*10^{-10} Вт, а пікова потужність становить 1,2959*10^{-8} Вт. Найбільша диференціальна потужність у позитивних екстремумах становить 2,5363*10^{-9} Вт. Тому затвор Фредкіна із зміщенням покращує середню потужність на 64,95%, пікову потужність — на 7,80%, диференціальну — на 16,32%. Різниця в потужності пояснюється погіршеним вихідним сигналом не зміщеного затвору Фредкіна.

Представимо методологію проектування високопродуктивної адіабатичної динамічної диференціальної логіки (ВАДДЛ) для пом'якшення атак ДАП у високопродуктивних програмах.

Метою ВАДДЛ є розробка універсальної комірки, здатної динамічно виконувати всі основні логічні обчислення з двома входами (AND, NAND, OR, NOR, XOR та XOR) з мінімальною диференціальною потужністю для кожного логічного обчислення. Пристрій є як логічно, так і фізично однозначним. Це означає, що вхідні сигнали можуть бути однозначно визначені шляхом зчитування вихідних сигналів, що є необхідністю при реалізації малопотужних оборотних та адіабатичних конструкцій.

Логічні обчислення вихідних сигналів ВАДДЛ є $P = \bar{A}$, $\bar{P} = A$, $Q = \overline{(A + B) \oplus C}$, $\bar{Q} = (A + B) \oplus C$, і $\bar{R} = AB \oplus C$. Логічні виходи ВАДДЛ наступні: якщо контрольний сигнал A=0, то значення виходів P – A', P' – A, Q – $\overline{B \oplus C}$, Q' – $B \oplus C$, R – C`, R' – C; якщо контрольний сигнал A=1, то значення виходів P – A', P' – A, Q – C', Q' – C, R – $\overline{B \oplus C}$, R' – $B \oplus C$; якщо контрольний сигнал B=0, то значення виходів P – A', P' – A, Q – $\overline{A \oplus C}$, Q' – $A \oplus C$, R – C`, R' – C; якщо контрольний сигнал B=1, то значення виходів P – A', P' – A, Q – C', Q' – C, R – $\overline{B \oplus C}$, R' – $B \oplus C$; якщо контрольний сигнал C=0, то значення виходів P – A', P' – A, Q – $\overline{A + B}$, Q' – $A + B$, R – $\overline{AB}$, R' – AB; якщо контрольний сигнал C=1, то значення виходів P – A', P' – A, Q – $A + B$, Q' – $\overline{A + B}$, R – AB, R' – $\overline{AB}$.

Метою базової квадратної схеми є визначення перемикачів, необхідних для проходження вхідного сигналу від входу до виходу. Для того, щоб вихід Q був «1», коли вхід C є «1», або A, або B повинні бути «1», що замикає перемикач.

Конструкція рівня затвора комірки ВАДДЛ має 32 транзистора, кожен з яких має затвор, стік і витік, прив'язані до вхідного або вихідного сигналу. PMOS-транзистори зміщені до номінальної напруги живлення, яка становить 0,8 В у 22-нм моделі [86], а NMOS-транзистори зміщені до землі. Перевага цього підходу полягає в тому, що не потрібні сигнали оцінки та розрядки, що означає, що схема споживає менше енергії, навіть якщо пристрій має більше транзисторів.

Середня потужність пристрою ВАДДЛ на 1 МГц становить $1{,}4634*10^{-9}$ Вт. Таким чином, він споживає $1{,}4634*10^{-15}$ Дж енергії за кожен цикл обчислення. Подібно до інвертора ЛЕВЗ, миттєва розсіювана потужність однакова для кожного перемикання, незалежно від вхідного та вихідного сигналів. Найбільша пікова потужність перемикання пристрою становить $9{,}6867*10^{-8}$ Вт, а найменша пікова потужність перемикання — $8{,}6750*10^{-8}$ Вт, що дає диференціальну потужність $1{,}0117*10^{-8}$Вт.

Середня потужність 4-транзисторного затвора NAND становить $2{,}4926*10^{-9}$ Вт, тобто схема ВАДДЛ дає покращення на 41,29%, незважаючи на наявність на 28 транзисторів більше. Крім того, диференціальна потужність найбільшого та найменшого піків становить $1{,}1413*10^{-6}$ Вт, що означає, що елемент ВАДДЛ покращує диференціальну потужність на коефіцієнт 112,81.

Середня потужність пристрою ВАДДЛ на частоті 13,56 МГц становить $8{,}5963*10^{-9}$ Вт, тому він споживає $6{,}3394*10^{-16}$ Дж енергії за кожен цикл обчислення. Найвища пікова потужність перемикання пристрою становить $1{,}4430*10^{-6}$ Вт, а найменша пікова потужність перемикання — $1{,}3274*10^{-6}$ Вт, що дає диференціальну потужність $1{,}156*10^{-7}$ Вт. Середня потужність затвора NAND становить $2{,}6382*10^{-8}$ Вт, що забезпечує покращення осередку ВАДДЛ на 67,42%. Найбільша пікова потужність перемикання затвора NAND на 13,56 МГц становить $8{,}9768*10^{-6}$ Вт, а найменша пікова потужність — $1{,}0433*10^{-6}$ Вт, що дає диференціальну потужність $7{,}8355*10^{-6}$ Вт. Таким чином, комірка ВАДДЛ покращує на диференціальній потужності в 67,78 разу.

Представлений дизайн ВАДДЛ має перевагу перед попередніми дизайнами в середній потужності для кожного з основних обчислень «ТА», NAND, «АБО», «НЕ», XOR і XNOR. ВАДДЛ покращує ЛЗДМт на 76,41%, порівняно з RCCDL на 93,98% і на 89,65% порівняно з ХДДЛ. Реалізація ЛЗДМт є раніше найкращою реалізацією, оскільки вона використовує фази оцінки та розрядки, подібні до методів ЛВЗРР та ЛЕВЗ. На локальному рівні ЛЗДМт є перевагою з точки зору необхідних транзисторів, оскільки реалізація ЛЗДМт вимагає 16 транзисторів на відміну від 32 транзисторів, необхідних у запропонованій нами реалізації. Однак ця перевага стирається при каскадному з'єднанні комірок разом. Накладні витрати на апаратне забезпечення, необхідні для забезпечення належного часу оцінки та етапів розряду кожної комірки, зростають експоненціально зі збільшенням довжини критичного шляху пристрою [30]. Схема ВАДДЛ не вимагає жодних накладних витрат для підтримки фаз оцінки та розрядки, що робить її кращою коміркою для більших реалізацій, таких як схеми шифрування DES. Однак покращення площі пристрою ВАДДЛ є важливим.

Мета ВАДДЛ полягає в тому, щоб відокремити форму миттєвої потужності від виходу, щоб ускладнити визначення функціональності схеми шляхом зчитування форми миттєвої потужності та вихідних сигналів. Слід зазначити, що комірка ВАДДЛ є біективною, тому вхідні сигнали можуть бути однозначно визначені шляхом вивчення вихідних сигналів. В цьому випадку функціональність клітини можна легко визначити, вивчивши вихід. Ця схема є пристроєм з подвійною рейкою 3*3, тому функцію можна легко визначити шляхом зчитування 2^3 вхідних сигналів. Однак, оскільки осередок ВАДДЛ є універсальним, його можна поєднувати з іншими осередками ВАДДЛ для створення більших схем, що ускладнює ефективність цієї стратегії. Наприклад, 3-вхідний шлюз NAND потребує 7 входів, оскільки він вимагає двох каскадних комірок ВАДДЛ. Таким чином, замість того, щоб читати лише 8 виходів, зловмисник мав би розглянути 512 вводів, щоб належним чином визначити функціональність схеми. Крім того, стандарт потрійного шифрування DES використовує розмір ключа шифрування 56 біт, що означає, що зловмиснику

доведеться проаналізувати вихідні сигнали 7,2057*1016, щоб належним чином провести зворотне проектування схеми.

Крім того, на відміну від ЛВЗРР і ЛЕВЗ, ВАДДЛ не потребує додаткової оцінки та сигналів розряду для генерації результатів далі в каскаді. Кожна комірка в інших методах вимагає унікального оцінювання та сигналу розряду. Це означає, що накладні витрати, необхідні для керування вхідними та вихідними сигналами, значно зменшуються. Це вигідно, оскільки методи пом'якшення ДАП, такі як РЛ, мають труднощі з поширенням сигналу через схему через погіршення сигналу. Підхід ВАДДЛ використовує існуючі сигнали для оцінки та розрядки, що є перевагою перед ХДДЛ, RCCDL та ЛЗДМт. Таким чином, незважаючи на те, що кількість транзисторів вище в ВАДДЛ, додаткова потужність, необхідна для генерації сигналів оцінки та розряду в інших методах, полегшує атаку ДАП.

Представимо метод реалізації адіабатичної динамічної диференціальної логіки при підпороговій роботі для реалізації наднизької потужності. Щоб досягти цього, ми використовуємо метод зміщення переднього тіла, щоб зменшити середнє енергоспоживання, диференціальне енергоспоживання та дозволити використовувати коротші шлюзи. Цей метод є методом адіабатичної динамічної диференціальної логіки, зміщенаої за тілом (АДДЛЗ). Конструкція АДДЛЗ по суті така сама, як і конструкція ВАДДЛ. Різниця полягає в тому, що номінальна напруга становить 0,5 В, нижче порогової напруги транзисторів.

Було виконано моделювання на робочій частоті 13,56 МГц смарт-карти ISO/IEC 14443. Після основного зміщення 0,09 В покращення диференціальної потужності є незначним, однак величини стрибків зростають лінійно. Під час певних вихідних комбінацій пристрій намагається повернути сигнал до середньої точки через транзистори NMOS. Зміщення тіла постійно вирішує цю проблему. Середнє споживання електроенергії досягає мінімуму при 0,25 В. Отже, найгірше перемикання вихідного сигналу та середнього енергоспоживання оптимізується, коли зміщення тіла встановлено на середню точку.

Середня потужність пристрою АДДЛЗ на частоті 13,56 МГц становить $9{,}6093*10^{-10}$Вт; тому він споживає $6{,}3394*10^{-16}$Дж енергії за кожен цикл

обчислення. Це покращення на 88,82% у порівнянні з ВАДДЛ і на 97,4% у порівнянні з ЛЗДМт. Найвища пікова потужність перемикання пристрою становить $4{,}4800*10^{-8}$ Вт, а найменша пікова потужність перемикання — $3{,}9069*10^{-8}$ Вт, що дає диференціальну потужність $5{,}7305*10^{-9}$ Вт. Ця конструкція покращує 56,64% через ВАДДЛ. Отже, осередок АДДЛЗ покращує диференціальну потужність звичайного вентиля NAND на коефіцієнт 199,16.

Зміщення транзистора NMOS дозволяє зменшити його ширину до 300 нм. Крім того, це дозволяє зменшити довжину транзистора як для PMOS, так і для NMOS з 50 нм до 30 нм, що є мінімальною довжиною транзистора.

Компроміс у дизайні для ВАДДЛ і АДДЛЗ полягає в продуктивності проти низької потужності. Конструкція АДДЛЗ успішно моделює частоту 13,65 МГц, що є достатнім для стандарту ISO/IEC 14443. Однак сигнал швидко починає погіршуватися після 20 МГц через низьку напругу живлення. ВАДДЛ ефективно моделює на частотах, що перевищують 100 МГц.

3.3 Висновки

Результати моделювання показують, що наша квантова модель поведінки VHDL, заснована на інтегрованих вентилях Кубіт, є більш надійним і ефективним методом проектування, моделювання та перевірки оборотних логічних структур, ніж раніше існуючі моделі. Було також показано, що використання VHDL у цьому методі дозволяє використовувати надійну мову програмування, що забезпечує паралелізм із вбудованими можливостями цієї мови. Цей метод пропонує новий спосіб проектування та перевірки оборотних конструкцій. Можливість повторного використання конструкцій дозволяє спростити проектування більших складних конструкцій. Крім того, замість проектування цих структур виключно на квантовому представленні, цей метод надає нові засоби для проектування на вищих рівнях абстракцій, тобто дизайнер тепер може використовувати блок-схеми вищого рівня.

У цьому розділі представлено метод оптимізації синтезованих оборотних логічних структур, заснований на перетворенні попередньо синтезованої схеми в інтегровану бібліотеку Кубіт. Представлений метод заміни затворів їх реалізацією в ІК дозволив зменшити затримку шляхом заміни двох суміжних затворів меншою реалізацією, що забезпечує ту саму логіку обчислення. Це дозволяє нам об'єднувати квантові біти, щоб зменшити вартість, і видаляти біти, які лише сприяють обчисленню сміттєвих виходів. Синтезовані схеми все ще логічно оборотні за своєю природою, але зменшені з точки зору квантової вартості. Впровадження паралелізму у дизайн, дозволило успішно реалізували цей алгоритм O(N) раз. Продемонстровано, що квантова вартість схем може бути значно знижена.

Представлено новий алгоритм для синтезу адіабатичних логічних структур у CMOS. Результати синтезу показують, що в середньому запропонований алгоритм представляє покращення на 36% порівняно з найкращими відомими реверсивними конструкціями з оптимізованими осередками з двома шинами. Синтез двошинної адіабатичної схеми для шифру Rijndael показує покращення диференціальної потужності порівняно з одношиною та традиційною реалізаціями. Ці результати свідчать про те, що адіабатична логіка подвійної шини є багатообіцяючою методологією проектування для схем, де безпека є найважливішим проектним показником, а частоти нижчі, як, наприклад, стандарт смарт-карт ISO 14443.

Було використане пряме зміщення для покращення роботи адіабатичних логічних структур CMOS, а також пікової, середньої та диференціальної потужності при напрузі живлення 0,5 В та 1 МГц. Було вияснено, що адіабатичний інвертор ЛЕВЗ із зміщенням тіла може отримати до 96% покращення порівняно зі звичайним інвертором при 0,5 В. Інвертор зі зміщенням корпусу покращує диференціальну потужність звичайного інвертора при номінальній напрузі живлення в 16000 разів. Цей метод зміщення корпусу було застосовано до адіабатичних затворів Тоффолі та Фредкіна. Конструкції вдосконалені в порівнянні з реалізацією без зміщення тіла за всіма показниками

потужності, а також покращені вихідні сигнали. Конструкції покращили диференціальну потужність порівняно зі звичайними вентилями NAND та MUX на коефіцієнти 128,43 та 74,27 відповідно. Ці результати свідчать про те, що адіабатична логіка прямого тіла є перспективним методом проектування для застосувань із наднизьким енергоспоживанням. Запропоновано методологію розробки адіабатичної динамічної диференціальної логіки для пом'якшення атак ДАП на захищені інтегровані чіпи. Перша конструкція – це високоефективна адіабатична динамічна диференціальна логіка (ВАДДЛ), яка оптимізована для дуже високих робочих частот. Цей дизайн покращує представлені раніше тести [53] на 76,41% для середньої потужності завдяки зменшеній надійності оцінки та розрядних мереж. Комірка ВАДДЛ також покращила диференціальну потужність звичайного затвора NAND у 112 разів. Друга конструкція, АДДЛЗ, використовує зміщення прямого тіла для покращення часу перемикання та диференціальної потужності ультра-низьких потужностей. Було показано, що зміщення тіла на адіабатичних інверторах покращує диференціальну потужність звичайного інвертора в 16000 разів. Цей метод було реалізовано в АДДЛЗ, і результати моделювання показують, що диференціальну потужність було покращено на коефіцієнт 199,16.

4 УДОСКОНАЛЕННЯ МЕТОДУ ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ ПРОГРАМ ЗГІДНО ОБОРОТНОЇ ЛОГІКИ

4.1 Реалізація адіабатичної схеми

Шифрування захищає передані дані від модифікації або пошкодження за допомогою контрольних сум і хеш-функцій для автентифікації користувачів, сприяння неспростуванню та збереження конфіденційності. Шифрування маскує звичайний текст, перемішуючи його вміст у зашифрований текст, замінюючи символи через відображення відкритого тексту в інший елемент, а потім переставляючи елементи за допомогою перетворення. Пристрої, які використовують стандарти шифрування, такі як AES, стають об’єктом шахрайства та крадіжки через їх поширення та бажаний вміст. Зокрема, смарт-карти, які відповідають стандарту ISO 14443 [11, 15], сприйнятливі до атак із бічного каналу, які базуються на кореляції витоку вторинної інформації та вихідних сигналів ІС. У PI-пристроях це включає електромагнітне випромінювання (витік EM) [16], вимірювання кількості часу, необхідного для виконання операцій із закритим ключем [17], і аналіз шумового споживання енергії [18]. Однією з найефективніших атак є атака диференціального аналізу потужності (ДАП) [18], коли зловмисник аналізує споживання електроенергії в мікросхемі та порівнює її з вихідними сигналами мікросхеми. Ці атаки є ефективними, оскільки більшість сучасних обчислювальних технологій засновані на CMOS, а тенденції енергоспоживання цих пристроїв добре вивчені. Зменшення енергоспоживання схеми ускладнює атаку ДАП. Крім того, основною перешкодою при розробці ефективних безпечних апаратних пристроїв, таких як смарт-карти та медичні пристрої, є подолання обмежених ресурсів живлення та площі. Пристрої PI живляться від вхідного сигналу, тому методи маскування потужності, які споживають більше енергії, не є ідеальними. У імплантованих медичних пристроях заміна батареї вимагає інвазивної хірургії [87].

Представимо алгоритм оптимізації схеми:

1) перемикаємо вентилі в синтезованій схемі з їх еквівалентом у бібліотеці Кубіт;
2) відбувається перевірка двох шлюзів, які примикають один до одного, щоб перевірити, чи існує менша реалізація, яка може досягти того самого логічного обчислення з меншою вартістю;
3) отриманий граф розбивається на вузли наступним чином: задано оборотну схему з n входами та n виходами з вартістю c, отриманий граф матиме 2n+c-1 вузлів і n+2c ребер:

1) вузли від 0 до n-1 представляють входи;
2) вузли від n до n+c-1 представляють кожен квантовий вентиль у порядку їх затримки;
3) вузли від n+c до 2n+c-1 представляють виходи пристрою;
4) ребра мають напрямок, і оскільки алгоритм працюватиме від виходів до входів, напрямок йде від виходів до входів.
5) далі, обходимо вузли, які зустрічає кожен процес, і на основі типу вузла визначаємо, чи потрібно вузол видаляти. Якщо вузол видаляється, два вхідні дроти з’єднуються з вихідними;
6) після проходження контуру ми використовуємо два правила, щоб визначити, чи потрібно видалити ребро чи вузол:
    6.1) якщо жодне вхідне ребро не позначено, тоді відповідний вентиль не потрібен;
    6.2) якщо вузол має один позначений вхідний рядок, який є лінією керування, і відповідний рядок також є лінією керування, тоді шлюз не потрібний, і потрібно встановити значення «Наступний шлюз» попереднього вузла на значення шлюзу наступного вузла;
7) наступним кроком ми оптимізуємо схему, виконавши пошук з потрібних виходів.

Нещодавні дослідження адіабатичної логіки з подвійною шиною показали, що парадигма дизайну оборотної логіки є багатообіцяючою для пом’якшення атак ДАП шляхом значного зниження диференціального споживання електроенергії

схемою [19, 20]. Завдяки мінімізації стирання бітів у схемі та контролю потоку струму через схему, щоб мінімізувати розсіювання енергії через перемикання, екстремальні значення потужності є більш послідовними.

У цьому розділі стверджується, що адіабатична логіка подвійної шини є вигідною методологією проектування для ланцюгів малої потужності, де безпека є основним проектним показником на низькій частоти, такій як 13,56 МГц, що використовується в стандарті смарт-карт ISO 14443. Новизна удосконаленої схеми полягає в тому, що вона вперше включає методологію подвійної шини для маскування джерела живлення, адіабатичну логіку для зниження середньої потужності та транзисторну логіку для зменшення площі. Ця методологія дозволяє створювати однозначну, ефективну схему, яка також є фізично оборотною, дозволяючи повторне використання дизайну як для шифрування, так і для дешифрування з мінімальними накладними витратами, зменшуючи компроміс із площею, який зазвичай виникає при використанні адіабатичної логіки подвійної шини.

Частина ByteSub алгоритму Rijndael передбачає використання таблиці пошуку 16x16 для визначення результату. Перші 4 біти відповідають рядку таблиці, а наступні 4 біти відповідають стовпцю таблиці схеми. Значення таблиці є однозначними, тобто вхідне значення може бути однозначно визначено, знаючи результат. І оскільки результатом є 8-розрядний вихід, ми можемо побудувати повністю логічно та фізично оборотну схему, використовуючи адіабатичні схеми CMOS. Сигнали A-Hb представляють 8-розрядні входи та їх реалізацію з подвійною шиною, а сигнали P-Wb є вихідними значеннями та їх результатами з подвійною шиною. Результатом є те, що вхідні значення можна отримати шляхом розміщення вихідного сигналу на початковому виході.

Представлена конструкція вимагає 9612 транзисторів для прямого шифрування схеми. Порівняно з більшістю тестів, це значно вище – у деяких випадках понад 56 відсотків.

Однак перевага використання адіабатичного дизайну з двома шинами в цьому підході полягає в тому, що фізична оборотність дозволяє використовувати

ту саму схему як для шифрування, так і для дешифрування. Більшість реалізацій Rijndael вимагають прямих і зворотних апаратних реалізацій схеми. Метод вимагає лише одного додаткового керуючого сигналу та 32 вентилів Фредкіна для керування потоком через пристрій.

Ключ може проходити через зворотню схему в тому самому порядку, зменшуючи витрати на програмне забезпечення, а сама схема може використовуватися для шифрування та дешифрування, зменшуючи витрати на апаратне забезпечення.

Таким чином, використовуючи лише 256 додаткових транзисторів і один керуючий сигнал, ми можемо виконати дешифрування шляхом реверсування потоку через схему.

Синтезована двошинна адіабатична схема є функціонально правильною і працює на 13,56 МГц, стандартній частоті для дизайнів смарт-карт за стандартом ISO 14443.

4.2 Порівняння контрольних показників

Оскільки в багатьох представлених роботах у розробці AES Rijndael використовуються різні розміри технології реалізації, порівняння енергоспоживання та розміру реалізації не обов’язково є справедливим, оскільки 22-нм технологічна модель з низьким енергоспоживанням, така як модель, яку ми використовували для отримання наших результатів, за своєю суттю буде правильною. споживають менше електроенергії та площі, ніж шифр із використанням технологічної моделі 0,35 мкм, оскільки напруга живлення буде значно нижчою. Кількість транзисторів, кількість проводів і дисбаланс енергії, метрика, представлена в [31], забезпечує набагато більш справедливе порівняння між технологічними файлами. Враховуючи дві вхідні комбінації e_1 і e_2, варіацію між споживанням енергії (енергетичним дисбалансом) схеми протягом цих часових проміжків можна знайти за допомогою формули:

$$\left|\frac{e_1 - e_2}{e_1 + e_2}\right| * 100\%, \quad (4.1)$$

де e_1 і e_2 – вхідні комбінації.

Порівнюючи попередні тести (Таблиця 4.1), ми використовували різні технологічні файли та робочі частоти, щоб забезпечити парне й ретельне порівняння. За допомогою Design Compiler ми змогли створити файли, оптимізовані за потужністю та площею, використовуючи моделі TSMC для 0,35 мкм і 0,25 мкм, файли IBM для 0,18 мкм, 0,13 мкм і 90 нм, а також файли PTM для 65 нм, 45 нм, 32 нм і 22 нм технології на різних частотах. Також, була синтезована схема у кількох технологіях, щоб показати, що запропонований підхід є ефективним, оскільки технологія масштабується. Безпека імплантованих медичних пристроїв має продемонструвати довгострокову ефективність, оскільки атаки стають лише сильнішими із законом Мура [8].

У таблиці 4.1 наведено порівняння технології реалізації, напруги живлення, одиночної або подвійної шини, транзисторів, необхідних для шифрування та дешифрування схеми, загальної площі, середнього споживання енергії та енергетичного дисбалансу. Покращення запропонованого методу в порівнянні з попереднім найкращим випадком енергетичного дисбалансу, CSSAL в [8], становить 41% на 50 МГц.

Таблиця 4.1. Порівняння тестів удосконаленої схеми

Еталон	Одинич-ний/ Подвій-ний	Техн., нм	Входи джерела живлення (В)	Частота (МГц)	Кіль-кість шлюзів	Всьо-го	Площа нм2	Електри-чна сила (Ватт)
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
[343] (Відкрите ядро)	Один.	0.35	3.3	333	3092	6272	44593	14000
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
[343] (comp)	Один.	0.35	3.3	333	2242	4606	101364	32100
[343] (Comp)	Подв.	0.35	3.3	333	4442	9072	68603	8906

Запропоно-вано	Подв.	*0.35*	*3.3*	*50*	*512*	*10124*	*25663*	*1037*
Запропоновано	Подв.	*0.35*	*3.3*	*13.56*	*512*	*10124*	*25663*	*446.2*
Запропоновано	Подв.	*0.35*	*3.3*	*1*	*512*	*10124*	*25663*	*6.59*
[344] (Без маскування)	Подв.	0.25	2.5	13.56	n/a	n/a	n/a	165
[344] (Маскування)	Подв.	0.25	2.5	13.56	n/a	n/a	n/a	210
[353]	Одини.	0.25	2.5	10	2976	7856	8560	122
[354]	Один.	0.25	2.5	10	3316	8880	8150	416
Запропоновано	Подв.	*0.25*	*2.5*	*13.56*	*512*	*10124*	*6874.2*	*143.7*
Запропоновано	Подв.	*0.25*	*2.5*	*10*	*512*	*10124*	*6874.2*	*106.2*
[349] (LUT)	Один.	0.18	1.8	0.1	1361	2566	n/a	1.85
[348] (уніфікований)	Один.	0.18	1.8	0.1	1181	2468	n/a	1.85
[348] (Bitslice)	Один.	0.18	1.8	0.1	1192	2369	n/a	1.85
Запропоновано	Подв.	*0.18*	*1.8*	*0.1*	*512*	*10124*	*3474*	*0.342*
[356] (CSSAL)	Подв.	0.18	1.8	1.25	8063	16178	n/a	4.187
[356] (ЛЕВЗ)	Один.	0.18	1.8	1.25	3270	6436	n/a	6.375
Запропоновано	Подв.	*0.18*	*1.8*	*1.25*	*512*	*10124*	*3474*	*0.69*
[356] (CSSAL)	Один.	0.18	1.8	12.5	8063	16178	n/a	83.375
[356] (ЛЕВЗ)	Один.	0.18	1.8	12.5	3270	6436	n/a	172.75
Запропоновано	Подв.	*0.18*	*1.8*	*12.5*	*512*	*10124*	*3474*	*26.97*

Конструкція є вигідною з точки зору найгіршого випадку EI та Pavg для всіх тестів. У більшості випадків область шифрування значно вища, як ми й очікували. Фізична оборотність запропонованої нами структури суттєво зменшує компроміс

із областю, якщо також враховувати накладні витрати на область, необхідні для дешифрування схеми. Це особливо вигідно порівняно з CSSAL, який фізично не є однозначним, що призводить до значних витрат транзистора на дешифрування.

4.3 Висновки

Ми пропонуємо методологію розробки адіабатичної динамічної диференціальної логіки для пом’якшення атак ДАП на захищені інтегровані чіпи. Представлений адіабатичний S-блок із подвійною шиною для реалізації в шифрі Rijndael для алгоритму AES для додатків із низьким енергоспоживанням і низькою частотою, таких як смарт-карти на 13,56 МГц. Було синтезовано та змодельовано запропоновану схему в численних технологіях реалізації, щоб забезпечити чесне та точне порівняння між попередньо синтезованими тестами. Схема покращує енергетичний дисбаланс порівняно з попередньою роботою в середньому на 65 відсотків, що робить запропоновану схему ефективним засобом пом’якшення атак ДАП. Компромісом є робоча частота. Початковий компроміс у площі компенсується використанням оборотної властивості адіабатичного контуру з подвійною шиною, щоб дозволити повторне використання дизайну як для шифрування, так і для дешифрування, що фізично неможливо в усіх попередніх тестах.

ВИСНОВКИ

У кваліфікаційній роботі магістра за результатами виконаних теоретичних та практичних досліджень було розроблено метод забезпечення безпеки програм згідно адіабатичної динамічної диференціальної логіки для додатків у безпечному дизайні для пом’якшення атак ДАП.

У першому розділі проведено аналіз математичних моделей із оборотної логіки, розглянуті основи оборотної та адіабатичної логіки. Також, в роботі звернуто увагу на смарт-картки (невеликі інтегровані схеми, вбудовані в пластик або жетони, які використовуються для автентифікації, ідентифікації та зберігання персональних даних). Розглянуто використання енергоспоживання для отримання компрометуючої інформації відомого як атака диференціального аналізу потужності (ДАП), коли зловмисник аналізує інформацію, отриману з деталей практичної реалізації безпечних алгоритмів. Розглянуто алгоритм Rijndael для забезпечення ефективного захисту переданих і збережених даних від криптоаналітичних атак.

У другому розділі розглянуто допустимість послідовної логіки в оборотних обчислювальних системах. Представлено результат моделювання послідовної оборотної логічної структури, запропоноване покращення пристрою вихідної пам'яті CMOS та структури SRAM для покращення квантових характеристик пристрою. Також, представлені результати симуляції прикладу адіабатичної логіки, де бінарна комутаційна мережа розсіює менше kTln(2) джоулів енергії за подію комутації. Показано, що енергія біта більша, ніж kTln(2), але результуюча дисипація менша, ніж kTln(2).

Третій розділ магістерської роботи включає розгляд методу оптимізації синтезованих оборотних логічних структур, заснований на перетворенні попередньо синтезованої схеми в інтегровану бібліотеку реверсивної логіки Кубіт (бібліотеки, що є надійною багатозначною логічною системою для визначення заданих детермінованих квантових станів і узагальнення недетермінованих станів). Проведено синтез двошинної адіабатичної логіки. Представлено алгоритм

для синтезу адіабатичних логічних структур у CMOS. Було використане пряме зміщення для покращення роботи адіабатичних логічних структур CMOS, а також пікової, середньої та диференціальної потужності при напрузі живлення 0,5 В та 1 МГц. Було вияснено, що адіабатичний інвертор ЛЕВЗ із зміщенням тіла може отримати до 96% покращення порівняно зі звичайним інвертором при 0,5 В. Інвертор зі зміщенням корпусу покращує диференціальну потужність звичайного інвертора при номінальній напрузі живлення в 16000 разів.

У четвертому розділі запропоновано методологію розробки адіабатичної динамічної диференціальної логіки для пом'якшення атак ДАП на захищені інтегровані чіпи. Представлена удосконалена адіабатична схема із подвійною шиною для реалізації в шифрі Rijndael для алгоритму AES для додатків із низьким енергоспоживанням і низькою частотою, таких як смарт-карти на 13,56 МГц. Було синтезовано та змодельовано запропоновану схему в декількох технологіях реалізації. Схема покращує енергетичний дисбаланс порівняно з попередньою роботою в середньому на 65 відсотків, що робить запропоновану схему ефективним засобом пом'якшення атак ДАП.

За темою кваліфікаційної роботи магістра опубліковано статтю на тему «Метод забезпечення безпеки програм згідно оборотної логіки» в матеріалах конференції X International Scientific and Practical Conference Science, Education, Innovation Topical Issues and Modern Aspects, 2024, Tallin, Estonia, що індексуються в наукометричній базі Google Scholar [88], (Додаток А) та отриманий сертифікат участі у конференції з кількісттю годин дистанційної роботи – 12 годин (0,4 ECTS credits) (Додаток Б).

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Toffoli T. Reversible Computing, *Technical Report MIT/LCS/TM-151*, 1980.
2. Feynman R. Simulating Physics with Computers. *International Journal of Theoretical Physics*. 1982.
3. Born M., Fock V. A. Beweis des Adiabatensatzes. *Zeitschrift für Physik.* 1928. Vol.A 51. No. 3–4. Pp. 165–180.
4. Kato T. On the Adiabatic Theorem of Quantum Mechanics. *Journal of the Physical Society of Japan.* 1950. vol.5. no.6. pp. 435–439.
5. Mohammadi M., Eshghi M. Behavioral description of quantum V and V+ gates to design quantum logic circuits. *Systems, Signals and Devices, 2008. IEEE SSD 2008. 5th International Multi-Conference*, 2008, vol.2. pp.1-5, 20-22.
6. Morrison M., Ranganathan M. Synthesis and Optimization of Dual Rail Adiabatic Logic Using A Novel Parallel Algorithm. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2013.
7. Sasanian Z., Saeedi M., Sedighi M., Samani M. A Cycle-Based Synthesis Algorithm for Reversible Logic. *Design Automation Conference*, 2009, pp. 745-750.
8. Maslov D., Dueck G.W., Miller D.M. Synthesis of Fredkin-Toffoli reversible networks. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*. 2005. vol.13. no.6. pp.765-769.
9. Wang Hanwu Chen D., Zhu W. Bidirectional matrix-based algorithm for 4-Qubit reversible logic circuits synthesis. *Evolutionary Computation (CEC), IEEE Congress on*, 2010 vol., no., pp.1-5, 18-23.
10. Ярецька Н.О., Кучерук Д.В. Моделювання системи контакту двох співвісних ідентичних попередньо напружених циліндрів та шару з початковими напруженнями. *Proceedings of the XVI International Scientific and Practical Conference. Prague*, Czech Republic. 2023. Pp. 407-410. URL: https://isg-konf.com/methods-of-solving-complex-problems-in-science.
11. Feynman R. Space-Time Approach to Quantum Electrodynamics. *Physical Review*, 1949. vol. 76,

12. Clausius R. On a Modified Form of the Second Fundamental Theorem in the Mechanical Theory of Heat. *Philosophical Magazine and Journal of Science*, 1856. vol. 12, iss. 77, pp. 81-98.
13. Clausius R. On Several Convenient Forms of the Fundamental Equations of the Mechanical Theory of Heat. *Philosophical Magazine and Journal of Science*. 1865.
14. Boltzmann L. On the Relation Between the Second Fundamental Law of the Mechanical Theory of Heat and the Probability Calculus with Respect to the Theorems of Heat Equilibrium. *Wiener Berichte*, 1877.
15. Schrödinger E. Quantization as a Problem of Proper Values. Part I. *Annalen der Physik*, 1926. vol. 79.
16. Feynman R. Mathematical Formulation of the Quantum Theory of Electromagnetic Interaction. *Physical Review*, 1950. vol. 80.
17. Turing A. M. On Computable Numbers, with an Application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 1937. vol. s2-42, iss. 1, pp. 230-265.
18. Landauer R. Irreversibility and Heat Generation in the Computational Process. *IBM Journal of Research and Development*, 1961. vol. 5. pp. 183-91.
19. Swanson J.A. Physical versus Logical Coupling in Memory Systems. *IBM Journal*, Jul. 1960. vol. 4. no. 3, pp. 305-310.
20. Bennett C. Logical Reversibility of Computation. *IBM Journal of Research and Development*. 1973. vol. 17. pp. 525-532.
21. Keyes R.W. Landauer R., Minimal energy dissipation in logic. IBM Journal. 1970. Vol. 14. no. 2. pp. 152-157.
22. Davis M. Computability and Unsolvubility. *McGraw-Hill Book Co.*, Inc., New York. 1958. pp. 25-26.
23. Morrison M., Lewandowski M., Ranganathan N. Design of a Tree-Based Comparator and Memory Unit Based on a Novel Reversible Logic Gates. *ISVLSI*, 2012. pp. 231-236.

24. Yang G. W., Song X. Y., Hung W. N. N., Perkowski M. Fast synthesis of exact minimal reversible circuits using group theory. *Proceedings of IEEE ASP-DAC 2005*. Shanghai, China. 2005. vol. 2, pp. 18-21.
25. Arabzadeh M., Saeedi M., Zamani M.S. Rule-based optimization of reversible circuits. *Design Automation Conference (ASP-DAC), 2010 15th Asia and South Pacific*. 2010. vol. no. pp.849-854, 18-21
26. Wille R., Große D., Teuber L., Dueck G.W., Drechsler R. RevLib: An Online Resource for Reversible Functions and Reversible Circuits. *Int'l Symp. On Multi-Valued Logic*, 2008.
27. Kerntopf P. Synthesis of multipurpose reversible logic gates. *Digital System Design, Proceedings. Euromicro Symposium*. 2002. vol., no., pp. 259- 266.
28. Kerntopf P. Binary decision diagram based on single and multiple generalized Shannon expansions. *In Proc. 6' Internalionel Symposium on Representotions and Methodologv of Future Computing Technologv, Trier*, Germany, March 2003, vol. 183. Pp.190.
29. Mohammadi M., Eshghi M., Bahrololoom A. Behavioral model of V and V+ gates to implement the reversible circuits using quantum gates. *TENCON 2008 - IEEE Region 10 Conference*. 2008. vol. no. pp.1-6, 19-21
30. Frank M.P. Common mistakes in adiabatic logic design and how to avoid them. *Embedded Systems and Applications*, 2003. Pp.216-222,
31. Yibin Ye Y., Roy K. Energy recovery circuits using reversible and partially reversible logic. *IEEE Trans. Circuits and Systems I*. 1996. vol.43, no.9, pp.769-778.
32. Hisakado T., Iketo H., Okumura K. Logically reversible arithmetic circuit using pass-transistor. *in ISCAS*, 2004. vol.2. pp. 853-6,
33. Younis S. G. Asymptotically Zero Energy Computing Using Split-Level Charge Recovery Logic. *Ph.D. Dissertation, Massachusetts Institute of Technology*, 1994.
34. Van Rentergem Y., De Vos A. Optimal design of a reversible full adder. *International Journal of Unconventional Computing*. 2005. Vol.1. no.4. pp.339.

35. Szilard L. On the Decrease of Entropy in a Thermodynamic System by the Intervention of Intelligent Beings. *Z. Phys*. 1929. Vol.53, pp. 840
36. Kerntopf P. Binary decision diagram based on single and multiple generalized Shannon expansions. *In Proc. 6' Internalionel Symposium on Representotions and Methodologv of Future Computing Technologv*. Trier, Germany. 2003. Vol.183. pp.190.
37. Jing H., Ma G., Feng G. Efficient Algorithm for Positive-polarity Reedmuller Expansions of reversible circuits. *Microelectronics,* 2006. *ICM '06. International Conference on*, 2006. vol., no., pp.63-66,
38. Guan Z., Qin X., Ge Z., YIKing Z. Reversible Synthesis with Minimum logic function. *Computational Intelligence and Security, 2006 International Conference on*. 2006. vol.2, no., pp.968-971
39. Tiri K., Akmal M., Verbauwhede I. A dynamic and differential cmos logic with signal independent power consumption to withstand differential power analysis on smart cards. *ESSCIRC*. 2002. pp. 403–406.
40. Ramakrishnan L. SDMLp - Secure Differential Multiplexer Logic : Logic Design For DPA Resistant Cryptographic Circuits. *M.S. Thesis, University of Cincinatti*, 2012.
41. Tiri K., Verbauwhede I. A logic level design methodology for a secure DPA resistant ASIC or FPGA implementation. *in Proc. DATE*, 2004, pp. 246–251.
42. Sundaresan V., Rammohan S., Vemuri R. Power invariant secure-IC design methodology using reduced complementary dynamic and differential logic. i*n Very Large Scale Integration*. VLSI – SoC, 2007. Oct. 2007, pp. 1–6.
43. Ramakrishnan L.N., Chakkaravarthy M., Manchanda A.S., Borowczak M., Vemuri, R. SDMLp: On the use of complementary Pass transistor Logic for design of DPA resistant circuits. *Hardware-Oriented Security and Trust (HOST)*, 2012.
44. Sana P.K., Satyam M. An Energy Efficient Secure Logic to Provide Resistance against Differential Power Analysis Attacks. *Electronic System Design (ISED), 2010 International Symposium*. 2010.vol.3, no.4. pp.61,65, 20-22

45. Takahashi Y., Sekine T., Yokoyama M. A comparison of adiabatic logic as a countermeasures against power analysis attacks. *System Science and Engineering (ICSSE), 2010 International Conference* , 2010, vol.1, no.3. pp.615,618.
46. Sana P.K., Satyam M. A low power secure logic style to counteract differential power analysis attacks. *VLSI Design, Automation and Test (VLSI-DAT). International Symposium.* 2011, vol.23 no.7. pp.1,4.
47. Bai X., Huang L., Wang Y., Hu X. Evaluation of ДАП Attack Resistance of Transistor-Based Adiabatic Logic Styles. *e-Business and Information System Security (EBISS), 2010 2nd International Conference on*.2010. pp.1,3, 22-23
48. Tessier R., Jasinski D., Maheshwari A., Natarajan A., Weifeng X., Burleson W. An energy-aware active smart card. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions.* 2005. vol.13, no.10, pp.1190,1199,
49. Monteiro C., Yasuhiro T., Toshikazu S. Low power secure AES Sbox using adiabatic logic circuit. *Faible Tension Faible Consommation (FTFC), 2013 IEEE*, 2013. vol.6, no.3. pp.1,4
50. Mok K-K., Chan C-F. A 13.56 MHz adiabatic smart card / RFID. *ASIC, 2007. ASICON '07. 7th International Conference.*2007. vol.25. no.5. pp.874,877
51. Hisakado T., Iketo H., Okumura K. Logically reversible arithmetic circuit using pass-transistor. *in ISCAS*, 2004. vol.2. pp. 853-6,
52. Diffie W., Hellman M. New directions in cryptography. *Information Theory, IEEE Transactions.*1976. Vol.22. no.6. pp. 644-654.
53. Dickinson A.G., Denker J.S. Adiabatic dynamic logic. *Solid-State Circuits, IEEE Journal.* 1995.vol.30, no.3. pp.311,315
54. Merkle R.C. Towards Practical Reversible Logic. *Physics and Computation, 1992. PhysComp '92., Workshop*,1992. vol.22. no.9. pp.227-228
55. De V.K., Meindl J.D. A dynamic energy recycling logic family for ultra-low-power gigascale integration (GSI). *Low Power Electronics and Design, 1996., International Symposium*.1996. vol.18. no.6. pp.371,375.

56. Yang M.M. Barby J.A. A novel fast low voltage dynamic threshold true single phase clocking adiabatic circuit. *Circuits and Systems, 2004. ISCAS '04*. 2004. vol.2, no.4. pp.289- 92,
57. Monteiro C., Takahashi Y., Sekine T. Resistance against power analysis attacks on adiabatic dynamic and adiabatic differential logics for smart card. *Trans. ISPACS,* 2011, vol.14. no.4. pp.1,5.
58. Bérut A., Arakelyan A., Petrosyan A., Ciliberto S., Dillenschneider R., E. Lutz. Experimental verification of Landauer's principle: linking information and thermodynamics. *Nature*, 2012. vol. 483. pp. 187-189.
59. Li J., Zhang Y., Yoshihara T. A novel charge recovery logic structure with complementary pass-transistor network. *SoC Design Conference (ISOCC), 2012 International*. 2012. pp.17,20, 4-7
60. Newsome J., Song D. X. Dynamic Taint Analysis for Automatic Detection, Analysis, and Signature Generation of Exploits on Commodity Software. *In the Proceedings of the Network and Distributed System Security Symposium*. San Diego, CA. 2005.
61. Crandall J. R., Chong F. T. MINOS: Control Data Attack Prevention Orthogonal to Memory Model. *In the Proceedings of the 37th Intl. Symposium on Microarchitecture*. Portland, 2004.
62. Cowan C. StackGuard: Automatic adaptive detection and prevention of buffer-overflow attacks. *Proceedings of the 7th USENIX Security Symposium*. 1998. Vol. 81.
63. Frantzen M., Shue M. Stackghost: Hardw are f acili- tated stack protection. *In Proc. 10th USENIX Security Symposium.* Washington, D.C. 2001.
64. Chen S., Xu J., Nakka N., Kalbarczyk Z., Iyer R. K. Defeating Memory Corruption Attacks via Pointer Taintedness Detection. *In the Proceedings of the International Conference on Dependable Systems and Networks (DSN).* Yokohama, Japan, June 2005.
65. NIST Federal Information Processing Standards (FIPS) PUB 197 *Advanced Encryption Standard*.. 2001.

66. Mehdi S., Igor L. Markov. Synthesis and optimization of reversible circuits—a survey. *ACM Computing Surveys (CSUR)* 2013. Vol.45. no.2. pp. 21.
67. Nechvatal J. Report on the Development of the Advanced Encryption Standard. 2000.
68. Thiagarajan E., Gourishetty M. Study of AES and its Efficient Software Implementation.2003.
69. Fredkin E., Toffoli T. Conservative Logic. *International Journal of Theoretical Physics*. 1982, vol. 21, nn. 3-4, pp 219-53.
70. Deustch D. Quantum Computational Networks. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences* . 1989. vol. 425. 1868. pp. 73- 90.
71. Nielsen M., Chuang I. Quantum Computation and Quantum Information. Cambridge Univ. Press, 2000.
72. Morrison M., Ranganathan N. Analysis of Reversible Logic Based Sequential Computing Structures Using Quantum Mechanics Principles. *ISVLSI*, 2012. pp. 219- 224.
73. DiVincenzo D. P., Smolin J. A. Results on two-bit gate design for quantum computers. Proceedings of the Workshop on Physics and Computation. *PhysComp*, 1994.
74. Lewandowski M., Ranganathan N., Morrison M. Behavorial Modeling of Integrated Кубіт Gates in Quantum Reversible Logic Structures. *To Appear in ISVSLI 2013*. 2013.
75. Boechler G., Whitney J., Lent C., Orlov A., Snider G. Comment on `Fundamental limits of energy dissipation in charge-based computing. *Appl. Phys*. Lett. 98, 2011.
76. Filanovsky I. M, Sinencio S., Ram rez-Angulo E., Thapliyal J., Ranganathan H., Soderstrand N., Silva-Martinez M.A., Yun Chiu Gopalan J., Acharya K., Saxena V., Salama V., K.N. "[Seven tutorials]," *Circuits and Systems (MWSCAS). IEEE 55th International Midwest Symposium on*. 2012.
77. Standard 1076-1993. IEEE Standard VHDL Language Reference Manual. *IEEE*, 1993. p. 249,

78. Barenco A., Bennett C., Cleve R., DiVincenxo D., Margolus N., Shor P., Sleator T., Smolin J., Weinfurter H. Elementary gates for quantum computation. *Phys Rev. A*, 1995. Vol. 53, pp. 3457-3467.
79. Peres A. Quantum Theory. *Concepts and Methods*, (Kluwer, 1993), Chap. 8.6.
80. Milburn G. Quantum Optical Fredkin Gate. *Physical Review Letters*, 1989. vol.62, no.18. pp. 2124-2127.
81. Picton P. A Universal Architecture for Multiplevalued Reversible Logic. *MVL Journal*. 2000. *vol*.5. pp.27-37.
82. Snider G., Blair E., Boechler G., Thorpe C., Bosler N., Wohlwend M., Whitney J., Lent C., Orlov A. Minimum energy for computation, theory vs. experiment. *IEEE-NANO*. 2011. pp.478-481
83. Morrison M., Ranganathan D. Forward Body Biased Adiabatic Logic for Peak and Average Power Reduction in 22nm CMOS. *IEEE Symposium on Very Large Scale Integration and Design*. 2014.
84. Moon Y., Jeong D. An efficient charge recovery logic circuit. *IEEE Journal of Solid State Circuits*. 1996. Vol.31. no.4. pp. 514-522.
85. Tschanz J., Kao J., Narendra S., Nair R., Antoniadis D., Chandrakasan, A., De V. Adaptive body bias for reducing impacts of die-to-die and within-die parameter variations on microprocessor frequency and leakage. *Solid-State Circuits, IEEE Journal of*. 2002. vol.37, no.11, pp. 1396- 1402
86. PTM 22nm Low Power HSPICE Model. ptm.asu.edu/modelcard/LP/22nm_LP.pm.
87. D. M. Miller and G. W. Dueck. Spectral technIKues for reversible logic synthesis. *In 6th Intemotional Symposium on Representations and Methodology of fiture Computing Technologzes*. 2003.
88. Савенко О., Кучерук Д., Ярецька Н. Метод забезпечення безпеки програм згідно оборотної логіки. *Scientific Collection «InterConf»*, (196): with the Proceedings of the 10th International Scientific and Practical Conference «Science, Education, Innovation: Topical Issues and Modern Aspects» (Tallinn, Estonia, April 16-18 квіт. 2024 р.) comp. by LLC SPC «InterConf». Tallinn: Ühingu Teadus juhatus, 2024. Pp. 388-393. URL: https://elar.khmnu.edu.ua/handle/123456789/15889.

ДОДАТОК А

КОПІЯ ПУБЛІКАЦІЇ У ВИДАННІ, ЩО ІНДЕКСУЄТЬСЯ В НАУКОМЕТРИЧНІЙ БАЗІ GOOGLE SCHOLAR

Proceedings of the 10th International Scientific and Practical Conference «Science, Education, Innovation: Topical Issues and Modern Aspects» (April 18-19, 2024). Tallinn, Estonia

110
196

INTERCONF

INFORMATION AND WEB TECHNOLOGIES

Метод забезпечення безпеки програм згідно оборотної логіки

Савенко Олег Станіславович[1], Кучерук Дмитро Віталійович[2], Ярецька Наталія Олександрівна[3]

[1] д.т.н., професор, декан факультету інформаційних технологій; Хмельницький національний університет; Україна

[2] здобувач магістерського ступеня, 2 курс, кафедра комп'ютерної інженерії та інформаційних систем; Хмельницький національний університет; Україна

[3] к.ф.-м.н., доцент, доцент кафедри вищої математики та комп'ютерних застосувань; Хмельницький національний університет; Україна

Анотація. В роботі розглянуто оборотну логіку в CMOS та запропоновано реалізацію адіабатичної динамічної диференціальної логіки для додатків для більш сильного пом'якшення атак DPA.
Ключові слова: *оборотна логіка, адіабатична динамічна диференціальна логіка, CMOS.*

Вступ. Виробництво економічно ефективних безпечних інтегрованих мікросхем, таких як смарт-карти, вимагає від розробників апаратного забезпечення врахування компромісів у розмірі, безпеці та енергоспоживанні. Для створення успішних проектів, орієнтованих на безпеку, апаратне забезпечення низького рівня повинно містити вбудовані механізми захисту, які доповнюють криптографічні алгоритми, такі як AES і Triple DES, запобігаючи атакам на бокових каналах, таким як диференціальний аналіз потужності (DPA). Динамічна логіка затьмарює вихідні сигнали та роботу схеми, знижуючи ефективність атаки DPA. Тому було запропоновано реалізацію адіабатичної динамічної диференціальної логіки для додатків у безпечному дизайні для більш сильного пом'якшення атак DPA.

Оборотні логічні структури. Принципи квантової механіки керують фізичними обмеженнями обчислювальних схем і систем. Ці системи розсіюють енергію через стирання бітів у своїх взаємопов'язаних примітивних структурах, що є важливим фактором, оскільки щільність транзисторів зростає. Збільшення ентропії в цих середовищах безпосередньо пов'язане з ймовірністю того, що квантова частинка займе будь-який із

Продовження Додатку А

Proceedings of the 10th International Scientific and Practical Conference «Science, Education, Innovation: Topical Issues and Modern Aspects»
(April 16-18, 2024).
Tallinn, Estonia

110
196

INFORMATION AND WEB TECHNOLOGIES

своїх станів. Щоб створити ідеальний універсальний комп'ютер, який розсіює довільно низьку енергію, має бути реалізована обертова логіка, оскільки закони фізики вказують на оборотність у часі.

Оборотні логічні структури є задовільними для проектування та реалізації в обчислювальних структурах та організації, коли ці правила проектування забезпечують оборотність логічної структури [1]. Таким чином, універсальна обчислювальна машина може бути реалізована для того, щоб ідеально моделювати кожну кінцево реалізовану фізичну систему, оскільки кожен електрон у квантовому комп'ютері представлений постійним унітарним оператором у гамільтоновому просторі [2].

Основний принцип оборотного обчислення полягає в тому, що біективний пристрій з однаковою кількістю вхідних і вихідних ліній не матиме розсіювання тепла. Електродинаміка системи дозволяє передбачити всі майбутні стани на основі відомих минулих станів, і система досягає кожного можливого стану. Є дві окремі, але однаково важливі парадигми оборотної логіки. По-перше, це логічна оборотність, яка використовує принципи, які керують оборотною логічною структурою, щоб визначити логічні обчислення, необхідні для здійсненних проектів. По-друге, це фізична оборотність, яка передбачає розробку фізичної структури, вхідні значення якої можуть однозначно визначатися виходом кожного обчислювального циклу, і розсіювання енергії якої не перевищує бар'єр Ландауера (kTln(2) джоулів на обчислювальний цикл). Різниця між цими двома парадигмами є важливою, оскільки логічно-обертова структура все ще може перевищувати бар'єр Ландауера. Наприклад, CMOS – це інвертор, розроблений за технологією, що працює при кімнатній температурі, VDD якого становить 1 В і має вихідну ємність 100 пФ, буде розсіювати $5 \cdot 10^{-11}$ Джоулів на перехід стану, що в $1{,}75 \cdot 10^{10}$ разів більше, ніж kTln(2), навіть якщо інвертори логічно оборотні.

Реалізацією оборотної логіки в CMOS, де струм (що протікає через схему) контролюється, щоб мінімізувати розсіювання енергії через перемикання є – адіабатична логіка. Існують значні дослідження щодо проектування та аналізу локально оптимальних адіабатичних елементів для пом'якшення атак бічних каналів. Однак жодна з цих робіт не розглядала використання адіабатичної логіки в реалізації гнучких і програмованих політик апаратної безпеки. Адіабатична логіка також не застосовувалася в додатках апаратної безпеки, таких як надійні системи голосування та стандарти шифрування даних.

Продовження Додатку А

Proceedings of the 10th International Scientific and Practical Conference «Science, Education, Innovation: Topical Issues and Modern Aspects» (April 18-19, 2024). Tallinn, Estonia

110
196

INFORMATION AND WEB TECHNOLOGIES

Адіабатичну теорему вперше представили Борн і Фок [3]. Вони описують фізичну систему такою, що залишається у своєму миттєвому власному стані, якщо дане збурення діє на неї досить повільно і якщо існує розрив між власним значенням і рештою спектра гамільтоніана. Тому, уповільнюючи зміну умов системи, система сама адаптується до нової конфігурації, змінюючи щільність ймовірності. Це означає, що якщо система починається у власному стані початкового гамільтоніана, вона закінчиться у відповідному власному стані кінцевого гамільтоніана [4].

В роботі розглянуто питання про те, чи можна маніпулювати схемами перемикання потоку електронів оборотно за допомогою логічних структур CMOS. Представлені результати симуляції прикладу адіабатичної логіки, де бінарна комутаційна мережа розсіює менше kTln(2) джоулів енергії за подію комутації. Також, розглянуто допустимість послідовної логіки в оборотних обчислювальних системах. Я представляють суто математичний доказ того, що послідовні оборотні логічні структури фізично можливі. Розроблено набір обчислювальних логічних примітивів, щоб забезпечити підтримку фізичної оборотності для всіх оборотних логічних структур із шляхами зворотного зв'язку.

Диференціальний аналіз потужності. Використання енергоспоживання для отримання компрометуючої інформації відоме як атака диференціального аналізу потужності (DPA). Зловмисник аналізує інформацію, отриману з деталей практичної реалізації безпечних алгоритмів [5]. Більшість сучасних обчислювальних систем використовують технологію CMOS, і динамічне споживання енергії вентилем CMOS пропорційне його вхідним сигналам. Таким чином, аналіз вихідної потужності дозволяє зловмиснику визначити кореляцію між даними та ключем, оскільки перемикання в вентилях CMOS залежить від цих вхідних даних. Коли зловмиснику відомий відкритий текст і круглий підключ, він може визначити вхідні дані для логічної функції та вивести їх вихід за допомогою таблиці пошуку. Алгоритми відкритого ключа можна аналізувати за допомогою DPA шляхом співвіднесення значень кандидатів для проміжних обчислень із вимірюванням енергоспоживання. Для операцій модульного піднесення до степеня можна перевірити припущення бітів експоненти, перевіривши, чи співвідносяться прогнозовані проміжні значення з фактичним обчисленням.

Ефективність атаки DPA можна продемонструвати за допомогою простого звичайного інвертора. Напруга живлення звичайного інвертора CMOS становить 0,95 В при 1 МГц. Середня потужність звичайного інвертора становить $2{,}0617 \cdot 10^{-8}$ Вт, з

Продовження Додатку А

Proceedings of the 10th International Scientific and Practical Conference «Science, Education, Innovation: Topical Issues and Modern Aspects» (April 16-18, 2024). Tallinn, Estonia

110
196

INTERCONF

INFORMATION AND WEB TECHNOLOGIES

піковим підвищенням $P_{Peak_{rise}}$ =4,5604·10^{-6} Вт, та піковим падінням $P_{Peak_{fall}}$ =1,0325·10^{-5} Вт, викликаючи P_{diff} =5,7644·10^{-6} Вт. Це означає, що пік потужності, коли вхід перемикається з 1 на 0, становить 5,7644·10^{-6} Вт перевищує пікову потужність, коли вхід перемикається з 0→1. Тому зловмисник може правильно визначити логічне розташування схеми.

Основним недоліком усунення DPA-атак на програмному рівні є те, що зміни потужності та струму, які аналізуються зловмисником, відбуваються на апаратному рівні, і жоден програмний алгоритм, яким би ефективним він не був, не може вплинути на роботу затвору CMOS після отримання вхідного сигналу. Наприклад, вставлення випадкових переривань процесу для запобігання послідовній роботі алгоритму [6] можна обійти методами повторної синхронізації та інтеграції [5]. Крім того, бітове маскування [7] можна подолати за допомогою атак DPA.

Запобігання атакам DPA. Таким чином, найефективніший підхід до запобігання атакам DPA полягає в тому, щоб включити логіку на основі безпеки в саму апаратну реалізацію, щоб ускладнити для зловмисника визначення необхідної інформації для визначення вхідних даних. Три найважливіші показники, які слід враховувати при розробці схем CMOS для цієї мети, це споживана потужність, площа та робоча частота, оскільки

$$E_{diss}=C_L \cdot V_{dd}^2 \cdot f,$$

де C_L – ємність навантаження, V_{dd} – напруга живлення, а f – робоча частота.

В роботі також розглядається два алгоритми синтезу для оборотної та адіабатичної логіки: 1) надійну модель поведінки для фундаментального затвору Integrated Qubit (IQ) для проектування локально оборотних логічних структур; 2) метод оптимізації для оборотного логічного синтезу на основі бібліотеки Integrated Qubit (IQ). Причому моделювання затвору IQ, на відміну від затвору Control-V або затвору Тоффолі, дозволяє створити більш надійну модель, яка точніше відображає теоретичну оборотну обчислювальну структуру. А алгоритм на основі бібліотеки IQ працює за O(N) часу та знижує квантову вартість синтезованої схеми до 45%. Алгоритм паралельного адіабатичного синтезу для адіабатичної логіки з двома шлюзами, покращує вартість схеми на 36,85% порівняно з попередніми тестами. Також була застосована техніка прямого зміщення тіла, [8], до звичайного інвертора, щоб

Продовження Додатку А

Proceedings of the 10th International Scientific and Practical Conference «Science, Education, Innovation: Topical Issues and Modern Aspects» (April 18-19, 2024). Tallinn, Estonia

110
196

INFORMATION AND WEB TECHNOLOGIES

продемонструвати ефективність зміщення тіла в динамічній диференціальній логіці.

Результати. Було показано, що PMOS- і NMOS-транзистори для регулювання порогової напруги (VTH) контролюють підпорогові витоки й уникають значного розсіювання статичної потужності й оптимізують продуктивність системи. Це пояснюється тим, що порогова напруга є функцією напруги джерела в організмі, яку можна модулювати для підвищення продуктивності за допомогою прямого зміщення. Додатковою перевагою є те, що вплив ефектів короткого каналу зменшується в міру застосування зміщення, що також зменшує коливання порогової напруги.

Крім того, було показано, що зміщення корпусу в транзисторах покращує вразливість схеми CMOS проти атак DPA. Оскільки порогова напруга підтягуючих транзисторів, що використовуються для відновлення заряду, змінюється пороговою напругою, саме відновлення погіршується, збільшуючи різницю між піковим і середнім енергоспоживанням і робить схему більш уразливою до атак аналізу потужності. Реалізуючи корпусне зміщення в транзисторах PMOS, динамічне енергоспоживання було зменшено в середньому на 50%, а також зменшено залежність даних від енергоспоживання.

Висновок. Оборотна логіка є багатообіцяючою парадигмою обчислювального дизайну, яка представляє метод побудови комп'ютерів, які виробляють довільно низьке розсіювання тепла. Основний принцип оборотних обчислень полягає в тому, що біективний пристрій з однаковою кількістю вхідних і вихідних ліній створює обчислювальне середовище, де електродинаміка системи дозволяє передбачати всі майбутні стани на основі відомих минулих станів, і система досягає всіх можливих станів, що призводить до відсутності розсіювання тепла. Оборотна логіка має важливе значення в майбутніх реалізаціях CMOS [149], квантових обчислень, оптичних обчислень і ДНК-обчислень, оскільки ці структури потрібні для подолання бар'єру kTln(2) для розсіювання енергії.

Отже, в дослідженні розглянуто дизайн та проведений аналіз із використанням високоефективної адіабатичної динамічної логіки (PADDL) для пом'якшення атак DPA.

References:

[1] T. Toffoli, "Reversible Computing," Technical Report MIT/LCS/TM-151, 1980.
[2] R. Feynman, "Simulating Physics with Computers," International

Закінчення Додатку А

Proceedings of the 10th International Scientific and Practical Conference «Science, Education, Innovation: Topical Issues and Modern Aspects» (April 16-19, 2024). Tallinn, Estonia

110
196

INTERCONF

INFORMATION AND WEB TECHNOLOGIES

Journal of Theoretical Physics, 1982.

[3] M. Born and V. A. Fock (1928). "Beweis des Adiabatensatzes". Zeitschrift für Physik A 51 (3-4): 165-180.

[4] T. Kato, "On the Adiabatic Theorem of Quantum Mechanics". Journal of the Physical Society of Japan 5 (6): 435-439, 1950.

[5] C. Clavier, J.-S. Coron, and N. Dabbous, "Differential power analysis in the presence of hardware countermeasures," in CHES '00. London, UK, UK: Springer-Verlag, 2000, pp. 252-263.

[6] J. Daemen and V. Rijmen, "Resistance Against Implementation Attacks: A Comparative Study of the AES Proposals", in Proc. of the Second Advanced Encryption Standard (AES) Candidate Conf. March 1999.

[7] S. Chari, C.S. Jutla, J.R. Rao, and P. Rohatgi, "Towards Sound Approaches to Counteract Power-Analysis Attacks", in Proc. of CRYPTO '99, Lecture Notes in Computer Science, vol. 1666, 1999, pp. 398-412.

[8] Tschanz, J.W.; Kao, J.T.; Narendra, S.G.; Nair, R.; Antoniadis, D.A.; Chandrakasan, A.P.; De, V.; "Adaptive body bias for reducing impacts of die-to-die and within-die parameter variations on microprocessor frequency and leakage," Solid-State Circuits, IEEE Journal of , vol.37, no.11, pp. 1396- 1402, Nov 2002.

ДОДАТОК Б

СЕРТИФІКАТ УЧАСНИКА МІЖНАРОДНОЇ КОНФЕРЕНЦІЇ

InterConf
Scientific Publishing Center

The Certificate confirme 12 hours of remote work on the preparation of scientific paper. The organizing committee recommends to award a 0,4 ECTS credits for participant for being involved.

Proceedings of the International Scientific and Practical Conference are available on a website:
https://archive.interconf.center/index.php/conference-proceeding/issue/archive

OPEN ACCESS

Certificate Number
Ap-2416050

Certificate of Participation

We are honored to present this certificate to

Dmytro Kucheruk

for participation in the
X International Scientific and Practical Conference
SCIENCE, EDUCATION, INNOVATION: TOPICAL ISSUES AND MODERN ASPECTS
held on April 16-18, 2024 in Tallinn, Estonia.

and for publishing a scientific paper
МЕТОД ЗАБЕЗПЕЧЕННЯ
БЕЗПЕКИ ПРОГРАМ ЗГІДНО ОБОРОТНОЇ ЛОГІКИ

SCIENTIFIC PUBLISHING CENTER
INTERCONF

ДОДАТОК В

ПРЕЗЕНТАЦІЯ ДО МАГІСТЕРСЬКОЇ РОБОТИ

Продовження Додатку В

- **Метою роботи** є реалізація адіабатичної динамічної диференціальної логіки для додатків у безпечному дизайні для пом'якшення атак ДАП.
- **Об'єктом дослідження** є процес забезпечення безпеки програм згідно оборотної логіки.
- **Предметом дослідження** є метод забезпечення безпеки програм згідно адіабатичної динамічної диференціальної логіки для додатків у безпечному дизайні для пом'якшення атак ДАП.
- **Методи дослідження**. Для розв'язання поставлених задач використовувалися методи:
  - аналіз математичних моделей із оборотної логіки;
  - аналіз допустимості послідовної логіки в оборотних обчислювальних системах;
  - моделювання фундаментальних оборотних логічних структур в адіабатичній логіці;
  - реалізація та експериментальні дослідження удосконаленої адіабатичної схеми з двома шинами для пом'якшення ДАП.

Наукова новизна одержаних результатів:

- Запропоновано математичний доказ того, що послідовні оборотні логічні структури фізично можливі. Розроблено набір обчислювальних логічних примітивів, щоб забезпечити підтримку фізичної оборотності для всіх оборотних логічних структур із шляхами зворотного зв'язку;
- Удосконалено методологію розробки адіабатичної динамічної диференціальної логіки для пом'якшення атак ДАП на захищені інтегровані чіпи. Представлено адіабатичну схему із подвійною шиною для реалізації в шифрі Rijndael для алгоритму AES для додатків із низьким енергоспоживанням і низькою частотою, таких як смарт-карти на 13,56 МГц.

Продовження Додатку В

Практична цінність

- Полягає в розробленні методу забезпечення безпеки програм згідно адіабатичної динамічної диференціальної логіки для додатків у безпечному дизайні для пом'якшення атак ДАП. Він полягає в тому, щоб включити логіку на основі безпеки в саму апаратну реалізацію, щоб ускладнити для зловмисника визначення необхідної інформації для визначення вхідних даних.

Постановка задачі дослідження

- Провести аналіз відомих математичних моделей із оборотної логіки, розглянути важливі парадигми оборотної логіки – логічну та фізичну оборотність, розглянути допустимість послідовної логіки в оборотних обчислювальних системах.
- Розробити набір обчислювальних логічних примітивів, щоб забезпечити підтримку фізичної оборотності для всіх оборотних логічних структур із шляхами зворотного зв'язку.
- Розробити найефективніший підхід до запобігання атакам ДАП, щоб ускладнити для зловмисника визначення необхідної інформації для визначення вхідних даних.
- Запропонувати реалізацію оборотної логіки в CMOS та SRAM, розробити алгоритм синтезу двошинної адіабатичної логіки.
- Представити удосконалену адіабатичну схему з двома шинами для пом'якшення атак ДАП та провести експериментальні дослідження для перевірки ефективності запропонованого вдосконалення методу забезпечення безпеки програм згідно оборотної логіки.

Продовження Додатку В

Основи оборотної та адіабатичної логіки

- **Оборотна система** — це унікальна система, у якій кожна частинка досягає всіх можливих станів. Оскільки ці перетворення компенсують одне одного, зміна ентропії дорівнює нулю, тобто тепло не розсіюється в системі.
- Ентропія пов'язана з кількістю частинок у системі:

$$\int \frac{dQ}{T} = k\ In\ (W_f) - k\ In\ (W_0),$$

- де W_f - кількість вихідних станів, W_0 - кількість вхідних станів.
- Сценарій, коли виникає мінімальна кількість розсіюваного тепла в необоротній системі, при ідентичності всіх вхідних та вихідних ймовірностей та підстановка цих значень у рівняння ентропії дає бар'єр Ландауера $k_B T In(2)$.

Основи оборотної та адіабатичної логіки

- **Адіабатична логіка** — це реалізація оборотної логіки в CMOS, де струм, що протікає через схему, контролюється таким чином, що розсіювання енергії через перемикання та розсіювання конденсатора мінімізується. Це досягається шляхом переробки енергії контуру, а не розсіювання її в навколишнє середовище.
- Зменшення розсіювання енергії досягається за рахунок використання функції наростання замість швидшого перемикання, досягнутого в ступінчастих функціях. Тому, транзистори можуть використовуватися в адіабатичній роботі, незважаючи на те, що вони продемонстровані як пристрої з втратами, і це досягається шляхом застосування двох правил:
- 1) транзистор завжди включений, коли через нього протікає значний струм;
- 2) коли є значна різниця між джерелом і напругою стоку, транзистор повинен бути вимкнений.
- Адіабатичні схеми забезпечують зниження розсіювання енергії на 60% при 20 МГц і на 35% менше енергії при 100 МГц.

Продовження Додатку В

Основи адіабатичного проектування двох рейок.

- Для реалізації адіабатичної логіки в CMOS необхідно дотримуватися двох правил:
- 1) транзистор ніколи не можна вмикати, коли на ньому є напруга;
- 2) відмінна від нуля напруга ніколи не повинна прикладатися до транзистора під час будь-якого переходу.
- Для досягнення цих проектних цілей CMOS-транзистори використовуються як перемикачі, і вони вмикаються лише тоді, коли ми хочемо, щоб напруга джерела та стоку були однаковими. Крім того, вхідними сигналами необхідно керувати за допомогою динамічного перемикання замість звичайних прямокутних форм сигналу, і синхронізувати їх так, щоб два вхідні сигнали не перемикалися одночасно.

Пропозиція щодо покращення пристрою вихідної пам'яті CMOS

- Для моделювання представимо комбінаційну реверсивну логічну структуру, яка використовує мережу PMOS/NMOS, керовану системним входом +Vo, -Vo та oV. В ній замість мережі RC використовується мережа CMOS.
- Схема працює під пороговою напругою та виконує два оборотні обчислення: «Копіювати S до M» і «Стирати за допомогою копії». Ці операції порівнювалися з незворотною операцією «Стерти без копіювання». Енергія, яку утримує біт (E_{bit}), визначається зарядом вихідного конденсатора та визначається за допомогою рівняння

$$E_{bit} = \frac{1}{2} * C_I V_0^2$$

Продовження Додатку В

Пропозиція щодо покращення пристрою вихідної пам'яті CMOS

- Розсіювання енергії для схеми PMOS і NMOS.

Операція	Напруга 1.4мВ Час перемикання 60мс	Напруга 1мВ Час перемикання 250мс
Копія «1»	2.8116E-21	1.8688E-21
Утримувати «1»	1.7954E-21	1.3235E-21
Видалити «1»	1.7476E-21	8.6546E-22
Копіювати «0»	2.8102E-21	1.8744E-21
Утримувати «0»	1.4948E-21	1.5047E-21
Видалити «0»	4.9396E-22	6.2761E-22

Пропозиція щодо покращення структури SRAM

- Рис. 1. Логічна конфігурація декодера RD 2-to-4.

Продовження Додатку В

Проектування локально оборотних логічних структур

- Метою оборотного проектування є:
  - мінімізація квантової вартості – кількості оборотних обчислень 1*1 і 2*2, необхідних для створення логічного виводу;
  - мінімізація затримки – логічної глибини пристрою;
  - зменшення допоміжних входів і "сміттєвих" виходів – входів і виходів, які не реалізовані в конструкції шлюзу і служать лише для підтримки оборотності пристрою.

Проектування локально оборотних логічних структур

- Кубіт (Qubit) – це бібліотека, що є надійною багатозначною логічною системою для визначення заданих детермінованих квантових станів і узагальнення недетермінованих станів.
- Представлення станів у двовимірному гільбертовому просторі знайдено за допомогою комплексної проективної лінії, яка є геометричною сферою, відомою як сфера Блоха (рис.2)

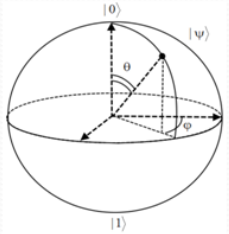

Рис. 2. Графічне представлення кубіта сфери Блоха

Продовження Додатку В

Проектування локально оборотних логічних структур

- Покажемо, що буфер 1x1, реалізований у інтегрованій бібліотеці Кубіт, має вартість і затримку рівну одиниці.
- Розглянемо реалізацію шлюзу NOT в Кубіт. Ідентична реалізація може бути досягнута шляхом розміщення двох вентилів V або V+ послідовно. Вони задаються операторами:

$$R_y(\theta) = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & \sin\left(\frac{\theta}{2}\right) \\ -\sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{pmatrix} R_z(\varphi) = \begin{pmatrix} e^{i\varphi} & 0 \\ 0 & e^{-i\varphi} \end{pmatrix}$$

- Два послідовних вентиля V+ забезпечують наступну унітарну операцію:

$$\frac{1}{i+1}\begin{pmatrix} 1 & i \\ i & 1 \end{pmatrix} * \frac{1}{i+1}\begin{pmatrix} 1 & i \\ i & 1 \end{pmatrix} = \frac{1}{2i}\begin{pmatrix} 1+i^2 & i+i \\ i+i & 1+i^2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

Проектування локально оборотних логічних структур

- Це операція NOT. Це означає, що робота V і V або V+ і V+ разом послідовно становить вартість рівну одиниці. Це пояснюється тим, що ці операції представляють повне обертання, досягаючи стану «0» або «1». Коли V і V+ розташовані послідовно, вони дають таку унітарну операцію:

$$\frac{1+i}{2}\begin{pmatrix} 1 & -i \\ -i & 1 \end{pmatrix} * \frac{1}{i+1}\begin{pmatrix} 1 & i \\ i & 1 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 1-i^2 & i-i \\ i-i & 1-i^2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

- Це матриця ідентичності. Ворота буфера не представляють обертання, але все одно досягають стану «0» або «1», як і NOT. А послідовне розміщення V і V+ за вартістю еквівалентно розміщенню V і V або V і V+ послідовно, що спричиняє вартість і затримку в одиницю. Таким чином, реверсивний буфер 1x1 ІК також несе вартість і затримку в одиницю.

Продовження Додатку В

УДОСКОНАЛЕНИЙ МЕТОД ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ ПРОГРАМ ЗГІДНО ОБОРОТНОЇ ЛОГІКИ

Представимо алгоритм оптимізації:

- перемикаємо вентилі в синтезованій схемі з їх еквівалентом у бібліотеці Кубіт;
- відбувається перевірка двох шлюзів, які примикають один до одного, щоб перевірити, чи існує менша реалізація, яка може досягти того самого логічного обчислення з меншою вартістю;
- отриманий граф розбивається на вузли наступним чином: задано оборотну схему з n входами та n виходами з вартістю c, отриманий граф матиме 2n+c-1 вузлів і n+2c ребер:
  - вузли від 0 до n-1 представляють входи;
  - вузли від n до n+c-1 представляють кожен квантовий вентиль у порядку їх затримки;
  - вузли від n+c до 2n+c-1 представляють виходи пристрою;
  - ребра мають напрямок, і оскільки алгоритм працюватиме від виходів до входів, напрямок йде від виходів до входів.

УДОСКОНАЛЕНИЙ МЕТОД ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ ПРОГРАМ ЗГІДНО ОБОРОТНОЇ ЛОГІКИ

- далі, обходимо вузли, які зустрічає кожен процес, і на основі типу вузла визначаємо, чи потрібно вузол видаляти. Якщо вузол видаляється, два вхідні дроти з'єднуються з вихідними;
- після проходження контуру ми використовуємо два правила, щоб визначити, чи потрібно видалити ребро чи вузол:
- 1) якщо жодне вхідне ребро не позначено, тоді відповідний вентиль не потрібен;
- 2) якщо вузол має один позначений вхідний рядок, який є лінією керування, і відповідний рядок також є лінією керування, тоді шлюз не потрібний, і потрібно встановити значення «Наступний шлюз» попереднього вузла на значення шлюзу наступного вузла;
- наступним кроком ми оптимізуємо схему, виконавши пошук з потрібних виходів.

Продовження Додатку В

УДОСКОНАЛЕНИЙ МЕТОД ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ ПРОГРАМ ЗГІДНО ОБОРОТНОЇ ЛОГІКИ

- Новизна удосконаленої схеми полягає в тому, що вона вперше включає методологію подвійної шини для маскування джерела живлення, адіабатичну логіку для зниження середньої потужності та транзисторну логіку для зменшення площі.
- Ця методологія дозволяє створювати однозначну, ефективну схему, яка також є фізично оборотною, дозволяючи повторне використання дизайну як для шифрування, так і для дешифрування з мінімальними накладними витратами, зменшуючи компроміс із площею, який зазвичай виникає при використанні адіабатичної логіки подвійної шини.
- Перевагою використання адіабатичного дизайну з двома шинами в цьому підході полягає в тому, що фізична оборотність дозволяє використовувати ту саму схему як для шифрування, так і для дешифрування.

Експериментальні дослідження удосконаленої схеми

Еталон	Один./ Подв.	Техн., нм	Входи джерела живл. (В)	Частота (МГц)	Кіль-кість шлюзів	Всьо-го	Площа нм2	Електри-чна сила (Ватт)
[343] (Відкрите ядро)	Один.	0.35	3.3	333	3092	6272	44593	14000
[343] (comp)	Один.	0.35	3.3	333	2242	4606	101364	32100
[343] (Comp)	Подв.	0.35	3.3	333	4442	9072	68603	8906
Запропоновано	Подв.	*0.35*	*3.3*	*50*	*512*	*10124*	*25663*	*1037*
Запропоновано	Подв.	*0.35*	*3.3*	*13.56*	*512*	*10124*	*25663*	*446.2*
Запропоновано	Подв.	*0.35*	*3.3*	*1*	*512*	*10124*	*25663*	*6.59*
[344] (Без маскування)	Подв.	0.25	2.5	13.56	n/a	n/a	n/a	165
[344] (Маскування)	Подв.	0.25	2.5	13.56	n/a	n/a	n/a	210
[353]	Одини.	0.25	2.5	10	2976	7856	8560	122
[354]	Один.	0.25	2.5	10	3316	8880	8150	416
Запропоновано	Подв.	*0.25*	*2.5*	*13.56*	*512*	*10124*	*6874.2*	*143.7*
Запропоновано	Подв.	*0.25*	*2.5*	*10*	*512*	*10124*	*6874.2*	*106.2*

Продовження Додатку В

Експериментальні дослідження удосконаленої схеми

Еталон	Одинич-ний/ Подвій-ний	Техн., нм	Входи джерела живлення (В)	Частота (МГц)	Кіль-кість шлюзів	Всьо-го	Площа нм2	Електри-чна сила (Ватт)
[349] (LUT)	Один.	0.18	1.8	0.1	1361	2566	n/a	1.85
[348] (уніфікований)	Один.	0.18	1.8	0.1	1181	2468	n/a	1.85
[348] (Bitslice)	Один.	0.18	1.8	0.1	1192	2369	n/a	1.85
Запропоновано	Подв.	*0.18*	*1.8*	*0.1*	*512*	*10124*	*3474*	*0.342*
[356] (CSSAL)	Подв.	0.18	1.8	1.25	8063	16178	n/a	4.187
[356] (ЛЕВЗ)	Один.	0.18	1.8	1.25	3270	6436	n/a	6.375
Запропоновано	Подв.	*0.18*	*1.8*	*1.25*	*512*	*10124*	*3474*	*0.69*
[356] (CSSAL)	Один.	0.18	1.8	12.5	8063	16178	n/a	83.375
[356] (ЛЕВЗ)	Один.	0.18	1.8	12.5	3270	6436	n/a	172.75
Запропоновано	Подв.	*0.18*	*1.8*	*12.5*	*512*	*10124*	*3474*	*26.97*

Висновки

- В роботі за результатами теоретичних та практичних досліджень реалізовано метод забезпечення безпеки програм згідно адіабатичної динамічної диференціальної логіки для додатків у безпечному дизайні для пом'якшення атак ДАП.
- При цьому отримані такі результати:
- Проведено аналіз відомих математичних моделей із оборотної логіки, розглянуто важливі парадигми оборотної логіки – логічну та фізичну оборотність, розглянуто допустимість послідовної логіки в оборотних обчислювальних системах.
- Удосконалено набір обчислювальних логічних примітивів, що забезпечують підтримку фізичної оборотності для всіх оборотних логічних структур із шляхами зворотного зв'язку.

Закінчення Додатку В

Висновки

- Розроблено ефективний підхід для запобігання атак ДАП, щоб ускладнити для зловмисника визначення необхідної інформації для визначення вхідних даних.
- Запропоновано реалізацію оборотної логіки в CMOS та SRAM.
- Удосконалено адіабатичну схему з двома шинами для пом'якшення атак ДАП та проведено експериментальні дослідження для перевірки ефективності запропонованого методу забезпечення безпеки програм згідно оборотної логіки.
- За темою магістерської роботи опубліковано статтю на тему «Метод забезпечення безпеки програм згідно оборотної логіки» в матеріалах конференції X International Scientific and Practical Conference Science, Education, Innovation Topical Issues and Modern Aspects, 2024, Tallin, Estonia, що індексуються в наукометричній базі Google Scholar та отриманий сертифікат участі у конференції з кількісттю годин дистанційної роботи – 12 годин (0,4 ECTS credits).

Доповідь завершена.
Дякую за увагу!

UNICHECK
by Turnitin

Ім'я користувача:
Кафедра КІ

ID перевірки:
1016266546

Дата перевірки:
21.05.2024 01:43:54 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
21.05.2024 07:27:22 EEST

ID користувача:
100005591

Назва документа: **Кучерук_Метод забезпечення безпеки програм згідно оборотної логіки**

Кількість сторінок: **98** Кількість слів: **24909** Кількість символів: **179852** Розмір файлу: **343.18 KB** ID файлу: **1016056792**

7.3%
Схожість

Найбільша схожість: **5.36%** з Інтернет-джерелом (https://archive.interconf.center/index.php/conference-proceeding/art..

7.16% Джерела з Інтернету	253	Сторінка 100
1.01% Джерела з Бібліотеки	74	Сторінка 103

0.21% Цитат

Цитати	7	Сторінка 104
Посилання	1	Сторінка 104

0%
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи	165

Mon May 20 18:30:39 EEST 2024, Медзатий Дмитро Миколайович, Хмельницький національний університет, ХНУ

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 0.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилок в документах: 10%**

	Документ		Сумарний збіг по Базі Даних	
ID: 126711 Назва: МКР Метод забезпечення безпеки програм згідно оборотної логіки Додано в БД: 2024-05-20 Автора: Кучерук Д.В. Керівники: Березька К.М. Консультанти: Опоненти:	Символи	Лексеми	Символи	Лексеми
	158958	1242	891 (1%)	13 (1%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Здобувач: Кучерук Дмитро Віталійович

Тема: Метод забезпечення безпеки програм згідно оборотної логіки

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень – ; кількість сторінок записки 88

1. Короткий зміст роботи та прийнятих рішень У роботі запропоновано реалізацію адіабатичної динамічної диференціальної логіки для додатків у безпечному дизайні для пом'якшення атак диференціального аналізу потужності DPA

2. Висновок про відповідність роботи дипломному завданню Кваліфікаційна робота магістра відповідає виданому завданню

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі проведено аналіз математичних моделей із оборотної логіки, розглянуті основи оборотної та адіабатичної логіки. Досліджено відомі рішення та засоби в цій сфері. У другому розділі розглянуто допустимість послідовної логіки в оборотних обчислювальних системах. Запропоноване покращення CMOS та структури SRAM. У третьому розділі запропоновано метод оптимізації синтезованих оборотних логічних структур. У четвертому розділі запропоновано методологію розробки адіабатичної динамічної диференціальної логіки для пом'якшення атак DPA на захищені інтегровані чіпи. Представлений адіабатичний S-блок із подвійною шиною.

4. Позитивні сторони роботи: Запропоновано метод забезпечення безпеки програм згідно адіабатичної динамічної диференціальної логіки для додатків у безпечному дизайні для пом'якшення атак DPA.

5. Негативні сторони роботи: В роботі присутні певні логічні помилки щодо опису моделей із оборотної та адіабатичної логіки, а також деякі синтаксичні помилки.

6. Оцінка графічного оформлення та пояснювальної записки роботи: –

7. Відгук про роботу в цілому: В загальному робота виконана на достатньому рівні.

8. Інші зауваження: –

9. Оцінка кваліфікаційної роботи:

Розглянувши позитивні та негативні сторони представленої кваліфікаційної роботи вважаю, що робота заслуговує оцінки «добре» 4.00 (С)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи)

Бурмак Володимир Олександрович, доктор технічних наук, професор, зав. каф. комп'ютерних наук ХНУ

" 13 травня " _____ 2024р.

Завідувачу кафедри КІС
д-р.техн.наук, проф. Говорущенко Т. О.

Кучерука Дмитра Віталійовича
ПІБ здобувача вищої освіти

ФІТ, 2 курсу, групи КІ2м-22-2

ЗАЯВА

З правилами чинного Положення «Про дотримання академічної доброчесності в Хмельницькому національному університеті» від 26.09.2020 (зі змінами від 26.11.2020), згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений. Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

13.05.2024
дата

підпис

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ

КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОМАЦІЙНИХ СИСТЕМ

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Метод забезпечення безпеки програм згідно оборотної логіки

Автор: Кучерук Дмитро Віталійович

Спеціальність: 123 – Комп'ютерна інженерія

Освітня програма: освітньо-наукова

Науковий керівник: Березька К. М., к.т.н, доцент

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
2) усі запозичення фрагментарні, або мають належним чином оформлені посилання;
3) окремі виявлені фрагменти схожості є загальновживаними фразами або виразами;
4) в якості запозичень в окремих місцях системою зафіксовано послідовності чотирьохрозрядних двійкових кодів, які є вхідними даними до великої кількості задач і не можуть розглядатися як об'єкт авторських прав і, відповідно, їх порушення;
5) всі зафіксовані системою ознаки модифікації тексту (165 символів) відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 7.3% і адресується до 327 першоджерела, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи _____ К. М. Березька

Гарант ОП _____ О. С. Савенко

Завідувач кафедри КІІСч _____ Т. О. Говорушенко