

УДК 004.312.2

Суханов Д.О., магістрант,

Безнос Д.В., магістрант

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

СТРУКТУРНО-ФУНКЦІОНАЛЬНА ОРГАНІЗАЦІЯ НАВЧАЛЬНОЇ СИСТЕМИ НАСКРІЗНОГО ПРОЄКТУВАННЯ ЦИФРОВИХ АВТОМАТІВ

Фундаментальним підґрунтям для розробки сучасних обчислювальних засобів, від найпростіших мікроконтролерів до високопродуктивних багатоядерних процесорів, є теорія цифрових автоматів [1]. У контексті підготовки майбутніх фахівців з комп'ютерної інженерії, електроніки та електронних комунікацій критично важливим є глибоке розуміння студентами архітектури дискретних систем. Досягнення такого розуміння неможливе без використання спеціалізованих систем автоматизованого проєктування (САПР), які дозволяють наочно досліджувати всі етапи створення пристрою: від початкової формалізації поведінки до генерації синтезопридатного коду апаратного рівня.

Детальний аналіз існуючих рішень у цій галузі показує суттєвий методичний розрив між промисловими та навчальними інструментами. З одного боку, професійні комплекси, такі як Intel Quartus Prime або AMD Xilinx Vivado, забезпечують вичерпний цикл проєктування. Проте вони мають надзвичайно високий поріг входження для початківців і реалізують процедури синтезу за принципом «чорного ящика». Інструменти такого класу приховують від користувача проміжні математичні перетворення, відразу видаючи оптимізовану RTL-схему (Register-Transfer Level), що унеможливує вивчення самого математичного апарату синтезу [2].

З іншого боку, популярні навчальні середовища (наприклад, Logisim чи LogiCly) орієнтовані переважно на структурно-схемотехнічний рівень. Вони забезпечують візуальне конструювання пристроїв з готових логічних вентилів та тригерів, проте абсолютно не підтримують високорівневий абстрактний синтез із регулярних виразів та не мають вбудованих механізмів для пояснення процесів мінімізації чи кодування станів [3]. У таких програмах студент змушений виконувати громіздкі матричні перетворення вручну, що часто призводить до помилок ще до етапу симуляції.

Очевидним недоліком існуючих рішень є відсутність єдиного легкого кросплатформного інструменту, який би інтерактивно

проводив студента через усі без винятку етапи математичного синтезу. З огляду на це, метою даної роботи є розробка структурно-функціональної організації веб-орієнтованої САПР та комплексу алгоритмів, які реалізують концепцію прозорого наскрізного проектування дискретних керуючих автоматів.

Запропоноване рішення являє собою кросплатформний клієнт-серверний веб-додаток. Вибір веб-орієнтованої архітектури зумовлений необхідністю забезпечення доступу до САПР з будь-якого пристрою без потреби встановлення спеціалізованого програмного забезпечення, що є критично важливим для організації дистанційного та змішаного навчання. Структурно-функціональна організація розробленої системи складається з чотирьох інтегрованих модулів:

1. Модуль формалізації вхідних даних. Цей компонент забезпечує зручну взаємодію з користувачем через два альтернативні режими. Перший режим це текстовий парсер регулярних виразів (мовою алгебри подій), який здійснює лексичний та синтаксичний аналіз формул. Другий режим це інтерактивний редактор графів, що дозволяє проектувальнику візуально створювати вершини (стани) та дуги (переходи) за допомогою графічного інтерфейсу (Drag-and-Drop).

2. Ядро математичного синтезу. Центральний обчислювальний блок системи, розгорнутий на стороні сервера. Він забезпечує об'єднання алгоритмів абстрактного, структурного та комбінаційного синтезу, виконуючи складні матричні, теоретико-множинні та алгебраїчні перетворення, необхідні для оптимізації дискретної системи.

3. Модуль візуалізації. Відповідає за реалізацію концепції «прозорого ящика». Модуль здійснює динамічну генерацію графів переходів на кожному проміжному етапі синтезу, виконує кольорове маркування та вимальовування карт Карно-Вейча для демонстрації процесу склеювання імплікант, а також буде підсумкові логічні схеми безпосередньо у браузері користувача за допомогою технологій HTML5 Canvas/SVG.

4. Модуль експорту результатів. Забезпечує інтеграцію навчальної САПР із зовнішніми професійними маршрутами проектування. Цей блок виконує трансляцію синтезованих ядром структур даних у мову опису апаратури Verilog HDL, що дозволяє перенести згенерований дизайн на реальні ПЛІС/FPGA платформи.

Програмний конвеєр ядра математичного синтезу базується на такій послідовності алгоритмічних кроків:

Абстрактний синтез. Виконує перехід від поведінкового опису пристрою до строгої математичної моделі автомата з пам'яттю. На базі

алгоритму Глушкова виконується синтаксичний аналіз регулярного виразу, розстановка основних та передосновних місць, обчислення матриці підлеглості та формування базової множини станів [4]. Відмінною рисою розробленого алгоритму є підтримка багатоваріантності цільової моделі: система здатна автоматично генерувати відмічені таблиці для автомата Мура (використовуючи канонічний спосіб відміток), розщеплювати дані на окремі таблиці переходів та виходів для класичного автомата Мілі, а також реалізовувати вироджену функцію виходів для автомата Медведєва. Одразу після цього ядро виконує алгоритмічну мінімізацію абстрактного автомата. З метою оптимізації апаратних витрат реалізовано ітеративний алгоритм Ауфенкампа-Хона, який шляхом розщеплення класів еквівалентності ефективно виявляє та видаляє дублюючі стани, суттєво зменшуючи розмірність матриці переходів.

Структурний синтез. Відповідає за апаратне відображення оптимізованої математичної моделі на фізичні лінії зв'язку. Архітектурною особливістю цього блоку є принципова відмова від жорсткої прив'язки виключно до класичної двійкової (булевої) логіки. Система дозволяє користувачеві самостійно визначати базис структурного алфавіту (m -значна логіка). Це відкриває шлях для дослідження перспективних недвійкових обчислювальних середовищ (наприклад, трійкової логіки). Після математичного розрахунку необхідної розрядності ліній та виконання процедури кодування станів (State Assignment), алгоритм звертається до внутрішньої бібліотеки елементарних автоматів. Система підключає характеристичні рівняння обраних елементів пам'яті (синхронні D, T, RS, JK-тригери або їх спеціалізовані багатозначні аналоги) та автоматично формує розширену структурну (кодовану) таблицю переходів.

Комбінаційний синтез. Виконує фінальний розрахунок кінцевої логічної схеми, що реалізує функції керування та пам'яті. На основі структурної таблиці алгоритм генерує канонічні системи рівнянь у досконалій диз'юнктивній нормальній формі (ДДНФ) для функцій збудження кожного тригера та функцій виходів апарата. Для забезпечення високої освітньої цінності, етап логічної мінімізації є варіативним та адаптивним. Для задач малої розмірності (до 5-6 змінних) система автоматично генерує візуально-евристичні карти Карно-Вейча, дозволяючи студентів побачити процес оптимізації. Тоді як для складних систем з великою кількістю станів застосовуються строгі таблично-аналітичні алгоритми (метод Куайна-Мак-Класкі) та сумісна оптимізація систем булевих функцій, що забезпечує пошук

спільних імплікант. Завершується конвеєр синтезу технологічним відображенням (Technology Mapping) знайдених мінімізованих рівнянь (МДНФ) у заданий цільовий логічний базис реалізації (наприклад, суто І-НЕ або АБО-НЕ) [5].

Створення веб-додатка на основі запропонованої структурно-функціональної організації та комплексу алгоритмів дасть змогу не просто автоматизувати повний цикл проектування цифрових пристроїв. Головною перевагою рішення є реалізація педагогічної концепції «прозорого ящика» (glass-box design). Розроблена система інтерактивно візуалізує кожен проміжний етап перетворення абстрактної математики у кремнієву структуру. Вона дозволяє студентів детально відстежити увесь ланцюг трансляції: від того, як конкретний символ регулярного виразу породжує абстрактний стан, далі кодується у двійковий вектор і, зрештою, матеріалізується у фізичний логічний вентилювальний тригер.

Крім того, запропонована організація системи повноцінно підтримує гнучке дослідження багатозначної логіки та застосування різних алгоритмів мінімізації. Це виводить її за рамки простого навчального тренажера і робить повноцінним науково-дослідним інструментом, який є легкодоступним безпосередньо у веб-браузері без необхідності розгортання та ліцензування «важкого» індустріального програмного забезпечення.

Список використаних джерел

1. Баркалов О. О., Бабаков Р. М. Прикладна теорія цифрових автоматів в сучасній українській науці. Матеріали наукових конференцій.- 2021. - С. 318–320.
2. Тарасов О. В. Системи автоматизованого проектування мікроелектронної апаратури: навчальний посібник. Харків: ХНУРЕ, 2019. - 215с.
3. Burch C. Logisim: a graphical system for logic circuit design and simulation. Journal of Educational Resources in Computing (JERIC). 2002. - Vol. 2, No. 1. - P. 5–16.
4. Нікольський Ю. В., Пасічник В. А., Щербина Ю. М. Дискретна математика: підручник. Львів : Магнолія плюс, 2006. - 608 с.
5. Редько І. В., Лисенко О. М. Функціонально-логічне проектування. Київ : КПІ ім. Ігоря Сікорського, 2023. - 152 с.