

УДК 681.3

## Оптимізація схеми формування мікрооперацій

### у мікропрограмних пристроях керування

Баркалов О.О., Зеленьова І.Я., Казачанський А.В., Толкачев Д.О.  
Донецький національний технічний університет  
[irina@cs.dgtu.donetsk.ua](mailto:irina@cs.dgtu.donetsk.ua)

#### Abstract

*Barkalov O., Zelenyova I., Kazachansky A. Optimization of the microoperations formation scheme of microprogram control units. Use of a composition of control algorithms flowgraph vertical fragmentation method and compatible microoperations fields coding method for microprogram control units operational part optimization is considered in article. Ways of algorithms flowgraph vertical fragmentation are considered. Analysis of the offered methods composition for microprogram control units operational part optimization is carried out.*

#### Вступ

У наш час цифрові системи використовуються майже в усіх галузях промисловості для контролю та керування технологічними процесами. Такі цифрові системи доцільно реалізовувати або на мікроконтролерах, або на програмованих логічних інтегральних схемах (ПЛІС). При реалізації електронних цифрових систем на ПЛІС у вигляді «систем на кристали» необхідно синтезувати та реалізовувати пристрій керування процесом функціонування таких систем.

Одними з найбільш поширених на сучасному ринку є ПЛІС типу FPGA (Field Programmable Gate Array). Ці ПЛІС мають найбільшу кількість логічних комірок серед усіх типів програмованих логічних інтегральних схем та можуть бути багаторазово перепрограмовані.

Наразі існує велика кількість структур пристроїв керування та методів синтезу та оптимізації цих структур у базисі програмованих логічних матриць (ПЛМ) [1]. Але цей базис вже є морально застарілим та майже ніде не використовується. Методи та алгоритми синтезу та оптимізації структур пристроїв керування у базисі ПЛМ не можна без змін використовувати для синтезу пристроїв керування у базисі FPGA. Тому існує необхідність розробки структур пристроїв керування і методів їх синтезу та оптимізації у базисі сучасних мікросхем FPGA.

Пристрої керування найчастіше реалізуються разом із об'єктом керування на одній мікросхемі FPGA [1]. Тому існує необхідність розробки такої структури пристрою керування, яка б була оптимальною щодо співвідношення апаратних витрат та швидкодії і при цьому оптимально використовувала

можливості базису FPGA. Для реалізації у цьому базисі найбільш доцільно використовувати структури керуючих автоматів, що реалізовані у вигляді мікропрограмних пристроїв керування, які мають регулярну структуру та які легко реалізуються у базисі сучасних FPGA з одночасною мінімізацією апаратних витрат.

Головна ідея запропонованого у статті підходу до оптимізації операційної частини мікропрограмних пристроїв керування полягає у застосуванні композиції методів вертикалізації граф-схеми алгоритму керування та кодування полів сумісних мікрооперацій. Застосування композиції цих двох методів скорочує апаратні витрати при реалізації операційної частини пристроїв керування без значного зменшення швидкодії і при цьому оптимально використовуються можливості базису FPGA.

#### Обґрунтування задачі

Для оптимізації операційної частини мікропрограмних пристроїв керування найчастіше використовуються три методи [2]:

- 1) метод максимального кодування наборів мікрооперацій;
- 2) метод кодування полів сумісних мікрооперацій;
- 3) метод вертикалізації граф-схеми алгоритму керування.

При використанні методу унітарного кодування мікрооперацій (МО) кожній мікрооперації ставиться у відповідність окремий розряд у форматі слова керування. При використанні цього методу досягається максимальна швидкодія функціонування мікропрограмного пристрою керування, але апаратні витрати є максимальними серед усіх чотирьох методів. Довжина формату слів

керування у керуючій пам'яті (КП) є максимальною.

При використанні методу максимального кодування наборів мікрооперацій у керуючій пам'яті формується код, що відповідає окремій мікрокоманді (МК) у мікропрограмі керування. Усі мікрокоманди, що присутні у мікропрограмі керування, кодуються за допомогою позиційного коду МК. Це максимально скорочує довжину формату слів керування, а, отже, і ємність КП. При застосуванні цього методу оптимізації операційної частини мікропрограмних пристроїв керування максимально скорочуються апаратні витрати при реалізації блоку керуючої пам'яті, але збільшуються апаратні витрати при реалізації схеми формування мікрооперацій (СФМО), а швидкодія функціонування пристрою керування є найменшою серед усіх трьох методів оптимізації.

Метод кодування полів сумісних мікрооперацій заснований на тому, що найчастіше у мікропрограмах керування у окремих станах автомата необхідно формувати далеко не усі мікрооперації, що присутні у алгоритмі керування. Тому мікрооперації, що не використовуються разом в жодній мікрокоманді у алгоритмі керування, можна об'єднати у одне поле (такі мікрооперації є сумісними) та формувати у окремому стані автомата тільки одну мікрооперацію з окремого поля. Мікрооперації у окремому полі кодуються за допомогою позиційного коду мікрооперацій. Цей метод оптимізації операційної частини мікропрограмних пристроїв керування є середнім за показниками апаратних витрат та швидкодії між методами унітарного кодування мікрооперацій та максимального кодування наборів мікрооперацій. Довжина формату слів керування у керуючій пам'яті скорочується у порівнянні із методом унітарного кодування мікрооперацій, а, отже, скорочуються і апаратні витрати при реалізації блоку КП. Окремі поля сумісних мікрооперацій реалізуються у вигляді дешифраторів полів у складі схеми формування мікрооперацій. Ці дешифратори, як правило, містять менше еквівалентних вентилів, аніж дешифратор коду мікрокоманд у методі максимального кодування наборів мікрооперацій. Це зменшує апаратні витрати при реалізації схеми формування мікрооперацій та зменшує часову затримку у блоці СФМО у порівнянні з методом максимального кодування наборів мікрооперацій.

При застосуванні методу кодування полів сумісних мікрооперацій виникає необхідність оптимального розподілення сумісних мікрооперацій по окремим полям. Розроблено ряд алгоритмів розподілення сумісних мікрооперацій по окремим полям, що мають свої

переваги та недоліки. Для того, щоб оцінити ефективність використання можливостей окремого поля, пропонується ввести коефіцієнт наповненості окремого поля:

$$K_m^F = \frac{P_m}{P_m^{MAX}}, \quad (1)$$

де:  $K_m^F$  - коефіцієнт наповненості поля  $m$ ;

$P_m$  - поточна потужність поля  $m$  (кількість мікрооперацій у цьому полі);

$P_m^{MAX}$  - максимально можлива потужність поля  $m$  (максимально можлива кількість мікрооперацій, що може бути введена до цього поля).

Коефіцієнт  $K_m^F$  дає можливість оцінити ефективність використання можливостей окремого поля  $m$  - чим більше значення цього коефіцієнту наближується до одиниці, тим ефективніше використовуються можливості відповідного поля. Для того, щоб оцінити ефективність використання можливостей усіх полів у СФМО в загалом, пропонується ввести середній коефіцієнт наповненості полів:

$$K_{SFMO}^F = \frac{\sum_{m=1}^M K_m^F}{M}, \quad (2)$$

де:  $K_{SFMO}^F$  - середній коефіцієнт наповненості полів;

$M$  - загальна кількість полів у схемі формування мікрооперацій.

На рис. 1 наведений приклад графіка зміни значення середнього коефіцієнту наповненості полів у залежності від кількості мікрооперацій, що присутні у алгоритмі керування. Для реалізації дешифраторів окремих полів при виконанні цього дослідження використовуються елементи LUT (Look-Up Table) із чотирма входами ( $I_{LUT} = 4$ ). Максимально можлива потужність окремого поля дорівнює 15 ( $P_m^{MAX} = 15$ ). При виконанні дослідження для кожного значення кількості мікрооперацій виконується розрахунок значення середнього коефіцієнту наповненості полів для трьох випадкових розподілень мікрооперацій по мікрокомандах. При цьому відкидаються крайні випадки, коли у кожній мікрокоманді міститься не більше за одну мікрооперацію, або коли у якійсь з мікрокоманд міститься більше за 80% усіх мікрооперацій, що присутні у алгоритмі керування. Після обчислення значення середнього коефіцієнту наповненості полів для трьох розподілень мікрооперацій розраховується середнє значення цього коефіцієнту:

$$K_{SFMO}^F = \frac{K_{SFMO}^{F1} + K_{SFMO}^{F2} + K_{SFMO}^{F3}}{3}. \quad (3)$$

Таким чином отримується достатньо об'єктивне значення цього коефіцієнту для окремого значення кількості мікрооперацій.

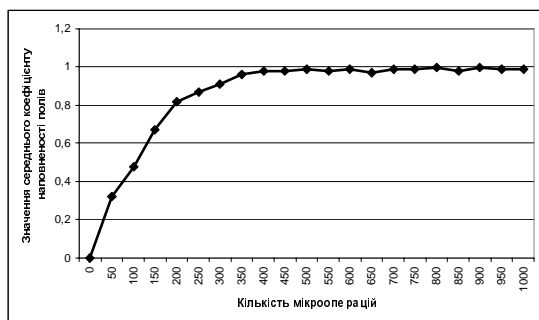


Рисунок 1 – Графік залежності значення середнього коефіцієнту наповненості полів від кількості мікрооперацій у алгоритмі керування

Аналізуючи графік (рис. 1), можна зробити висновок, що швидкість зростання лінії графіку до моменту наближення її до одиничного значення тим більша, чим більша ефективність роботи того чи іншого алгоритму розподілу сумісних мікрооперацій по окремим полям.

При використанні методу вертикалізації граф-схеми алгоритму керування, як правило, збільшується кількість станів у алгоритмі керування, що значно зменшує швидкодійність пристрою керування, який реалізує такий алгоритм. Але при вертикалізації граф-схеми алгоритму усі мікрооперації, що в ньому присутні, стають сумісними одна з одною і їх можна ввести до одного поля, закодувавши при цьому за допомогою позиційного коду мікрооперацій. При застосуванні цього методу набагато зменшується довжина формату слів керування, а, отже, і необхідна ємність керуючої пам'яті. Але при цьому необхідно реалізовувати дешифратор поля, який, найчастіше, повинен мати велику кількість входів та виходів, при цьому збільшується часова затримка на схемі формування мікрооперацій. При розбитті мікрооперацій, що присутні у мікропрограмі керування, на декілька полів зменшується часова затримка у блоці СФМО, а при реалізації дешифраторів полів на LUT-елементах у базисі FPGA зменшується і кількість елементів LUT, що необхідна для реалізації дешифраторів полів у складі схеми формування мікрооперацій. При цьому збільшується необхідний об'єм керуючої пам'яті. При застосуванні методу вертикалізації граф-схеми алгоритму керування ефективність використання можливостей окремих полів досягає максимального значення. На рис. 2 наведений графік залежності зміни значення середнього коефіцієнту наповненості полів від кількості мікрооперацій у алгоритмі керування при застосуванні методу вертикалізації граф-схеми алгоритму керування. Для дослідження застосовуються LUT-елементи з чотирма входами ( $I_{LUT} = 4$ ), максимально можлива потужність кожного поля дорівнює 15 ( $P_m^{MAX} = 15$ ).

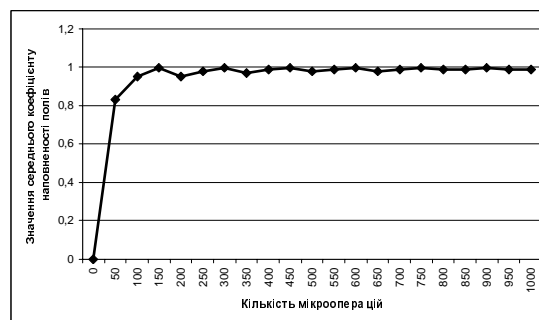


Рисунок 2 - Графік залежності значення середнього коефіцієнту наповненості полів від кількості мікрооперацій у алгоритмі керування при використанні методу вертикалізації граф-схеми алгоритму керування

Аналізуючи графік (рис. 2), можна зробити висновок, що швидкість зростання лінії до моменту досягнення одиничного значення значно вища, аніж при використанні методу кодування полів сумісних мікрооперацій. Деяке коливання значення середнього коефіцієнту наповненості полів у СФМО обумовлене тим, що одне з полів іноді не заповнюється повністю. Але із збільшенням кількості мікрооперацій ці коливання стають меншими.

### Модифікація методу вертикалізації граф-схем алгоритмів керування

При застосуванні класичного методу вертикалізації граф-схеми алгоритму керування алгоритм перетворюється таким чином, що у кожній мікрокоманді знаходиться не більше за одну мікрооперацію [3]:

$$|Y(b_q)| \leq 1, \quad (4)$$

де:  $Y(b_q)$  - мікрокоманда, що записана у вершині  $b_q$ .

Формула (4) передбачає варіант, коли у алгоритмі керування присутні порожні вершини.

Перед застосуванням методу вертикалізації граф-схеми алгоритму керування необхідно здійснювати пошук та видалення еквівалентних мікрооперацій, що можуть бути присутні у алгоритмі. Це дозволяє у деяких випадках скоротити апаратні витрати при реалізації операційної частини мікропрограмних пристроїв керування.

Метод вертикалізації граф-схеми алгоритму керування полягає у тому, що вершини алгоритму, які містять мікрокоманди, що задовольняють умові:

$$|Y(b_q)| > 1 \quad (5)$$

розбиваються на  $|Y(b_q)|$  операторних вершин. Кожна з таких вершин алгоритму містить по одній мікрооперації з мікрокоманди, яка знаходилась у вершині, що розбивалась (рис. 3).

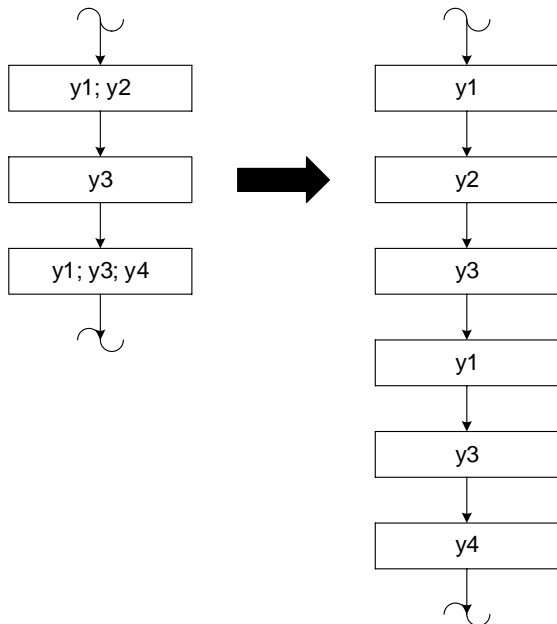


Рисунок 3 – Приклад вертикалізації граф-схеми алгоритму керування за допомогою розбиття вершин у алгоритмі

При застосуванні такого методу вертикалізації граф-схеми алгоритму керування значно збільшується кількість операторних вершин у алгоритмі, а, отже, і станів керуючого автомата. При цьому значно зменшується швидкодія пристрою керування, що реалізує такий алгоритм.

Модифікація класичного методу полягає у вертикалізації граф-схеми алгоритму керування за допомогою перекодування сигналів керування у об'єкті керування кожній мікрокоманді ставиться у відповідність окремий один сигнал керування (рис. 4).

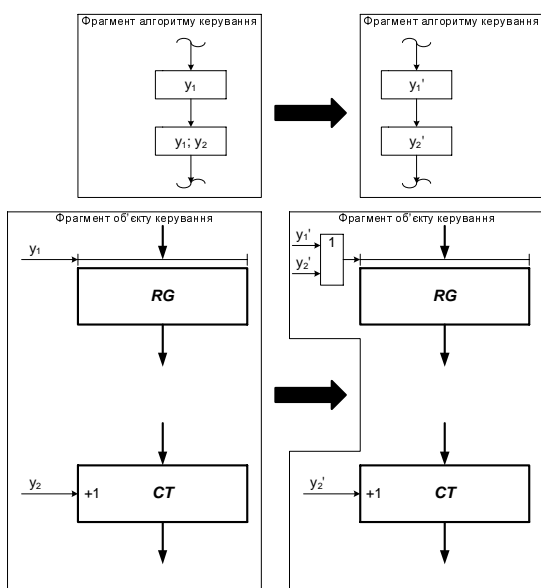


Рисунок 4 – Приклад вертикалізації граф-схеми алгоритму за допомогою методу перекодування сигналів керування у об'єкті керування

При застосуванні такого методу вертикалізації граф-схеми алгоритму керування не відбувається збільшення операторних вершин у алгоритмі. При цьому кінцеві мікрооперації формуються за допомогою об'єднання за логічною функцією АБО усіх перетворених сигналів керування, що замінюють мікрокоманди, у яких була присутня мікрооперація, яка розглядається.

При реалізації функції АБО за допомогою елементів LUT у базисі FPGA необхідна кількість таких елементів для формування окремої мікрооперації розраховується за наступною формулою:

$$N_{LUT}^{AND} = \left\lceil \frac{N_y - 1}{I_{LUT}} \right\rceil, \quad (6)$$

де:  $N_{LUT}^{AND}$  - кількість елементів LUT, що необхідна для формування окремої мікрооперації за допомогою логічної функції АБО;

$N_y$  - кількість мікрокоманд у не перетвореному алгоритмі керування (або наборі алгоритмів керування для одного об'єкту керування), у яких зустрічається мікрооперація, що формується;

$I_{LUT}$  - кількість входів у окремому LUT-елементі.

Якщо існує набір з декількох алгоритмів для одного об'єкту керування, то кожній мікрокоманді з цих алгоритмів ставиться у відповідність свій сигнал керування (рис. 5).

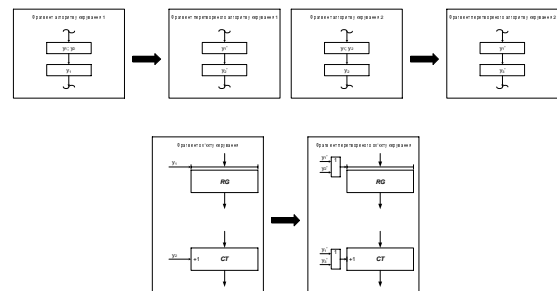


Рисунок 5 – Приклад перекодування сигналів керування у наборі алгоритмів для одного об'єкту керування

Перекодування сигналів керування у декількох алгоритмах для одного об'єкту керування дозволяє побудувати універсальну схему формування мікрооперацій для пристрою керування цим об'єктом.

Реалізація методу перекодування сигналів керування у об'єкті керування з метою вертикалізації граф-схеми алгоритму схожа на реалізацію методу максимального кодування наборів мікрооперацій для оптимізації операційної частини мікропрограмного пристрою керування. Але ж, на відміну від максимального кодування, при застосуванні

методу перекодування сигналів керування дешифратор коду мікрокоманд можна реалізовувати у вигляді набору менших дешифраторів, що зменшує апаратні витрати при реалізації цих дешифраторів на елементах LUT у базисі FPGA та збільшує швидкодію пристрою керування у порівнянні із методом максимального кодування наборів мікрооперацій.

При використанні методу перекодування сигналів керування у об'єкті керування до складу блоку схеми формування мікрооперацій крім блоку дешифрації коду мікрокоманд (DC) необхідно вводити блок безпосереднього формування мікрооперацій (БФМО), що будується на LUT-елементах, які виконують функції логічного АБО (рис. 6). Це дещо зменшує швидкодію пристрою керування.

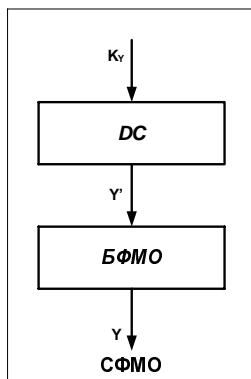


Рисунок 6 – Структура блоку СФМО при використанні методу перекодування сигналів керування у об'єкті керування для вертикалізації граф-схеми алгоритму

### Використання композиції методів вертикалізації граф-схеми алгоритму керування та кодування полів сумісних мікрооперацій

Для реалізації операційної частини мікропрограмних пристроїв керування використовуються наступні елементи базису FPGA [4]:

- 1) функціональні генератори, що реалізуються у вигляді таблиць перетворення (LUT) у складі логічних комірок міросхем FPGA;
- 2) блочна пам'ять (Block RAM).

При введенні усіх мікрооперацій до одного поля за умови використання методу вертикалізації граф-схеми алгоритму керування необхідно реалізовувати відносно великий дешифратор коду мікрооперацій на елементах LUT у базисі FPGA, що потребує великих апаратних витрат та зменшує швидкодію мікропрограмного пристрою керування. Тому пропонується розподілити мікрооперації по декількох полях.

Блок керуючої пам'яті у базисі FPGA доцільно реалізовувати на блочній пам'яті. Кількість розрядів шини даних блочної пам'яті, що необхідна для кодування мікрооперацій, у загальному випадку розраховується за наступною формулою:

$$N_{BRAM}^D = \sum_{m=1}^M \lceil \log_2(|Y_{FIELD}^m| + 1) \rceil, \quad (7)$$

де:  $N_{BRAM}^D$  - кількість розрядів шини даних блочної пам'яті, що необхідна для кодування мікрооперацій в усіх полях блоку СФМО;

$Y_{FIELD}^m$  - масив мікрооперацій  $\{y_1, \dots, y_N\}$ , що містяться у окремому полі  $m$ ;

$M$  - загальна кількість полів розбиття мікрооперацій.

Дешифратори окремих полів у базисі FPGA доцільно реалізовувати на елементах LUT. При цьому, якщо усі мікрооперації містяться у одному полі, то кількість елементів LUT, що необхідна для реалізації дешифратора цього поля без спрощення його структури, розраховується за наступною формулою:

$$N_{LUT} = |Y|^* \left\lceil \frac{\log_2(|Y| + 1) \lceil -1 \rceil}{I_{LUT} - 1} \right\rceil, \quad (8)$$

де:  $N_{LUT}$  - кількість LUT-елементів, що необхідна для реалізації дешифратора поля;

$Y$  - масив мікрооперацій  $\{y_1, \dots, y_S\}$ , що присутні у мікропрограмі керування;

$I_{LUT}$  - кількість входів у окремому елементі LUT.

При розбитті мікрооперацій по декільком полям кількість елементів LUT, що необхідна для реалізації дешифратора окремого поля  $m$  (без спрощення його структури), розраховується за наступною формулою:

$$N_{LUT} = |Y_{FIELD}^m|^* \left\lceil \frac{\log_2(|Y_{FIELD}^m| + 1) \lceil -1 \rceil}{I_{LUT} - 1} \right\rceil. \quad (9)$$

Для визначення кількості LUT-елементів, що необхідна для реалізації одного розряду дешифратора поля  $m$ , у якому міститься  $|Y_{FIELD}^m|$  мікрооперацій, вводиться поняття класу поля  $m$ .

*Визначення.* Класом поля  $m$  ( $i_m$ ) називається така кількість LUT-елементів, яка необхідна для реалізації одного розряду дешифратора поля, яке уміщує  $|Y_{FIELD}^m|$  мікрооперацій.

Клас поля  $m$  розраховується за наступною формулою:

$$i_m = \left\lceil \frac{\log_2(|Y_{FIELD}^m| + 1) \lceil -1 \rceil}{I_{LUT} - 1} \right\rceil, \quad (10)$$

де:  $i_m$  - клас окремого поля  $m$ .

Максимально можлива потужність поля  $m$  із визначеним значенням класу цього поля

$i_m$ , тобто максимально можлива кількість мікрооперацій, яку можна увести до цього поля без підвищення його класу, розраховується за наступною формулою:

$$P_m^{MAX} = 2^{I_{LUT} + (i_m - 1) * (I_{LUT} - 1)} - 1, \quad (11)$$

де:  $P_m^{MAX}$  - максимально можлива потужність поля  $m$ .

При використанні композиції методів вертикалізації граф-схеми алгоритму керування та кодування полів сумісних мікрооперацій вдається розподілити мікрооперації на оптимальну кількість полів із заданим значенням максимального класу цих полів ( $I_{MAX}$ ). При цьому ефективність використання можливостей кожного з полів досягає максимального значення. Загальна кількість полів, на яку розбивається масив мікрооперацій, у цьому випадку розраховується за допомогою наступної формули:

$$M = \left\lceil \frac{|Y|}{2^{I_{LUT} + (I_{MAX} - 1) * (I_{LUT} - 1)} - 1} \right\rceil, \quad (12)$$

де:  $M$  - необхідна кількість полів;

$Y$  - масив мікрооперацій  $\{y_1, \dots, y_S\}$ , що присутні у мікропрограмі керування;

$I_{LUT}$  - кількість входів у окремому елементі LUT;

$I_{MAX}$  - задане значення максимального класу полів.

За умови  $I_{MAX} = 1$  апаратні витрати на реалізацію дешифраторів полів досягають мінімального значення, бо за цієї умови для реалізації усіх полів знадобиться наступна кількість елементів LUT ( $N_{LUT}$ ):

$$N_{LUT} = |Y|. \quad (13)$$

А за умови  $I_{MAX} > 1$  для реалізації усіх полів знадобиться наступна кількість елементів LUT:

$$N_{LUT} = I_{MAX} * |Y|. \quad (14)$$

За умови  $I_{MAX} = 1$  також мінімізується часова затримка на схемі формування мікрооперацій ( $t_{SFMO}$ ), яка дорівнює максимальній затримці на окремому елементі LUT серед усіх цих елементів:

$$t_{SFMO} = \max(t_{LUT}^1, \dots, t_{LUT}^S), \quad (15)$$

де:  $t_{LUT}^1, \dots, t_{LUT}^S$  - часові затримки на елементах LUT;

$S$  - кількість мікрооперацій у мікропрограмі керування.

Часова затримка на окремому дешифраторі поля тим більша, чим більше каскадів LUT-елементів необхідно для реалізації окремого розряду цього дешифратора. При цьому кількість каскадів елементів LUT у дешифраторі окремого

поля  $m$  пов'язана із класом цього поля наступним чином:

$$i_m = \sum_{t=1}^T (I_{LUT})^{t-1}, \quad (16)$$

де:  $i_m$  - клас поля  $m$ ;

$T$  - кількість каскадів елементів LUT у дешифраторі поля  $m$ ;

$I_{LUT}$  - кількість входів у окремому елементі LUT.

Формула (16) показує максимально можливу кількість LUT-елементів, яку можна використовувати для реалізації окремого розряду дешифратора поля  $m$  при заданому значенні кількості каскадів ( $T$ ), що можна робити при реалізації цього дешифратора. Незавжди побачити, що формула (16) – це геометрична прогресія. З цієї формули можна легко вивести формулу розрахунку кількості каскадів, що утворюються при реалізації дешифратора поля  $m$  із заданим класом цього поля ( $i_m$ ):

$$T = \lceil \log_{I_{LUT}} (i_m * (I_{LUT} - 1) + 1) \rceil. \quad (17)$$

На рис. 7 наведений приклад реалізації дешифратора окремого поля  $m$  на елементах LUT із чотирма входами ( $I_{LUT} = 4$ ) та класом поля  $i_m = 1$ .

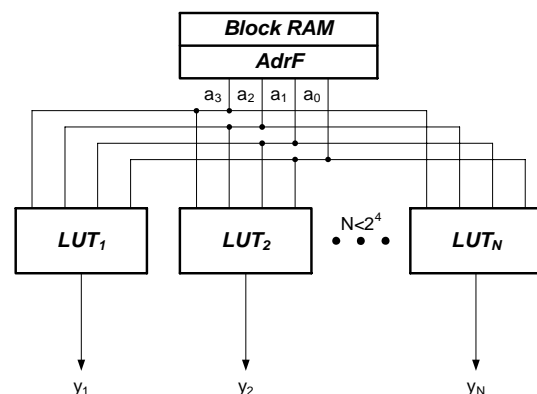


Рисунок 7 – Приклад реалізації дешифратора окремого поля у базисі FPGA

### Дослідження ефективності композиції методів вертикалізації граф-схеми алгоритму керування та кодування полів сумісних мікрооперацій

На рис. 8 наведений графік залежності кількості розрядів шини даних блоку керуючої пам'яті, що необхідні для кодування мікрооперацій, які присутні у мікропрограмі керування, від кількості мікрооперацій. Дослідження проводяться для трьох методів оптимізації операційної частини мікропрограмного пристрою керування: методу унітарного кодування мікрооперацій; методу

вертикалізації граф-схеми алгоритму керування із введенням усіх мікрооперацій до одного поля і кодування цих мікрооперацій за допомогою максимального кодування; методу вертикалізації граф-схеми алгоритму керування із розбиттям мікрооперацій по полям, що мають значення максимального класу  $I_{MAX} = 1$ .

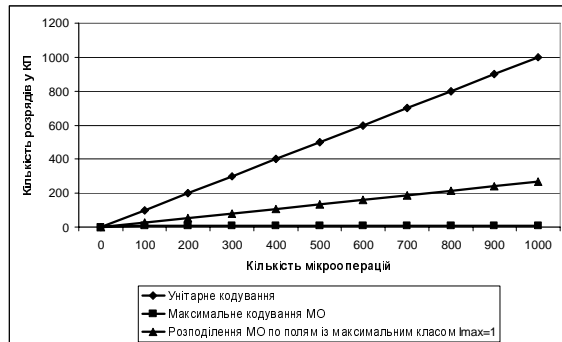


Рисунок 8 – Графік залежності кількості розрядів шини даних КП, що необхідні для кодування мікрооперацій, від кількості мікрооперацій

З цього графіку виходить, що при розподіленні мікрооперацій по декількох полях, що мають максимальний клас  $I_{MAX} = 1$  за умови використання методу вертикалізації граф-схеми алгоритму керування необхідна кількість розрядів шини даних КП має середнє значення серед трьох методів, що досліджуються.

На рис. 9 наведений графік залежності кількості елементів LUT, що необхідні для реалізації дешифраторів полів у СФМО, від кількості мікрооперацій у мікропрограмі керування. Дослідження проводяться для двох методів оптимізації операційної частини мікропрограмних пристроїв керування: методу вертикалізації граф-схеми алгоритму керування із введенням усіх мікрооперацій до одного поля і кодування цих мікрооперацій за допомогою максимального кодування; методу вертикалізації граф-схеми алгоритму керування із розбиттям мікрооперацій по полях, що мають значення максимального класу  $I_{MAX} = 1$ .

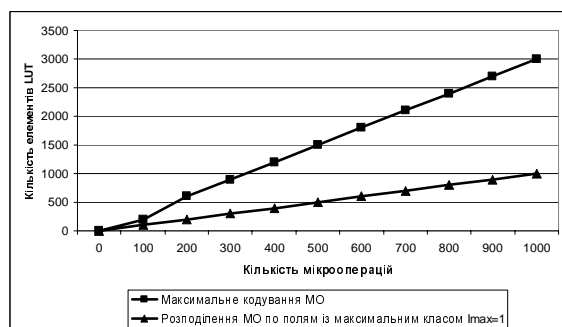


Рисунок 9 – Графік залежності кількості LUT-елементів, що необхідні для реалізації дешифраторів полів у блоці СФМО, від кількості мікрооперацій

З цього графіку виходить, що кількість елементів LUT, що необхідні для реалізації дешифраторів полів у схемі формування мікрооперацій, майже лінійно залежить від кількості мікрооперацій, що присутні у мікропрограмі керування для обох методів, що досліджуються. Кількість елементів LUT, що необхідна для реалізації полів мікрооперацій за умови розбиття цих мікрооперацій по декільком полям, які мають максимальний клас  $I_{MAX} = 1$ , значно менша, ніж для випадку, коли усі мікрооперації вводяться до одного поля.

## Висновки

Застосування композиції методів вертикалізації граф-схеми алгоритму керування та кодування полів сумісних мікрооперацій для оптимізації операційної частини мікропрограмних пристроїв керування дозволяє розподілити мікрооперації на оптимальну кількість полів та із максимальною ефективністю використовувати можливості кожного з полів мікрооперацій. При цьому швидкодія функціонування мікропрограмного пристрою керування більша за швидкодію пристрою керування при введенні усіх мікрооперацій, що присутні у мікропрограмі, до одного поля, а апаратні витрати при реалізації дешифраторів полів у складі схеми формування мікрооперацій менші, що підтверджується проведеними дослідженнями.

## Література

1. Соловьев В.В. Проектирование цифровых систем на основе программируемых логических интегральных схем / В.В. Соловьев – М.: Горячая линия-Телеком, 2001. – 636 с. ил.
2. Баркалов А.А. Синтез устройств управления на программируемых логических устройствах : учебное пособие для студентов ВУЗов, обучающихся по специальностям „Компьютерные системы и сети” и „Системное программирование” / А.А. Баркалов – Донецк : ДонНТУ, 2002. – 261 с.
3. Баркалов А.А. Синтез микропрограммных устройств управления / А.А. Баркалов, А.В. Палагин. – К., 1997. – 136 с.
4. Кузелин М.О. Современные семейства ПЛИС фирмы Xilinx. Справочное пособие / М.О. Кузелин, Д.А. Кнышев, В.Ю. Зотов – М. : Горячая линия-Телеком, 2004. – 440 с. ил.

Поступила в редколлегию 10.03.2009