

АНАЛІЗ МЕТОДІВ ЗМЕНШЕННЯ АПАРАТУРНИХ ВИТРАТ ПРИ РЕАЛІЗАЦІЇ СХЕМ СУМІЩЕНИХ МІКРОПРОГРАМНИХ АВТОМАТІВ НА CPLD

С. С. Грушко, І. Я. Зеленцова

Запорізький національний технічний університет

Анотація. *Наведений огляд моделей пристроїв керування. Докладно розглянута модель суміщеного мікропрограмного автомата. Проаналізована архітектура програмованих логічних інтегральних схем типу CPLD. Запропоновані базова та оптимізовані структури реалізації схем пристроїв керування в базисі CPLD. Отримані експериментальні дані, що дозволяють оцінити ступінь ефективності цих структур за критерієм зменшення апаратних витрат. Дослідження проведено для мікросхем CPLD фірм Xilinx та Altera.*

Ключові слова: *суміщений мікропрограмний автомат, граф-схема алгоритму, CPLD, PAL, кодування станів, псевдоеквівалентні стани.*

Вступ

Існує великий клас задач проектування цифрових систем, що можуть бути вирішені за допомогою синхронних цифрових логічних схем з пам'яттю. Прикладами корисних синхронних схем з пам'яттю є мікропроцесори, мережеві інтерфейси, що працюють за мережевим протоколом, контролери роботизованих машин і багато інших типів схем керування [1].

Синхронна цифрова логічна схема з пам'яттю складається з двох частин: логіки керування і операційної частини [1, 2]. Операційна частина складається із компонентів, що зберігають, об'єднують та іншим чином маніпулюють даними. Такі компоненти містять регістри, суматори, мультиплексори і т.д. Логіка керування складає схему, що необхідна для генерації сигналів-мікрооперацій для управління всіма компонентами в операційній частині.

Оскільки робота синхронної схеми з пам'яттю може бути описана алгоритмом, такі схеми можуть бути розроблені з алгоритмічного опису. Задача синтезу логічної схеми спочатку може бути вирішена розробкою алгоритму на мові високого рівня, який потім може бути перетворений в HDL опис, що далі синтезується для бажаної апаратної платформи [1].

Сучасна апаратна платформа представлена мікросхемами програмованої логіки FPGA і CPLD. Мікросхеми типу CPLD користуються широким попитом завдяки своїй доступності. CPLD, як і будь-які інші структури мають певні конструктивні параметри, тому алгоритми керування, що на них реалізуються можуть бути оп-

тимізовані за цими параметрами.

Критерії оптимізації можуть бути різними: зменшення енергоспоживання автоматів, підвищення швидкодії та інше [2]. Вибір пріоритетного критерію залежить від загальної концепції та кінцевої мети проекту. В даній роботі розглянуто методи оптимізації логічних схем пристроїв керування за критерієм апаратних витрат при реалізації на ПЛІС.

1. Аналіз літературних джерел

Логіка керування цифровим пристроєм може бути задана функціональним деревом, що інтерпретується далі у вигляді кінцевого мікропрограмного автомата. В залежності від вимог цифрової системи мікропрограмний автомат може бути поданий у вигляді моделі Мілі, Мура або суміщеного мікропрограмного автомата (СМПА) [3].

Синтез і оптимізація автомата Мілі на CPLD розглядається у роботах [4, 5], використання моделі автомата Мура та методи її оптимізації знайшли відображення у [6, 7, 8]. Щодо моделі суміщеного мікропрограмного автомата, в роботах [9, 10, 11] пропонуються методи оптимізації СМПА за апаратними витратами.

Метою даної роботи є аналіз та експериментальне дослідження методів оптимізації схем суміщених мікропрограмних автоматів за критерієм зменшення апаратних витрат, визначення ефективності методів, а також умов доцільності їх застосування.

2. Суміщений мікропрограмний автомат

Головна особливість моделі СМПА полягає в суміщенні моделей Мілі та Мура, що означає можливість формування двох відповідних типів вихідних функцій в одному такті. СМПА може

бути поданий у вигляді восьмикомпонентного вектору:

$$S = \langle A, X, Y^1, Y^2, \delta, \lambda_1, \lambda_2, \alpha_1 \rangle. \quad (1)$$

Вектор (1) складається з наступних компонентів:

$A = \{a_1, \dots, a_M\}$ – множина внутрішніх станів;

$X = \{x_1, \dots, x_L\}$ – множина логічних умов;

Y^1 – множина вихідних змінних автомата Мілі;

Y^2 – множина вихідних змінних автомата Мура;

δ – функція переходів;

λ_1 – функція виходів автомата Мілі;

λ_2 – функція виходів автомата Мура;

$a_1 \in A$ – початковий стан.

Функція δ служить для визначення стану переходу $a_s \in A$ в залежності від поточного стану $a_m \in A$ і вектору вхідних змінних:

$$a_s = \delta(a_m, X). \quad (2)$$

Множина вихідних змінних $Y = Y^1 \cup Y^2$ складається з $N_1 = |Y^1|$ змінних автомата Мілі і $N_2 = |Y^2|$ змінних автомата Мура. При цьому $Y^1 \cap Y^2 \equiv \emptyset$ і $N_1 + N_2 = N$.

Функція λ_1 визначає вихідні змінні $y_n \in Y^1$:

$$y_n = \lambda_1(a_m, X). \quad (3)$$

Функція λ_2 визначає вихідні змінні $y_n \in Y^2$:

$$y_n = \lambda_2(a_m). \quad (4)$$

СМПА подається граф-схемою алгоритму (ГСА). Для знаходження функцій (2) – (4) необхідно побудувати за ГСА пряму структурну таблицю (ПСТ).

Для реалізації схеми суміщеного мікропрограминого автомата за ГСА необхідно виконати деякі проміжні етапи:

1. Позначення ГСА станами СМПА.
2. Кодування станів $a_m \in A$ двійковими кодами $K(a_m)$ розрядності R .
3. Побудова ПСТ автомата.
4. Отримання системи функцій, що відповідають (2) – (4).

Розглянемо структуру СМПА при реалізації його а базисі ПЛІС. Для кодування станів $a_m \in A$ використовуються внутрішні змінні $T_r \in T$, де $|T| = R$. Закодуємо стани мінімальним числом внутрішніх змінних:

$$R = \lceil \log_2 M \rceil. \quad (5)$$

Для збереження кодів станів використовується регістр (RG), що містить R тригерів. Як правило, при синтезі в базисі ПЛІС використовуються D тригери [4]. Для зміни вмісту RG необхідно задати функції збудження $D_r \in \Phi$, де $\Phi = \{D_1, \dots, D_r\}$. Зміна вмісту RG відбувається за сигналом синхронізації Clock. Для встановлення в RG коду стану $a_1 \in A$ використовується імпульс Start.

Після виконання етапів 1 – 2, ПСТ формується з використанням правил [3]. ПСТ є основою для отримання систем:

$$\Phi = \Phi(T, X); \quad (6)$$

$$Y^1 = Y^1(T, X); \quad (7)$$

$$Y^2 = Y^2(T). \quad (8)$$

Система (6) визначає функцію (2), система (7) – функцію (3) і система (8) – функцію (4).

Аналіз систем (6) – (8) показує, що функції Φ і Y^1 залежать від одних і тих самих змінних. У системі Y^2 відсутня залежність від вхідних змінних $x_i \in X$. Розглянемо структуру СМПА у вигляді композиції двох комбінаційних схем (KC1 і KC2) і регістра RG (рис. 1).

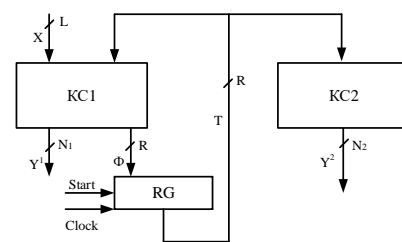


Рис. 1. Структурна схема суміщеного автомата

Згідно зі структурою, для реалізації схеми суміщеного автомата в базисі ПЛІС необхідно виконати такі етапи:

1. Формування множини станів A .
2. Кодування станів $a_m \in A$.
3. Формування ПСТ суміщеного автомата.
4. Отримання рівнянь для блока KC1.
5. Отримання рівнянь для блока KC2.
6. Реалізація схеми автомата.

В якості елементного базису розглянемо широко використовувані мікросхеми CPLD.

3. Базова структура СМПА в базисі CPLD

Внутрішня архітектура CPLD являє собою сукупність макрокомірок, що з'єднані між собою програмованою матрицею перемикань [2]. Для більшості CPLD функціональний блок має структуру, подібну PAL (Programmable Logic Array), тому його часто називають PAL-блоком.

Кожна макрокомірка являє q програмованих схем «І», що мають s загальних входів. Схеми «І» з'єднані схемою «АБО». Вхід схеми «АБО» може бути з'єднаний (або ні) зі входом Д-триггера. Тригер має входи синхронізації, обнуління і встановлення в одиницю. Макрокомірки пов'язані між собою за допомогою програмованої матриці міжз'єднань. Ця ж матриця пов'язує макрокомірки зі входами і виходами мікросхеми.

Позначимо символом PALer схему, що складається з елементів PAL. З формул (6) і (7) виходить, що функції (2) і (3) визначають схему з входами $x_l \in X$ і $T_r \in T$. Назвемо її схемою PALer1. Функція (8) відповідає схемі PALer2, що має тільки входи $T_r \in T$. Блок PALer1 містить регістр RG, який в цьому випадку є розподіленим між комірками PAL [9]. Таким чином, блок PALer1 повинен мати входи Clock и Start. Функції (6) – (8) визначають структурну схему СМПА (рис. 2). Як видно з рис. 2, виходами блоку PALer1 є внутрішні змінні $T_r \in T$, а не функції $D_r \in \Phi$.

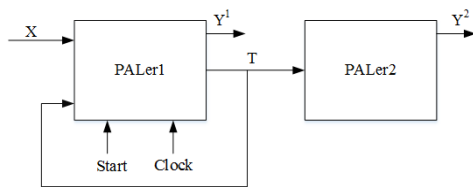


Рис. 2. Базова структурна схема СМПА U1

Функції (6), (7) залежать від термів F_h , що відповідають рядкам ПСТ:

$$F_h = A_m X_h (h = \overline{1, H}), \quad (9)$$

де A_m – кон'юнкція змінних $T_r \in T$, що відповідає стану $a_m \in A$ з h -го рядка ПСТ. Функції (6), (7) визначаються наступним чином:

$$D_r = \bigvee_{h=1}^H C_{rh} F_h (r = \overline{1, R}); \quad (10)$$

$$y_n = \bigvee_{h=1}^H C_{nh} F_h (n = \overline{1, N_1}). \quad (11)$$

Тут $C_{rh} (C_{nh})$ – булева змінна, що дорівнює 1, тоді і тільки тоді, якщо змінна $D_r (y_n)$ записана в h -ому рядку ПСТ.

Функція (8) залежить від термів A_m и визначається наступним чином:

$$y_n = \bigvee_{m=1}^M C_{nm} A_m (n = \overline{1, N_2}) \quad (12)$$

Тут C_{nm} – булева змінна, що дорівнює 1, тоді і тільки тоді, якщо змінна $y_n \in Y^2$ формується у стані $a_m \in A$ [9].

Утворимо множину функцій $O = \Phi \cup Y$, що містить K елементів, де $K = R + N$. Нехай $H(O_k)$ – множина термів в диз'юнктивній нормальній формі (ДНФ) функції $O_k \in O$. Очевидно, що зменшення числа термів необхідно тільки для функцій, у яких

$$q < H(O_k) (k = \overline{1, K}). \quad (13)$$

4. Метод синтезу СМПА зі зменшенням кількості термів з оптимізацією відносно вихідних функцій

Зменшення апаратних затрат в схемах СМПА напряму пов'язане зі зменшенням кількості термів в системах булевих функцій, що описують автомат.

Як правило, для мінімізації систем функцій $D_r \in \Phi$ і $y_n \in Y$ використовують різні методи кодування станів [10 - 12]. В даному методі пропонується оптимізація, заснована на наявності псевдоеквівалентних станів (ПЕС) автомата Мура [12].

Стани $(a_m, a_s) \in A$ є псевдоеквівалентними, якщо вершини ГСА, що ними позначені, пов'язані зі входом однієї і тієї ж вершини ГСА [12]. Це визначення дозволяє побудувати розбиття π_A множини A на класи ПЕС: $\pi_A = \{B_1, \dots, B_l\}$.

Закодуємо класи $b_i \in \pi_A$ двійковими кодами $K(B_i)$ розрядності

$$R_1 = \lceil \log_2 I \rceil. \quad (14)$$

Будемо використовувати для кодування ПЕС елементи множини $\tau = \{\tau_1, \dots, \tau_{R_1}\}$.

Очевидно, класи ПЕС відповідають станам автомата Мілі. Подамо функції (16), (7) в наступному вигляді:

$$\Phi = \Phi(\tau, X); \quad (15)$$

$$Y^1 = Y^1(\tau, X). \quad (16)$$

Для кодування класів $b_i \in \pi_A$ необхідно реалізувати систему функцій

$$\tau = \tau(T). \quad (17)$$

Порівняння функцій (6) і (15), а також (7) і (16) показує, що змінні $\tau_r \in \tau$ замінюють змінні $\tau_r \in T$. При цьому функції (8) не змінюються.

Структурна схема СМПА U2, перетворена за цим методом подана на рис. 3.

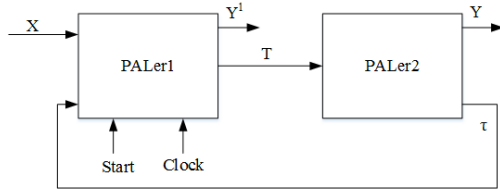


Рис. 3. Структурна схема СМПА U2

В СМПА U2 блок PALer1 реалізує функції (15) і (16). Блок PALer2 реалізує системи (8) і (17).

Для синтезу схеми U2 за ГСА необхідно побудувати ПСТ, яка відрізняється від ПСТ автомата U1. Відмінність зводиться до заміни стовпців a_m і $K(a_m)$ стовпцями B_i і $K(B_i)$ відповідно. При цьому виконується умова

$$H_1 < H, \quad (18)$$

де H_1 – число рядків ПСТ автомата U2, що є еквівалентним автомату U1. Очевидно, автомати U1 і U2 є еквівалентними, якщо вони синтезуються за однією і тією ж ГСА [9].

Запропонований підхід дозволяє для еквівалентних автоматів U1 і U2 виконати наступне:

1. Зменшити число термів в функціях (15), (16) у порівнянні з функціями (6) і (7), відповідно. Це впливає із умови (18).

2. Кодування станів $a_m \in A$ автомата U2 можна виконати так, щоб зменшити число термів в функціях системи (8). Це можливо, бо коди $K(a_m)$ практично не впливають на число термів в функціях (15) и (16).

Недоліком U2 є наявність системи (17), для реалізації якої необхідні деякі ресурси кристала. Однак кодування станів можна виконати таким чином, щоб мінімізувати число макрокомірок PAL, що необхідні для реалізації системи (17).

Синтез автомата U2 за даним методом, містить наступні етапи:

1. Формування позначеної ГСА.
2. Формування розбиття π_A на множині A.
3. Кодування класів $b_i \in \pi_A$.
4. Кодування станів СМПА.
5. Формування прямої структурної таблиці автомата U2.
6. Формування систем функцій, що задають схему U2.
7. Реалізація схеми СМПА в заданому базисі.

5. Метод синтезу СМПА зі зменшенням кількості термів з пріоритетом оптимізації відносно класів псевдоеквівалентності

Наступний підхід також заснований на використанні ПЕС.

Існує дві майже однакові структурні схеми U1 і U3. Різниця між U1 і U3 полягає у зменшенні кількості термів F_h у функціях. В обох випадках вираз A_m містить терми:

$$A_m = \bigwedge_{r=1}^R T_r^{l_{mr}}, \quad (m = \overline{1, M}). \quad (19)$$

Кодування станів з пріоритетом відносно класів псевдоеквівалентності дозволяє зменшити кількість рядків ПСТ до H_1 у випадку з U3 [13]. В свою чергу, це зменшує кількість термів в функціях (6) – (7) для U3 у порівнянні з еквівалентним СМПА U1. Зазначимо, що U1 і U3 вважаються еквівалентними, якщо вони синтезовані з використанням однієї і тієї ж ГСА.

Для синтезу СМПА U3 необхідно виконати такі кроки:

1. Позначення початкової ГСА станами автомата Мура.
2. Формування розбиття π_A на множині A.
3. Кодування класів.
4. Побудова прямої структурної таблиці для U2.
5. Формування систем функцій (6) – (8).
6. Реалізація схеми СМПА базисі CPLD.

Зауважимо, що кількість використовуваних макрокомірок може бути зменшена, якщо існує взаємозалежна частина $F_1 \vee F_2$ для деяких кон'юнкцій T_1 і T_2 . Схема для їх реалізації може бути побудована так, як показано на рис. 4.

Таким чином, даний метод орієнтований безпосередньо на внутрішню архітектуру CPLD.

Таким чином, даний метод орієнтований безпосередньо на внутрішню архітектуру CPLD.

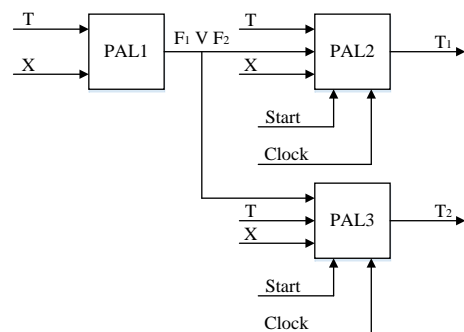


Рис. 4. Логічна схема для кон'юнкцій T1 і T2

6. Метод синтезу СМПА з кодуванням наборів вихідних змінних

Для зменшення числа термів в системах вихідних змінних автомата Мілі і функцій збудження пам'яті в [14] пропонується метод синтезу СМПА з кодуванням наборів вихідних змінних.

Для реалізації цього методу початкова ГСА позначається станами автомата Мілі [3], утворюючи множину $A = \{a_1, \dots, a_M\}$. В операторних вершинах ГСА записуються змінні $y_n \in Y^1$, а біля операторних вершин – змінні $y_n \in Y^2$. Утворюється множина наборів вихідних змінних, тобто наборів вихідних сигналів (НВС) $Y_q \subseteq Y^2$. Ці набори записуються біля операторних вершин ГСА. Якщо ГСА включає Q різних НВС, набори Y_q кодується двійковими кодами $K(Y_q)$ розрядності R_Q :

$$R_Q = \lceil \log_2 Q \rceil. \quad (20)$$

Для кодування НВС використовуються змінні $z_r \in Z$, де $|Z| = R_Q$.

Тоді система (8) може бути подана в такому виді:

$$Y^2 = Y^2(Z). \quad (21)$$

При цьому кожна змінна $y_n \in Y^2$ визначається наступним чином:

$$y_n = \bigvee_{q=1}^Q C_{nq} Z_q, \quad (n = \overline{1, N_2}). \quad (22)$$

Тут C_{nq} – булева змінна, що дорівнює одиниці тоді, і тільки тоді, якщо $y_n \in Y_q$; Z_q – кон'юнкція змінних $z_r \in Z$, яка відповідає коду $K(Y_q)$; $q \in \{1, \dots, Q\}$.

Для реалізації системи (21) необхідно реалізувати систему функцій

$$Z = Z(T, X). \quad (23)$$

Така модель СМПА реалізується структурною схемою U4 (рис. 5).

В автоматі U4 блок PALer1 реалізує системи (6), (7) і (23). Блок PALer2 реалізує систему (21). Виходи макрокомірок, які формують функції (23), пов'язані з входами D тригерів. Таким чином, змінні (21) не змінюють своїх значень, поки не змінюється стан автомата. Очевидно, автомати U1 і U4 є еквівалентними, якщо вони синтезуються за одною і той самою ГСА Г.

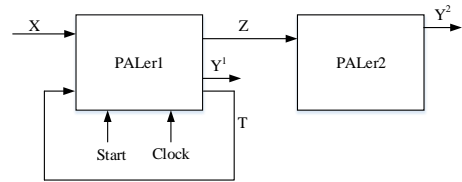


Рис. 5. Структурна схема СМПА U4

Підхід, що пропонується, надає такі можливості для еквівалентних автоматів U1 і U4:

1. Зменшення кількості рядків ПСТ автомата U4 в порівнянні з числом H , що також дозволяє зменшити кількість термів в системах (6) – (7) автомата U4 в порівнянні з U1.

2. Кодування станів відокремлене від кодування наборів вихідних сигналів. Таким чином, кодування станів необхідно виконати так, щоб зменшити число термів в кожній з функцій (6) – (7). Кодування НВС, в свою чергу, необхідно виконати так, щоб зменшити число термів в кожній з функцій $y_n \in Y^2$ автомата Мура.

Недоліком схеми автомата U4 є наявність системи (21), для реалізації якої необхідні певні ресурси кристала. Однак цей недолік компенсується зменшенням складності схем, що реалізують функції $D_r \in \Phi$, $y_n \in Y^1 \cup Y^2$.

Метод синтезу автомата U4, включає наступні етапи [14]:

1. Формування множини станів автомата Мілі для заданої ГСА.
2. Кодування станів $a_m \in A$.
3. Формування наборів вихідних змінних $Y_q \subseteq Y$.
4. Кодування наборів вихідних змінних.
5. Формування прямої структурної таблиці СМПА U4.
6. Формування систем функцій, що задають схему СМПА.
7. Реалізація схеми в заданому елементарному базисі.

Останній пункт методу – це, як зазвичай, реалізація схеми в заданому елементарному базисі.

7. Результати експериментальних досліджень

В якості початкових даних для досліджень був використаний фрагмент реального алгоритму керування, внаслідок чого отримали ГСА з параметрами: кількість станів $|A| = 37(a_0, \dots, a_{36})$; кількість логічних умов $|X| = 20(x_1, \dots, x_{20})$; кількість вихідних функцій автомата Мура

$|Y_1| = 5(y_6, \dots, y_{10})$; кількість вихідних функцій автомата Мілі $|Y_2| = 5(y_1, \dots, y_5)$.

Для отримання базової моделі СМПА стани автомата були закодовані тривіально, побудована пряма структурна таблиця та системи функцій (6) – (8). Отримані системи представлені синтезованими конструкціями мови VHDL та імплементовані в мікросхеми серій MAX V фірми Altera, а також CoolRunner 2 і XC9500XL фірми Xilinx.

Для апробації методу синтезу СМПА зі зменшенням кількості термів з пріоритетом оптимізації відносно вихідних функцій базова ГСА була перетворена до відповідного виду, визначені псевдоеквівалентні стани $|B| = 13(B_0, \dots, B_{12})$. Стани A закодовані за допомогою карт Карно з пріоритетом оптимізації відносно вихідних функцій Y^1 . Стани B закодовані тривіально, з використанням перетворювача кодів. За даною ГСА була побудована ПСТ, отримані системи функцій та VHDL опис, проведена імплементация в мікросхеми, що використовувалися для базової структури.

При оптимізації за методом синтезу СМПА зі зменшенням кількості термів з пріоритетом оптимізації відносно класів псевдоеквівалентності, кодування станів було виконане за допомогою карт Карно з оптимізацією відносно класів B. Результати імплементации наведені в табл. 1.

Для експериментального підтвердження ефективності оптимізації за методом синтезу СМПА з кодуванням наборів вихідних змінних були проведені перетворення базової ГСА. Після того, як ГСА була розмічена станами автомата Мілі число станів автомата склало $|A| = 12(a_0, \dots, a_{11})$. Інші параметри не змінилися.

Після опрацювання ГСА та імплементации проекту були отримані дані, що подані в табл. 1.

Таблиця 1
Результати імплементации

Модель СМПА	Altera MAX V [Logic elements]	Xilinx CoolRunner 2 [Macrocells]	Xilinx XC9500XL [Macrocells]
U1	123/160 (77%)	21/64 (33%)	19/72 (27%)
U2	106/160 (66%)	20/64 (32%)	23/72 (32%)
U3	71/160 (44%)	18/64 (29%)	16/72 (23%)
U4	69/160 (43%)	15/64 (24%)	14/72 (20%)

Результати досліджень щодо зменшення апаратних затрат при використанні структур U1 – U4 СМПА в різних серіях мікросхем CPLD подані на рис. 6 – 8.

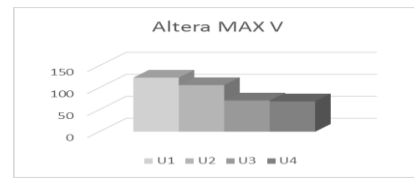


Рис. 6. Площа, займана структурами U1 – U4 при реалізації на мікросхемах серії MAX V

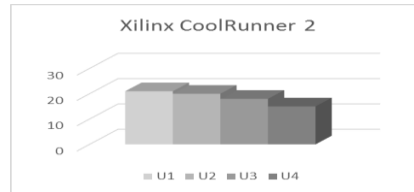


Рис. 7. Площа, займана структурами U1 – U4 при реалізації на мікросхемах серії CoolRunner 2

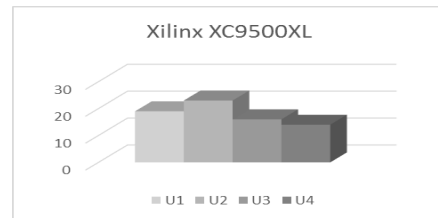


Рис. 8. Площа, займана структурами U1 – U4 при реалізації на мікросхемах серії XC9500XL

Одиницею виміру площі в даних випадках є кількість логічних елементів або макрокомірок.

8. Обговорення

Проаналізувавши результати досліджень можна зробити такі висновки:

1. Найбільш оптимально розташувати логічну схему СМПА, оптимізовану за критерієм апаратних витрат дозволяють мікросхеми Xilinx серії XC9500XL. Так, навіть використання базової структури U1 дозволяє зменшити займану площу мікросхеми з 77% до 27% порівняно з мікросхемами Altera серії MAX V.

2. З розглянутих моделей СМПА найбільш ефективним є використання структури U4. Використання цієї структури дає позитивний результат для всіх сімейств мікросхем, що досліджувались.

Так, використання структури U4 дозволяє зменшити апаратні витрати в середньому на 33% порівняно з базовою структурою. Найбільшу ефективність така структура має для мікросхем фірми Altera MAX V – 43,9%. Для Xilinx CoolRunner 2 – 28,6%, для Xilinx XC9500XL – 26,3%.

Заключення

Основною метою даної роботи було дослідження методів зменшення апаратурних витрат при реалізації схем суміщених мікропрограмних автоматів в базисі CPLD. Вихідні дані для досліджень подані у вигляді граф-схеми алгоритму, яка була опрацьована за різними методами і імплементована у мікросхеми CPLD, що представлені на ринку сучасної електроніки.

Дослідження показали, що навіть при використанні базової моделі, сімейства фірми Xilinx забезпечують кращу економічну ефективність порівняно з сімейством Altera MAX V мають. При ціні 5,1 \$ за мікросхему Altera 5M160ZE64A5 і 6,37 \$ за мікросхему Xilinx XC9572XL-5-VQ64 реалізація заданого алгоритму потребує 77% і 27% площі мікросхеми відповідно. Тобто використання Xilinx дорожче на 8%, але дозволяє на 50% ефективніше розташувати схему.

Щодо методів оптимізації за апаратурними витратами, то дослідження показали, що їх використання найбільш ефективно саме для мікросхем фірми Altera. Так, метод синтезу СМПА зі зменшенням кількості термів з оптимізацією відносно вихідних функцій дозволяє зменшити використовувану площу на 11%, метод синтезу СМПА зі зменшенням кількості термів з пріоритетом оптимізації відносно класів псевдоеквівалентності – на 33% відносно базового, а метод синтезу СМПА з кодуванням наборів вихідних змінних – на 34% відносно базового.

Подальшим перспективним напрямком роботи автори вважають оптимізацію схем СМПА в контексті «зелених» обчислень.

Перелік використаної літератури

1. Sunggu, Lee Advanced Digital Logic Design Using VHDL, State Machines, and Synthesis for FPGA's [Text] / Lee Sunggu. - Thomson-Engineering, 2005. – 488 pp.
2. Czerwinski, R. Finite State Machine Logic Synthesis for Complex Programmable Logic Devices [Text] / R. Czerwinski, D. Kania. – Springer Science & Business Media, 2013. – 172 pp. doi: [10.1007/978-3-642-36166-1](https://doi.org/10.1007/978-3-642-36166-1)
3. Baranov, S. Logic and System Desing of Digital Systems [Text] / S. Baranov. – Tallinn: TUT Press, 2008. – 267 pp.
4. Sklyarov, V. Synthesis and Optimization of FPGA-Based Systems [Text] / V. Sklyarov, I. Sklyarova, A. Barkalov, L. Titarenko. – Springer International Publishing, 2014. – 432 p. doi: [10.1007/978-3-319-04708-9](https://doi.org/10.1007/978-3-319-04708-9)

5. Micheli, G. Synthesis and Optimization of Digital Circuits [Text] / G. Micheli. – New York: Mc Graw-Hill, 1994. – 636 p.

6. Зеленева, И. Я. “Экспериментальное исследование методов оптимизации апаратурных затрат при реализации управляющего автомата Мура на CPLD” [Текст] / И. Я. Зеленева, С. С. Грушко, Д. В. Арапин // Збірник наукових праць «Системи обробки інформації», Випуск 2 (148), Харків, 2017. – с. 34–41.

7. Баркалов, А. А. “Уменьшение апаратурных затрат в схеме автомата Мура с использованием особенностей базиса CPLD” [Текст] / А. А. Баркалов, А. В. Матвиенко, С. А. Цололо // Комп'ютерні засоби, мережі та системи. — 2008. — № 7. — С. 86–94.

8. Баркалов, А. А. "Оптимизация схемы автомата Мура, реализуемой в базисе ПЛИС" [Текст] / Баркалов А. А., Мальчева Р. В., Солдатов К. А.. // Радиоэлектроника, информатика, управление, no. 1 (26), 2012, pp. 44–47.

9. Баркалов, А. А. “Метод уменьшения числа термов при реализации схемы совмещенного микропрограммного автомата в базисе CPLD” [Текст] / А. А. Баркалов, Л. А. Титаренко, И. Я. Зеленева, С. С. Грушко // Вісник НТУ «ХП», Харків, 2016, №49 (1221). – с. 25–31.

10. Villa, T. “NOVA: state assignment of finite state mashines for optimal two-level implementation” [Text] / T. Villa, A. Sangiovanni-Vincentelli // IEEE Transactions on CAD of Integrated Circuits and Systems, 1990, № 9. – 34–48 pp. doi: [10.1109/43.59068](https://doi.org/10.1109/43.59068)

11. Yang, S. “Optimum and suboptimum algorithms for input encoding and its relationships to logic minimization” [Text] / S. Yang, M. Ciesielski // IEEE Transactions on CAD of Integrated Circuits and Systems. – 1991, № 10 – 117–131 pp. doi: [10.1109/43.62787](https://doi.org/10.1109/43.62787)

12. Баркалов, А. А. “Принципы оптимизации логической схемы микропрограммного автомата Мура” [Текст] / А. А. Баркалов // Кибернетика и системный анализ. – 1998, №1. – С. 65-72.

13. Barkalov, A. “Implementing combined FSM with CPLDs” / A. Barkalov, L. Titarenko, I. Zeleneva, S. Hrushko // International Journal of Software Engineering and Computer Systems (IJSECS), Paper 006, Volume 4, Universiti Malaysia Pahang, 2017.

14. Кодування наборів вихідних сигналів в схемі суміщеного мікропрограмного автомата на CPLD [Текст] / А. А. Баркалов, Л. А. Титаренко, І. Я. Зеленьова, С. С. Грушко // У друці.

References

1. Sunggu, Lee. (2005), *Advanced Digital Logic Design Using VHDL, State Machines, and Synthesis for FPGA's*, Thomson-Engineering, 488 pp.
2. Czerwinski, R., Kania, D. (2013), *Finite State Machine Logic Synthesis for Complex Programmable Logic Devices*, Springer Science & Business Media, 172 pp. doi: [10.1007/978-3-642-36166-1](https://doi.org/10.1007/978-3-642-36166-1)
3. Baranov, S. (2008), *Logic and System Design of Digital Systems*, TUT Press, Tallinn, 267 pp.
4. Sklyarov, V., Sklyarova, I., Barkalov, A., Titarenko, L. (2014), *Synthesis and Optimization of FPGA-Based Systems*, Springer International Publishing, 432 pp. doi: [10.1007/978-3-319-04708-9](https://doi.org/10.1007/978-3-319-04708-9)
5. Micheli, G. (1994), *Synthesis and Optimization of Digital Circuits*, Mc Graw-Hill, New York, 636 pp.
6. Zeleneva, I. J., Hrushko, S.S., Arapin, D.V. (2017), "Experimental study of methods of optimization of hardware costs in the implementation of the Moore control automaton on CPLD" ["Eksperimentalnoe issledovanie metodov optimizatsii apparaturnykh zatrat pri realizatsii upravlyayuschego avtomata Mura na CPLD"] *Collection of scientific works "Systems of information processing"*, Issue 2 (148), Kharkiv, pp. 34–41
7. Barkalov, A. A., Matvienko, A.V., Tsololo, S.A. (2008), "Reduction of hardware costs in the Moore FSM circuit using the CPLD basis features" ["Umenshenie apparaturnykh zatrat v sheme avtomata Mura s ispolzovaniem osobennostey bazisa CPLD"], *Computer tools, networks and systems*, No. 7, pp. 86–94.
8. Barkalov, A. A., Malcheva R. V., Soldatov K. A. (2012), "Optimization of the Moore FSM scheme realized in the FPGA basis", *Radio Electronics, Computer Science, Control*, no. 1 (26), pp. 44–47.
9. Barkalov, A. A., Titarenko, L. A., Zeleneva, I. J., Hrushko, S. S. (2016), "The method of decreasing the number of terms when implementing a scheme of a combined firmware in CPLD basis" ["Metod umensheniya chisla termov pri realizatsii shemyi sovmeshchennogo mikroprogrammno avtomata v bazise CPLD"] *Bulletin of the NTU "KhPI"*, №49 (1221), pp. 25–31.
10. Villa, T., Sangiovanni-Vincentelli A. (1990), "NOVA: state assignment of finite state machines for optimal two-level implementation" *IEEE Transactions on CAD of Integrated Circuits and Systems*, №9, 34–48 pp. doi: [10.1109/43.59068](https://doi.org/10.1109/43.59068)
11. Yang, S., Ciesielski, M. "Optimum and suboptimum algorithms for input encoding and its relationships to logic minimization", *IEEE Transactions on CAD of Integrated Circuits and Systems*, №10, 117-131 pp. doi: [10.1109/43.62787](https://doi.org/10.1109/43.62787)
12. Barkalov, A. A. (1998), "Principles of optimizing the logic circuit of the Moore FSM" [Printsipy optimizatsii logicheskoy shemy mikroprogrammno avtomata Mura], *Cybernetics and system analysis*, №1, pp. 65–72.
13. Barkalov, A., Titarenko, L., Zeleneva, I., Hrushko S. (2017) "Implementing combined FSM with CPLDs", *International Journal of Software Engineering and Computer Systems (IJSECS)*, Paper 006, Volume 4, Universiti Malaysia Pahang.
14. Barkalov, A. Titarenko, L., Zeleneva I., Hrushko, S. "Encoding of output signal sets in the scheme of the combined FSM on the CPLD", In press.

ANALYSIS OF METHODS FOR HARDWARE AMOUNT REDUCTION IN THE COMBINED FSM SCHEME IMPLEMENTED IN THE CPLD

S. Hrushko, I. Zeleneva

Zaporizhzhya National Technical University

Abstract. *The model of finite state machine (FSM) is used in a lot of cases of control unit design. In this article a short review of the control unit models is given. The attention is paid to the model of combined finite state machine (CFSM), which is considered in detail. The main feature of the CFSM model is to combine the Mealy and Moore models, which means the ability to form two corresponding types of output functions in one cycle.*

There is some optimization problem connected with CFSM synthesis, and choice of its solution depends on the hardware base of implementation.

The modern hardware platform is represented by chips of programmable logic FPGA and CPLD. CPLD type circuits are popular due to their availability. CPLD, like any other structure, has certain design parameters, so the control algorithms that are implemented on CPLD, can be optimized for these parameters.

The criteria of optimization may be different: reducing the power consumption of the FSM circuit, productivity increasing, etc. The choice of a priority criterion depends on the overall concept and the ultimate goal of the project. In this article we discuss the methods of logic circuit optimization of CFSM type control unit. In our case the main criterion of optimization is the reduction of hardware amount.

The main task of this research is to reduce the hardware amount of combined FSM realization on the CPLD. The architectural features of complex programmable logic device are analyzed. Based on the analysis, the basic model of the implementation of combined finite state machine in CPLD is proposed. The reduction of hardware amount of the CFSM scheme is directly related to the decreasing of terms number in the Boolean functions system describing an automaton. The optimization methods for basic CFSM model are proposed, such as: the method of decreasing the number of optimized terms relative to the output function; the method of decreasing the number of terms relative to the classes of pseudoequivalence; the method of output variables sets encoding.

Studies have been conducted to evaluate the effectiveness of each method. For the each discussed model of CFSM, the states of the automaton have been encoded trivially, a direct structured table and functional systems are developed. The resulting systems are represented by synthesized VHDL constructs and implemented in Altera MAX V chips, as well as Xilinx CoolRunner 2 and XC9500XL.

Based on the results of the study, it is clear that the most optimal combined FSM schemes are implemented in microcircuits Xilinx XC9500XL, and the most effective method is the method of combined FSM synthesis with the output variable sets encoding.

Keywords: combined FSM, graph-scheme of the algorithm, CPLD, PAL, coding of states, pseudoequivalent states.

АНАЛИЗ МЕТОДОВ УМЕНЬШЕНИЯ АППАРАТУРНЫХ ЗАТРАТ ПРИ РЕАЛИЗАЦИИ СХЕМ СОВМЕЩЕННЫХ МИКРОПРОГРАММНЫХ АВТОМАТОВ НА CPLD

С. С. Грушко, И. Я. Зеленева

Запорожский национальный технический университет

Аннотация. Приведен обзор моделей устройств управления. Подробно рассмотрена модель совмещенного микропрограммного автомата. Проанализирована архитектура программируемых логических интегральных схем типа CPLD. Предложенные базовая и оптимизированные структуры реализации схем устройств управления в базе CPLD. Получены экспериментальные данные, позволяющие оценить степень эффективность этих структур за критерием аппаратных затрат. Исследование проведено для микросхем CPLD фирм Xilinx и Altera.

Ключевые слова: СМПА, граф-схема алгоритма, CPLD, PAL, кодирование состояний, ПЭС.

Отримано «15.03.2018»



Грушко Світлана Сергіївна, асистент кафедри комп'ютерних систем та мереж Запорізького національного технічного університету. Вул. Жуковського, 64, Запоріжжя, Україна, E-mail: grushko_ss@i.ua, тел. +38-061-769-82-49

Svitlana Hrushko, assistant of the Department of Computer Systems and Networks, Zaporizhzhya National Technical University, Zhukovskiy street, 64, Zaporizhzhya, Ukraine, E-mail: grushko_ss@i.ua, tel. +38-061-769-82-49

ORCID ID: 0000-0002-0064-408X



Зеленьова Ірина Яківна, кандидат технічних наук, доцент кафедри комп'ютерних систем та мереж Запорізького національного технічного університету. Вул. Жуковського, 64, Запоріжжя, Україна, E-mail: irina.zeleneva@gmail.com, тел. +38-061-769-82-49

Irina Zeleneva, PhD, assistant professor of the Department of Computer Systems and Networks, Zaporizhzhya National Technical University, Zhukovskiy street, 64, Zaporizhzhya, Ukraine, E-mail: irina.zeleneva@gmail.com, tel. +38-061-769-82-49

ORCID ID: 0000-0002-4042-4540