

Грушко С.С.

Запорізький національний технічний університет

КЛАСИФІКАЦІЯ СТРУКТУР СУМІЩЕНИХ МІКРОПРОГРАМНИХ АВТОМАТІВ ПРИ РЕАЛІЗАЦІЇ У БАЗИСІ FPGA

Проаналізовані способи реалізації логічної схеми суміщеного мікропрограмного автомата (СМПА) у базисі FPGA з урахуванням особливостей внутрішньої структури мікросхем, а також відповідні способи оптимізації схем. Запропонована класифікація структур СМПА за критерієм способу використання ресурсів мікросхеми для розташування окремих блоків логічної схеми. Наведені результати досліджень розглянутих структур СМПА, на підставі чого визначені умови доцільності використання для кожної структури.

Ключові слова: мікропрограмний пристрій управління, суміщений мікропрограмний автомат, LUT, вбудовані блоки пам'яті, FPGA, ПЛІС.

Постановка проблеми. Пристрій керування є однією з найважливіших частин будь-якої сучасної цифрової системи (далі – ЦС) [1]. Що стосується способів проектування реалізації алгоритмів керування, то зважаючи на те, що цифрові системи, включно із керуючою частиною, стають дедалі складнішими, наряду із широко відомими моделями реалізації алгоритмів керування у вигляді автоматів Мілі і Мура, актуальним у практиці проектування стає спосіб подання керуючого алгоритму у вигляді суміщеного мікропрограмного автомата (далі – СМПА) [2].

У зв'язку з технологічним прогресом в області виготовлення мікросхем, останнім часом набуває поширення практика реалізації окремих функціональних блоків ЦС, або всієї системи у цілому на програмованих логічних інтегральних схемах (далі – ПЛІС), які є найбільш широко використовуваним базисом у даний час. Архітектурною особливістю ПЛІС є внутрішня матрична структура, основою якої служать макрокомірки PLA/PAL для CPLD або мікрокомірки типу LUT для FPGA. Внутрішня передача сигналів забезпечується матрицею міжз'єднань [3; 4]. Для ефективного використання внутрішніх ресурсів ПЛІС з метою зменшення апаратних витрат [1; 2] використовуються різні способи оптимізації, що охоплюють здебільшого моделі Мілі та Мура. Аналіз літературних джерел показав, що кількість публікацій щодо СМПА є дуже обмеженою. Це дає підстави для більш детального аналізу актуальної задачі оптимізації схем пристроїв керування, поданих моделями суміщених мікропрограмних автоматів.

Аналіз базових структур СМПА

СМПА являє собою керуючий автомат, що подається у вигляді пристрою з одним зовнішнім входом, на який надходять вхідні сигнали X , і двома зовнішніми виходами, один з яких реалізує функцію виходу Y^1 , характерну для автомата Мілі, а інший – функцію виходу Y^2 , характерну для автомата Мура [5, с. 27]. Крім того, використовуються сигнали внутрішнього зворотного зв'язку для організації переходів автомата із поточного стану у наступний.

Математична модель СМПА докладно описана у публікаціях [1; 5].

Розглянемо компоненти моделі, які мають безпосереднє відношення до даних досліджень.

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

$$a_s = \delta(a_m, X). \quad (1)$$

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 $Y^1 \cap Y^2 = \emptyset$ та $N^1 + N^2 = N$, де N – загальна кількість вихідних сигналів СМПА.

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

$$y_n = \lambda_1(a_m, X). \quad (2)$$

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

$$y_n = \lambda_2(a_m, X). \quad (3)$$

Розглянемо структуру СМПА при реалізації його у базисі ПЛІС. Для кодування станів $a_m \in A$ використовуються внутрішні змінні $T_r \in T$, де $|T| = R$. Розрядність внутрішніх змінних визначається як:

$$R = \lceil \log_2 M \rceil, \quad (4)$$

де M – число станів автомата [1].

Для збереження кодів станів використовується регістр (RG), що містить R тригерів. Як правило, при синтезі у базисі ПЛІС використовуються D-тригери [3]. Для зміни вмісту RG необхідно задати функції збудження $D_r \in \Phi$, де $\Phi = \{D_r, \dots, D_r\}$. Зміна вмісту RG відбувається за сигналом синхронізації Clock. Для встановлення у RG коду стану $a_i \in A$ використовується імпульс Start.

У загальному випадку СМПА описується системами [1]:

$$\Phi = \Phi(T, X); \quad (5)$$

$$Y^i = Y^i(T, X); \quad (6)$$

$$Y^2 = Y^2(T). \quad (7)$$

Система (5) визначає функцію (1), система (6) – функцію (2) і система (7) – функцію (3).

Аналіз систем (5) – (7) показує, що функції Φ і Y^i залежать від одних і тих самих змінних. У системі (7) відсутня залежність вихідних сигналів від вхідних змінних $x_i \in X$, де $|X|=L$. Розглянемо уявлення СМПА у вигляді композиції двох комбінаційних схем (KC1 і KC2) і регістра RG (рис. 1).

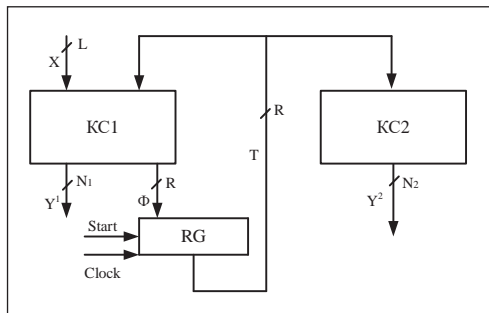


Рис. 1. Узагальнена базова структура суміщеного автомата

Далі розглянемо, яким чином можна реалізувати узагальнену структуру СМПА (рис. 1) з урахуванням особливостей матричної внутрішньої архітектури мікросхем ПЛІС.

Мікросхеми FPGA є поширеною платформою для реалізації цифрових пристроїв і зокрема пристроїв керування. Розподіл логічної схеми пристрою між структурними елементами мікросхеми виконується внутрішніми ресурсами систем автоматизованого проектування (далі – САПР). Оптимізація розташування логічної схеми ЦС виконується за алгоритмами, що закладені у САПР і, з огляду на результати проведених досліджень, має досить невисоку ефективність. Тому виникає задача оптимального розташування функціональних блоків схеми СМПА на ресурсах ПЛІС типу

FPGA, рішення якої можна було б закласти ще на етапі проектування схеми.

З [3 – 6] відомо, що для реалізації схеми СМПА можна використовувати три компоненти FPGA: елементи табличного типу LUT (Look Up Table), вбудовані блоки пам'яті ЕВМ (Embedded Memory Block) і програмовані міжз'єднання [5; 6; 7].

Елементи LUT – це оперативні запам'ятовуючі пристрої (далі – ОЗП), кожен з яких має S входів і один вихід. При цьому число входів визначає максимальну кількість аргументів булевої функції, яка може бути реалізована на одному LUT елементі. Як правило, число входів $S \leq 6$ [3; 4; 7].

Блоки ЕВМ – це пам'ять на базі ОЗП, що має S_A входів і t_F виходів. Загальна ємність блоків є константою і визначається наступним виразом:

$$V_0 = 2^{S_A} \cdot t_F \quad (8)$$

Параметри S_A і t_F можуть бути обрані зі стандартних пар виду $\langle S_A, t_F \rangle$ що приведені у технічній документації до мікросхем. Зазвичай це $\langle 256, 8 \rangle$, $\langle 512, 4 \rangle$, $\langle 1024, 2 \rangle$, $\langle 2048, 1 \rangle$ [3, 4, 7].

Введемо термін LUTer для схеми, що складається з елементів LUT і EMBer для схеми, що складається з елементів EMB.

Структуру СМПА, в якій при реалізації блоків KC1 і KC2 використовуються тільки елементи LUT, назовемо L-based структурою СМПА.

E-based структурою пропонується називати структуру, в якій блоки KC1 і KC2 реалізуються на елементах вбудованої пам'яті – EMB.

В обох випадках регістр RG в явному вигляді відсутній. У L-based моделі тригери розподілені між логічними елементами, в які входять елементи LUT. В E-based моделі використовується можливість синхронізації блоків EMB [2; 3; 4; 7].

Недоліком L-based і E-based моделей є надмірність реалізації для практичних прикладів. Це означає, що для тривіальної (однорівневої) реалізації схем автоматів реальної складності [1] потрібно більше логічних елементів, ніж для багаторівневих структур [2].

Для зменшення числа елементів у логічній схемі СМПА пропонується використовувати принцип гетерогенної реалізації [2]. Очевидно, що зменшення числа елементів веде до зменшення площі кристала FPGA, яку займає схема СМПА.

Враховуючи, що системи (5, 6) мають однакові аргументи, їх доцільно реалізовувати на однотипних елементах [6]. Якщо використовувати блоки EMB для реалізації KC1 і елементи LUT для побудови KC2, отримаємо структуру, яку будемо називати EL-based структурою СМПА.

При реалізації схем СМПА у базисі FPGA, з огляду на практичні результати досліджень автора, також пропонується структура, в якій елементи LUT використовуються для реалізації блоку KC1, а блоки EMB – для реалізації KC2. Будемо називати таку модель LE-based структурою СМПА.

Отже, розглянуто чотири структури СМПА (L-based, E-based, LE-based та EL-based), які можуть служити як базові.

Використання базових структур та способів їх оптимізації залежать від параметрів алгоритмів, що реалізуються на ПЛІС, а також від технічних характеристик мікросхем, що для цього використовуються.

Оптимізовані структури СМПА

При використанні базових моделей можуть виникати випадки, коли кількість вхідних сигналів більша за кількість входів матричної структури ПЛІС.

Для оптимізації кількості входів автомата пропонується використання методу заміни логічних умов [8].

Нехай $X(a_m) \subseteq X$ – множина логічних умов, тобто вхідних змінних, що визначають переходи зі стану $a_m \in A$. Знайдемо параметр

$$G = \max(|x(a_1)|, \dots, |x(a_m)|). \quad (9)$$

Параметр G визначає множину додаткових змінних $P = \{P_1, \dots, P_G\}$, яка може замінити множину X [2]. Тоді структурна схема СМПА прийме вигляд G-структури, поданий на рис. 2.

У такій моделі блок KC1 виконує перетворення $X \rightarrow P$. Це призводить до реалізації системи

$$P = P(T, X). \quad (10)$$

Блок KC2 реалізує системи функцій (5, 6), а блок KC3 – систему функцій (7).

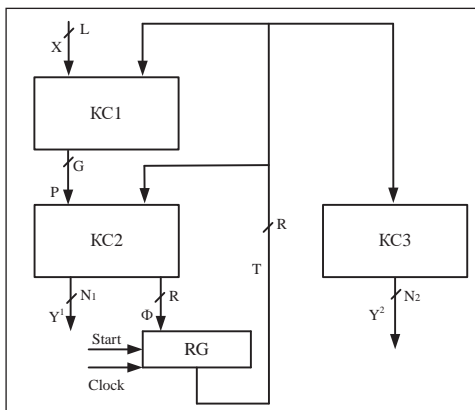


Рис. 2. G-структура СМПА

Аналіз [9] показує, що $G \ll L$. Це значно розширює круг автоматів, що можуть бути реалізовані у базисі FPGA.

Для подальшого зменшення кількості термів у системі функцій збудження СМПА пропонується використовувати метод псевдоеквівалентних станів (ПЕС) [10].

При використанні цього методу у ГСА визначаються так звані псевдоеквівалентні стани (вершини, що позначені цими станами, зв'язані зі входом однієї і тієї ж вершини ГСА), виконується формування класів псевдоеквівалентності та їх кодування.

Кількість класів ПЕС завжди менша, ніж кількість станів автомата, тому розрядність коду класу менша за розрядність коду станів. При застосуванні цього методу отримуємо В-структуру СМПА, подану на рис. 3.

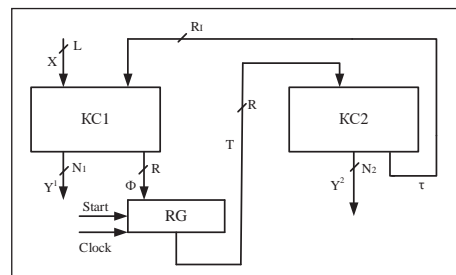


Рис. 3. В-структура СМПА

В автоматі В-структури (рис. 3) KC1 реалізує системи функцій (5, 6). Блок KC2 реалізує системи (7) і (11):

$$\tau = \tau(T). \quad (11)$$

Комплексне використання методів заміни вхідних змінних і ПЕС породжує GB-структуру СМПА, подану на рис. 4. При цьому має значення виконання умов $R_1 < R$ і $G < L$.

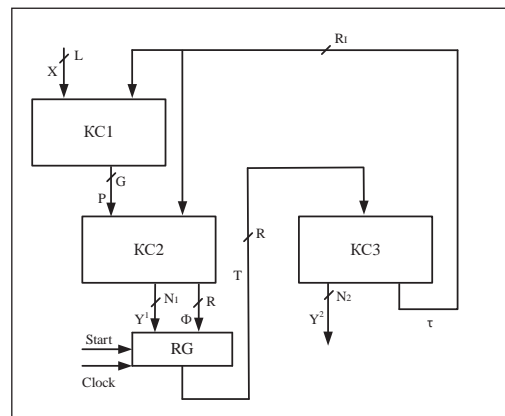


Рис. 4. GB-структура СМПА

Використання такого перетворення структурної схеми СМПА забезпечує суттєве зменшення розрядності вхідного коду, а також зменшення числа термінів у системі функцій збудження

пам'яті внаслідок зменшення кількості строк («стискання») прямої структурної таблиці (ПСТ).

Класифікація структур СМПА за способом використання ресурсів мікросхеми та за способом оптимізації у базисі FPGA наведена на рис. 5.

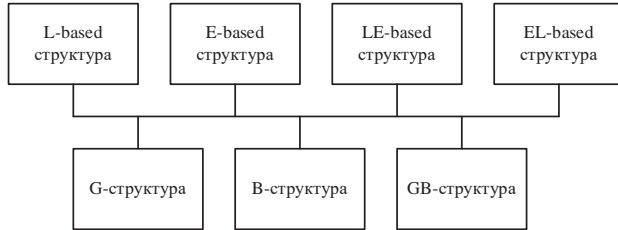


Рис. 5. Класифікація структур СМПА

Всі похідні структури пов'язані між собою загальними базисними, а вибір тієї чи іншої похідної структури виконується на умовах доцільності її використання.

Дослідження та висновки щодо вибору оптимізованої структури

На практиці автором було проведено дослідження структур суміщених керуючих автоматів у базисі FPGA. Практичне дослідження виконувалось для мікросхем фірм Xilinx, Altera і Microsemi. Як приклад було проведено порівняння L-based і LE-based моделей, як найбільш зручних у розробці. Дослідження проводились для СМПА з параметрами $X=101$, $Y^1=6$, $Y^2=16$.

Згідно з отриманими даними, для сімейств мікросхем фірми Altera при використанні LE-based структури система (7) реалізується на блоці EMB і кількість задіяних елементів LUT при цьому зменшується. Ефективність використання цієї структури залежить від співвідношення кількості функцій у системах Y_1 і Y_2 .

Дослідження методів оптимізації проводилися для L-based структури як найбільш місткої. Результати досліджень для мікросхем різних сімейств популярних фірм виробників, де вказана кількість елементів LUT при використанні різних структур СМПА, наведені на рис. 6–7.

Аналіз результатів доводить ефективність застосування даних методів і дозволяє оцінити ступінь редукації. На діаграмах рис. 6 і рис. 7 видно, що застосування методу заміни вхідних змінних (G-структура) дозволяє зменшити площу у середньому на 36%, 4% та 14% для мікросхем фірми Xilinx, Altera і Microsemi відповідно. Комплексне використання методів структурної редукації (GB-структура) дає зменшення займаної площі у середньому на 49%, 24% та 20% для мікросхем фірми Xilinx, Altera і Microsemi відповідно.

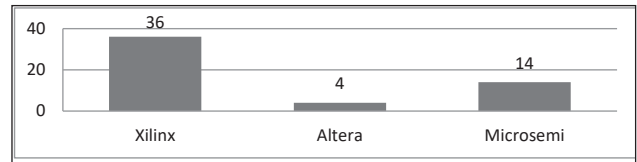


Рис. 6. Зменшення займаної площі FPGA при використанні G-структури на основі L-based моделі, %

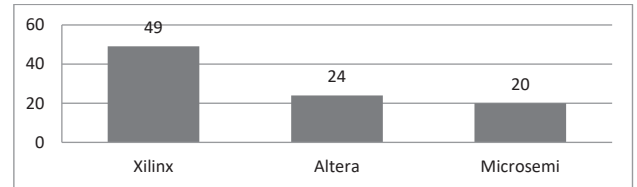


Рис. 7. Зменшення займаної площі FPGA при використанні GB-структури СМПА на основі L-based моделі, %

На підставі аналізу результатів досліджень можна зробити наступні висновки щодо доцільності використання тієї чи іншої структури СМПА.

L-based структуру доцільно використовувати для застарілих моделей FPGA, що не мають вбудованої пам'яті, або коли блоки вбудованої пам'яті використовуються для інших задач. Якщо у мікросхемі є вільні блоки EMB, то вибір L-based структури може бути зроблений при $2^{L+R}(N_1+R) \ll V_0$; $2^R N_2 \ll V_0$. Тобто і при реалізації KC1, і при реалізації KC2 на блоці EMBer (рис.1) могла би використовуватись лише незначна частина блоку, але його вже не можна було б використати для виконання інших задач, тому робиться вибір на користь L-based структури.

Перевагами L-based структури є бюджетність реалізації проекту у FPGA та можливість використовувати вільні блоки EMB для інших частин проекту.

Використання E-based структури має сенс, коли елементи LUT зайняті іншою частиною проекту, а блоки EMB не використовуються інакше.

Недоліками цієї структури є трудоємність програмної реалізації KC1 через велику таблицю, в якій необхідно вказати значення вихідних функцій при всіх можливих варіантах вхідних умов (максимальний розмір таблиці $S_{table}^{EMB} = 2^{L \cdot 2^R}$).

Для зменшення числа елементів у схемі СМПА використовується принцип гетерогенної реалізації, поданий EL-based і LE-based структурами.

EL-based структура має переваги, коли є незадіяні блоки EMB, що дозволяють звільняти елементи LUT для виконання інших задач. Така реалізація

лізація можлива при $2^{L+R}(N_1 + R) \leq V_0$, $t_f \geq N_1 + R$, а найбільшу ефективність має при умові $N_1 \gg N_2$, коли більша частина алгоритму виконується блоком ЕМВег, а менша розподілена по елементах LUT.

При програмній реалізації цієї структури розробник також стикається із необхідністю заповнення великої таблиці блоку ЕМВег. Якщо така реалізація все ж таки обрана, але порушується умова $2^{L+R}(N_1 + R) \leq V_0$, то є доцільним оптимізація структури методом заміни логічних умов при $S_A \geq G + R$ або методом ПЕС при $S_A \geq L + R_f$. Найбільшу ефективність при зменшенні вхідних сигналів блоку ЕМВег дає комплексне використання цих методів. Умовою його використання є вираз $S_A \geq G + R_f$.

Гетерогенна реалізація схеми СМПА у вигляді LE-based структури також можлива при наявності вільних блоків ЕМВ. Для її використання необхідне виконання умов $2^R N_2 < V_0$, $t_f \geq N_2$. Максимальна ефективність при використанні цієї структури досягається при $N_1 \ll N_2$. Завдяки тому, що

блок ЕМВег реалізує систему (7), таблиця, що описує вміст блоку, не містить опису вхідних сигналів автомата і значно скорочується порівняно з іншими структурами.

Висновки. При проектуванні схем суміщених мікропрограмних автоматів можна використовувати різні способи розташування частин логічної схеми автомата на внутрішніх ресурсах мікросхеми FPGA. Відповідно до обраного способу запропонована така класифікація структур СМПА: L-based, E-based, LE-based та EL-based структури – базові; G-, B- та GB-структури – оптимізовані за критерієм зменшення апаратних витрат.

Напрямок подальшої наукової роботи є вивчення та практичне дослідження методів структурної редукції, спрямованих на зменшення кількості вихідних сигналів схеми, а також дослідження ефективності застосування комбінації розглянутих методів із метою досягнення оптимальності реалізації схеми СМПА за критерієм мінімуму апаратних витрат.

Список літератури:

1. Baranov. S. Logic and System Desing of Digital Systems. Tallinn. 267 p.
2. Sklyarov V., Sklyarova I., Barkalov A., Titarenko L. Synthesis and Optimization of FPGA – based Systems. Berlin. 2014. 432 p.
3. Altera documentation. 2018. URL: www.altera.com/support/literature/lit-index.html (дата звернення: 20.01.2018).
4. Xilinx documentation. 2018. URL: <https://www.xilinx.com/products/silicon-devices.html> (дата звернення: 20.01.2018).
5. Баранов С. И. Синтез микропрограммных автоматов. Ленинград. 1979. 232 с.
6. Баркалов А. А., Титаренко Л. А., Зеленева И. Я. Реализация совмещенного микропрограммного автомата в базе FPGА. Наукові праці Донецького національного технічного університету. Серія: Інформатика, кібернетика та обчислювальна техніка. 2015. Вип. 2. С. 84 – 88.
7. Microsemi documentation. 2018. URL: <https://www.microsemi.com/products/fpga-soc/fpga-and-soc> (дата звернення: 20.01.2018).
8. Баркалов А.А., Зеленева И.Я., Грушко С.С. Оптимизация схемы совмещенного автомата в базе FPGА способом замены входных переменных. Науковий вісник Чернівецького національного університету імені Юрія Федьковича. Серія: Комп'ютерні системи та компоненти. Том 6. Випуск 2. 2015. С. 49 – 54.
9. Yang S. Logic Synthesis and optimization benchmarks user guide. Microelectronics Center of North Carolina. 1991. 43 p.
10. Баркалов А.А., Титаренко Л.А., Зеленева И.Я., Грушко С.С. Использование псевдоэквивалентных состояний в совмещенном микропрограммном автомате. Наукові праці Донецького національного технічного університету. Серія: Інформатика, кібернетика та обчислювальна техніка. 2017. № 1. С. 5 – 11.

КЛАССИФИКАЦИЯ СТРУКТУР СОВМЕЩЕННЫХ МИКРОПРОГРАММНЫХ АВТОМАТОВ ПРИ РЕАЛИЗАЦИИ В БАЗИСЕ FPGА

Проанализированы способы реализации логической схемы совмещенного микропрограммного автомата (СМПА) в базе FPGА с учетом особенностей внутренней структуры микросхем, а также соответствующие способы оптимизации схем. Предложена классификация структур СМПА по критерию способа использования ресурсов микросхемы для размещения отдельных блоков логической схемы. Приведены результаты исследований рассмотренных структур СМПА, на основании чего определены условия целесообразности использования для каждой структуры.

Ключевые слова: микропрограммный управляющий автомат, совмещенный микропрограммный автомат, LUT, встроенные блоки памяти, FPGА, ПЛИС.

**CLASSIFICATION OF COMBINED FSM STRUCTURES
FOR REALIZATION ON FPGA**

The methods of combined finite state machine (CFSM) logical scheme realization in FPGA basis as well as corresponding methods of circuit optimization are considered, taking into account features of the internal structure of microcircuits. The classification of CFSM structures according to the criterion of the chip resources using for the location of individual blocks of the logical scheme is proposed. The results of research of considered CFSM structures are presented, on the basis of which the conditions of use expediency are defined for each structure.

Key words: control unit, combined FSM, LUT, embedded memory, FPGA.