

КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

УДК 681.326

АНАЛІЗ КОРЕКТНОСТІ ГРАФІВ ПЕРЕХОДІВ КЕРУЮЧИХ АВТОМАТІВ ПРИ ПОБУДОВІ HDL-МОДЕЛЕЙ ДЛЯ АВТОМАТИЗОВАНОГО СИНТЕЗУ

ШКІЛЬ О.С., КУЛАК Е.М., ФІЛІППЕНКО І.В., ШАКУРА О.Г., ФОМЕНКО В.В.

Розглядається аналіз коректності умов переходів у графових моделях керуючих автоматів, представлених у вигляді HDL-моделей, з точки зору подальшого автоматизованого синтезу. Пропонується метод аналізу графа переходів, що дозволяє отримати ненадлишкову та коректну в запам'ятовуючій частині схемну реалізацію при синтезі HDL-моделі кінцевого автомата.

Ключові слова: кінцевий автомат, граф переходів, HDL-модель, автоматизований синтез, ортогональна булева функція.

Key words: finite state machine, state diagram, HDL-model, automatic synthesis, orthogonal Boolean function.

1. Вступ

Серед усієї множини керуючих пристроїв можна виділити пристрої логічного управління, у яких управляючі сигнали (control value) представляються в двійковому алфавіті. Оскільки для реалізації керуючої частини в таких пристроях, як правило, використовуються кінцеві автомати, то ці автомати називаються керуючі автомати (КА). При описі алгоритму функціонування цифрових пристроїв логічного керування в САПР цифрових пристроїв одним із стилів написання коду на мовах опису апаратури (HDL-коду) є стиль автоматного програмування. Суть його полягає у відділенні опису логіки поведінки (за яких умов необхідно виконати ті чи інші дії) від опису його семантики (власне сенсу кожного з дій). Автоматні програми строго структуровані і в них виділено три види функцій: функції переходів, функції виходів та функції призначення нового стану. Автоматні програми також строго шаблонизовані з використанням операторів багатопозиційного вибору (switch, case) та умовних операторів (if, elsif-else) [1].

При аналізі надійності систем управління широко використовується поняття ортогональності, як системи несумісних подій [2]. Поняття ортогоналізації також використовується для декомпозиції логічних функцій при синтезі цифрових систем [3] та при перевірці графових моделей цифрових автоматів на коректність [4].

З точки зору реалізації алгоритму логічного управління структурний автомат це модель

пристрою реального часу, яка характеризується видачею визначених вихідних сигналів у визначені моменти часу. Структурний автомат функціонує у автоматному часі, який вимірюється в тактах $\{t, t+1, t+2\}$, тобто автомат переходить з одного стану в інший за один автоматний такт. За один автоматний такт автомат формує множину вихідних сигналів та обчислює значення наступного стану.

У аналітичному вигляді модель структурного автомату така $Y(t) = g(X(t), Z(t))$, $Z(t+1) = f(X(t), Z(t))$, де g – функція виходів структурного автомату, f – функція переходів структурного автомата. При цьому $Z(t+1) \equiv Z(t)$, але у наступному такті.

Таким чином, апаратна реалізація моделі структурного автомата (модель Хаффмена), складається з комбінаційного і послідовнісного компонентів. Послідовнісний компонент містить елементи пам'яті, такі як синхронні тригери, які запам'ятовують значення внутрішніх змінних (стан) і дозволяють змінювати його синхронно. Комбінаційний компонент складається з логічних елементів, які реалізують дві логічні функції: функцію виходів, яка обчислює значення вихідних сигналів, і функцію переходів, яка обчислює нові значення елементів пам'яті або внутрішніх змінних (тобто значення наступного стану) [5]. На рис.1 наведена графічна інтерпретація апаратної реалізації структурного автомата.

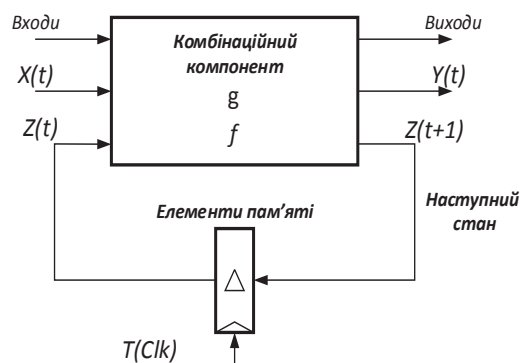


Рис. 1. Модель структурного автомата

Як алгоритмічна модель для керуючих автоматів застосовується такий непроцедурний візуальний формалізм, як графи переходів, які в англійській літературі називаються діаграмами станів (State Diagrams). Зазначимо, що графи переходів є не тільки візуальним відображенням алгоритму функціонування автомата, але й повною його математичною моделлю [4].

Вершинам графа переходів відповідають стани автомата a_i , дугам відповідають переходи між станами. На дугах автомата моделі Мілі вказуються функції умов переходів

$f_j(x_1, x_2, \dots, x_n)$ та вихідні (керуючі) сигнали. На дугах автомата моделі Мура вказуються тільки функції умов переходів, а вихідні сигнали вказуються біля вершин.

Предметом розгляду цієї роботи є аналіз коректної побудови функцій умов переходів графових моделей керуючих автоматів, представлених у вигляді HDL-моделей синтезованої підмножини, з точки зору подальшого автоматизованого синтезу.

2. Ортогоналізація логічних функцій умов переходів

Кожен граф переходів має бути семантично і синтаксично коректний. Перша властивість визначає, чи коректно побудована сама графова модель, а друга – чи немає формальних помилок при її побудові. При перевірці синтаксичної коректності граф переходів повинен перевірятися на досяжність, повноту, несуперечність і відсутність генеруючих контурів [4].

При перевірці синтаксичної коректності функцій умов переходів керуючого автомата вони повинні перевірятися на повноту та несуперечність.

На рис. 2 наведений фрагмент графа переходів з вершиною a_i , з якої виходять K дуг. Кожній дузі ставиться у відповідність логічний вираз функції умов переходів $f(x_1, x_2, \dots, x_n)$ у диз'юнктивній нормальній формі (ДНФ) :

$$f(x_1, x_2, \dots, x_n) = f_1(x_1, x_2, \dots, x_n) \vee \dots \vee f_j(x_1, x_2, \dots, x_n) \vee \dots \vee f_K(x_1, x_2, \dots, x_n).$$

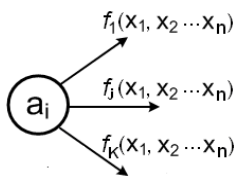


Рис. 2. Відображення функцій умов переходів у графовій моделі

Повнота перевіряється для кожної вершини графа переходів шляхом аналізу умов переходів всіх дуг, які виходять з цієї вершини, тобто

$$\bigvee_{j=1}^K f_j(x_1, x_2, \dots, x_n) = 1. \text{ Повнота умов визначається}$$

як покриття всіх 2^n термів сукупності булевих функцій переходів, де n – кількість умов переходів (вхідних змінних, які ініціюють переходи з цієї вершини), тобто реалізується функція $f(x_1, x_2, \dots, x_n) = 1$.

При забезпеченні несуперечливості для кожної вершини графа переходів перевіряється

ортогоналізація булевих виразів умов переходів (відсутність спільних термів у булевих виразах умов для різних дуг) функцій переходів для дуг, які виходять з вершини, що розглядається, тобто $\forall (f_g \cdot f_h = 0), g \neq h$ [5].

Диз'юнктивна нормальна функція алгебри логіки називається ортогональною, якщо всі її кон'юнктивні терми взаємно-ортогональні. Для такої ДНФ не існує набору значень її змінних, що входить більш ніж до однієї елементарної кон'юнкції, тобто на будь-якому наборі значень змінних значення одиниці може прийняти тільки одна кон'юнкція. Якщо логічну функцію представити у вигляді карти Карно, то їх зображення не будуть перетинатися. Прикладом ортогональної ДНФ може служити її досконала форма (ДДНФ, рос. мовою СДНФ), що складається з повних взаємно ортогональних кон'юнкцій [6].

Розглянемо метод побудови ортогональної повної системи функцій переходів для вершини графа переходів.

Введемо такі визначення: f – повна ДДНФ від n змінних, тобто $f(x_1, x_2, \dots, x_n) = 1$ – булева функція, яка приймає значення 1 на всіх 2^n наборах, f^* – повна ДДНФ від $(n-1)$ змінних, f^{**} – повна ДДНФ від $(n-2)$ змінних, f^{***} – повна ДДНФ від $(n-3)$ змінних і так далі. Таким чином, $f = f^* = f^{**} = f^{***} \equiv 1$.

Ортогональність термів булевої функції забезпечується за рахунок розкладання за відповідними змінними, з урахуванням повноти розкладання за всіма змінними [7].

За першою теоремою розкладання:

$$f = \overline{x_1} \cdot f^* \vee x_1 \cdot f^* = \overline{x_1} \cdot 1 \vee x_1 \cdot f^* = \overline{x_1} \vee x_1 \cdot f^* = \overline{x_1} \vee x_1 (\overline{x_2} \vee x_2 \cdot f^{**}) = \overline{x_1} \vee x_1 (\overline{x_2} \vee x_2 (\overline{x_3} \vee x_3 \cdot f^{***})) = \overline{x_1} \vee x_1 (\overline{x_2} \vee x_2 (\overline{x_3} \vee x_3 (\dots (\overline{x_n} \vee x_n))))).$$

Таким чином, повна булева функція від n змінних розкладається мінімум на $(n+1)$ кон'юнкцій при збереженні суттєвості всіх n змінних.

Як приклад розглянемо повну ДДНФ від трьох змінних $f(x_1, x_2, x_3) = 1$. За визначенням ДДНФ є ортогональною. Запишемо повну ДДНФ та виконаємо розкладання по x_1 з заміною лівої частини розкладання у дужках на 1:

$$f(x_1, x_2, x_3) = \overline{x_1} \overline{x_2} \overline{x_3} \vee \overline{x_1} \overline{x_2} x_3 \vee \overline{x_1} x_2 \overline{x_3} \vee \overline{x_1} x_2 x_3 \vee x_1 \overline{x_2} \overline{x_3} \vee x_1 \overline{x_2} x_3 \vee x_1 x_2 \overline{x_3} \vee x_1 x_2 x_3 = \overline{x_1} (\overline{x_2} \overline{x_3} \vee \overline{x_2} x_3 \vee x_2 \overline{x_3} \vee x_2 x_3) \vee x_1 (\overline{x_2} \overline{x_3} \vee \overline{x_2} x_3 \vee x_2 \overline{x_3} \vee x_2 x_3) = \overline{x_1} \cdot 1 \vee x_1 \cdot (\overline{x_2} \overline{x_3} \vee \overline{x_2} x_3 \vee x_2 \overline{x_3} \vee x_2 x_3) = \overline{x_1} \vee x_1 (\overline{x_2} \overline{x_3} \vee \overline{x_2} x_3 \vee x_2 \overline{x_3} \vee x_2 x_3).$$

Виконаємо аналогічно процедуру розкладання по x_2 для виразу у дужках. Розкриємо дужки та отримаємо повну ортогональну функцію від трьох змінних:

$$\begin{aligned}
 f(x_1, x_2, x_3) &= \bar{x}_1 \vee x_1 (\bar{x}_2 (x_3 \vee \bar{x}_3) \vee x_2 (\bar{x}_3 \vee x_3)) = \\
 &= \bar{x}_1 \vee x_1 (\bar{x}_2 \cdot 1 \vee x_2 (\bar{x}_3 \vee x_3)) = \bar{x}_1 \vee x_1 (\bar{x}_2 \vee x_2 \bar{x}_3 \vee \\
 &\vee x_2 x_3) = \bar{x}_1 \vee x_1 \bar{x}_2 \vee x_1 x_2 \bar{x}_3 \vee x_1 x_2 x_3 .
 \end{aligned}$$

Виходячи з цього, можна зробити висновок, що повна ортогональна функція від n змінних має в своєму складі не менше, ніж $(n+1)$ кон'юнкцій. Отже, з кожного стану автомата з функцією умов переходів від n змінних $f(x_1, x_2, \dots, x_n)$ повинно бути не менше, ніж $(n+1)$ переходів.

Одним із способів візуального аналізу ортогональності функцій переходів є представлення ортогональних функцій на картах Карно. На рис. 3, а зображена карта Карно для ортогональної функції (1). З наведеної карти видно, що для ортогональної функції обведення груп одиниць для повної ДДНФ не перетинаються, тобто кон'юнкції не мають спільних частин.

На рис. 3,б представлена карта Карно для ортогональної функції $f(x_1, x_2, x_3) = \bar{x}_1 \vee x_1 \bar{x}_2 \vee x_1 x_2 \bar{x}_3$, але наведена функція не є повною, бо в ній відсутнє обведення $x_1 x_2 x_3$. Це обумовлено тим, що при побудові вказаної функції порушено правило її повноти, тобто $f^* \neq 1$.

На рис. 3,в також представлена карта Карно для ортогональної функції

$$f(x_1, x_2, x_3) = \bar{x}_1 \vee x_1 \bar{x}_2 \vee x_1 x_2,$$

але наведена функція теж не є повною, бо в ній відсутня змінна x_3 . Це обумовлено тим, що при побудові вказаної функції порушено правило наявності у повній функції не менше, ніж $(n+1)$ кон'юнкцій.

| | | | | | |
|-------|--|-----------|-----|-----|-----|
| | | $x_2 x_3$ | | | |
| x_1 | | 00 | 01 | 11 | 10 |
| 0 | | (1) | (1) | (1) | (1) |
| 1 | | (1) | (1) | (1) | (1) |

а

| | | | | | |
|-------|--|-----------|-----|-----|-----|
| | | $x_2 x_3$ | | | |
| x_1 | | 00 | 01 | 11 | 10 |
| 0 | | (1) | (1) | (1) | (1) |
| 1 | | (1) | (1) | (1) | (1) |

б

| | | | | | |
|-------|--|-----------|-----|-----|-----|
| | | $x_2 x_3$ | | | |
| x_1 | | 00 | 01 | 11 | 10 |
| 0 | | (1) | (1) | (1) | (1) |
| 1 | | (1) | (1) | (1) | (1) |

в

Рис. 3. Карти Карно для ортогональних функцій

Ще одним способом візуального аналізу ортогоналізації повних булевих функцій є використання бінарних дерев рішень у формі ГСА (рис. 4).

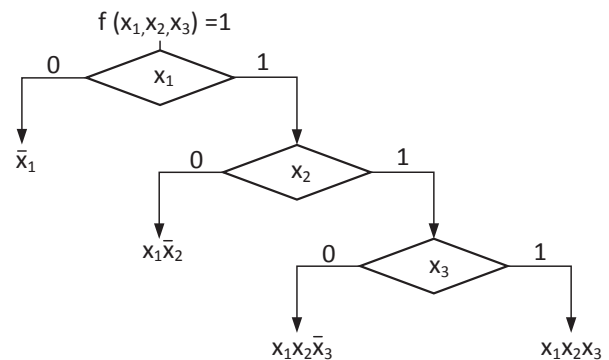


Рис. 4. Дерево ортогональних рішень

Дерево ортогональних рішень на цьому рисунку відповідає карті Карно, зображеній на рис. 3,а. Тут 3 умовних вершини та 4 переходи (шляхи розгалуження), умови переходу по кожній з гілок відповідають термам ортогональної ДНФ (1).

При представленні функції умов переходів бінарним деревом ортогональних рішень можна точно визначити необхідне число переходів. При незалежності умов x_1, x_2, \dots, x_n одної від одної це число буде дорівнювати $(m + 1)$, де m – число вершин дерева (але не змінних, оскільки змінні у вершинах таких дерев можуть повторюватися).

3. Аналіз коректності HDL-моделей керуючих автоматів при автоматизованому синтезі

У цьому розділі наведена ілюстрація запропонованих теоретичних викладень на прикладах графових моделей автоматів, їх опису на мові VHDL, результатів їх моделювання та синтезу.

Розглянемо приклад графа переходів (рис. 5) з коректними умовами переходів із стану a_1 відповідно до дерева ортогональних рішень, наведеного на рис. 4. Функція умов переходів із стану a_1 ортогональна, отже, умови переходів несуперечливі. Крім того, функція є повною. VHDL-модель цього автомата наведена на рис. 6.

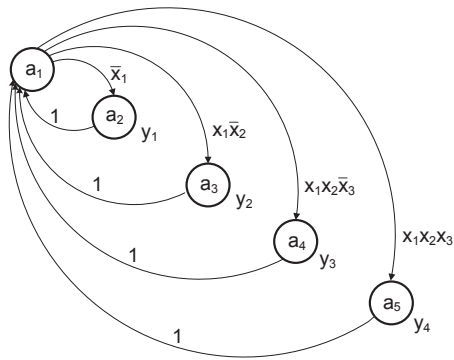


Рис. 5. Граф переходів автомата Мура з коректними умовами переходів

```
library IEEE;
use IEEE.std_logic_1164.all;
entity Fsm_right is
    port (x1, x2, x3, Clk, reset: in STD_LOGIC;
          y1, y2, y3, y4: out STD_LOGIC);
end;
```

```
architecture Fsm_right of Fsm_right is
    type State_type is (a1, a2, a3, a4, a5);
    signal State, NextState: State_type;
begin
    Sreg0_CurrentState: process (Clk, reset)
    begin
        if reset='1' then State <= a1;
        elsif Clk'event and Clk = '1'
        then State <= NextState;
        end if;
    end process;
    Sreg0_NextState: process (State, x1, x2, x3)
    begin
        case State is
        when a1=> if x1='0' then NextState <= a2;
        elsif x2='0' then NextState <= a3;
        elsif x3='0' then NextState <= a4;
        else NextState <= a5;
        end if;
        when a2=> NextState <= a1;
        when a3=> NextState <= a1;
        when a4=> NextState <= a1;
        when a5=> NextState <= a1;
        when others => NextState <= a1;
        end case;
    end process;
    y1 <= '1' when State=a2 else '0';
    y2 <= '1' when State=a3 else '0';
    y3 <= '1' when State=a4 else '0';
    y4 <= '1' when State=a5 else '0';
end;
```

Рис. 6. VHDL-модель автомата Мура з коректними умовами переходів

Часові діаграми роботи автомата наведені на рис. 7. На ньому відображені результати моделювання в системі Active-HDL ALDEC на усіх комбінаціях наборів умов x_1, x_2, x_3 . Діаграма свідчить, що функція умов переходів є повною та ортогональною і повністю повторює приклад з рис. 4. У період з 150 до 950 нс автомат переходить у стан a_2 ($y_1 = 1$), доки

виповнюється умова \bar{x}_1 , тобто ($x_1 = 0$), та повертається зворотно в a_1 ($y_1 = 0$). В період з 950 до 1350 нс автомат переходить в стан a_3 ($y_2 = 1$), доки виконується умова $x_1\bar{x}_2$, тобто ($x_1 = 1, x_2 = 0$), та повертається зворотно в a_1 ($y_2 = 0$). У період з 1350 до 1550 нс автомат переходить в стан a_4 ($y_3 = 1$), доки виконується умова $x_1x_2\bar{x}_3$, тобто ($x_1 = 1, x_2 = 1, x_3 = 0$), та повертається зворотно в a_1 ($y_3 = 0$). У період з 1550 до 1750 нс автомат переходить в стан a_5 ($y_4 = 1$), доки виконується умова $x_1x_2x_3$, тобто ($x_1 = 1, x_2 = 1, x_3 = 1$), та повертається назад в a_1 ($y_4 = 0$).

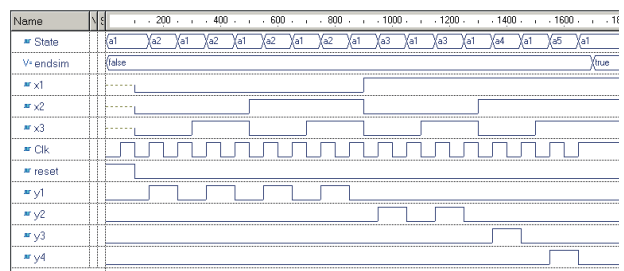


Рис. 7. Часові діаграми автомата Мура з коректними умовами переходів

Розглянемо наступний приклад графа переходів (рис. 8). У ньому умови переходів із стану a_1 некоректні з точки зору ортогоналізації функції умов переходів, при переході в a_4 і a_5 в записі термів бракує змінної x_1 , але вони не суперечать умовам переходів $x_1x_2\bar{x}_3$ та $x_1x_2x_3$. Функції умов переходів є повними.

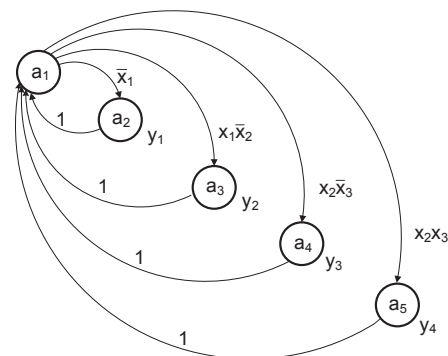


Рис. 8. Граф переходів автомата Мура з несуперечливими неповними умовами переходів

Переходи із стану a_1 можуть бути записані на мові VHDL так (рис.9, а або б). При цьому такий опис стилістично не є коректним, але не є суперечливим та при моделюванні дає такі ж результати, що і на рис. 7.

```

when a1=> if x1='0' then nextState <= a2;
          elsif x1='1' and x2='0' then nextState <= a3;
          elsif x2='1' and x3='0' then nextState <= a4;
          else nextState <= a5;
          end if;

```

а

```

when a1=> if x1='0' then nextState <= a2;
          elsif x1='1' and x2='0' then nextState <= a3;
          elsif x2='1' and x3='0' then nextState <= a4;
          elsif x2='1' and x3='1' then nextState <= a5;
          end if;

```

б

Рис. 9. Фрагменти VHDL- моделі автомата Мура з несуперечливими неповними умовами переходів

Крім того, моделі автоматів, що наведені на рис.6 та 9, при синтезі дають абсолютно однакові коректні результати. Синтез виконувався в системі XILINX ISE.

Далі розглянемо приклад графа переходів (рис. 10) з відсутнім переходом (по умові $x_1x_2x_3$) та неповною умовою переходу із стану a_1 в стан a_4 : $x_2\bar{x}_3$ замість $x_1x_2\bar{x}_3$. Функція умов переходів у даному випадку є неортогональною та неповною. Фрагмент VHDL- моделі цього автомата наведений на рис. 11.

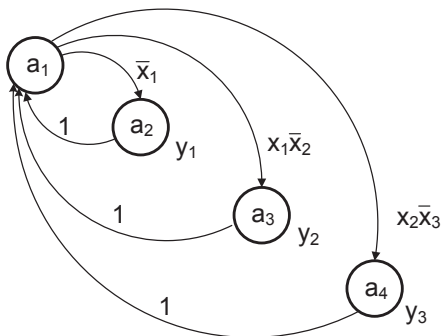


Рис. 10. Граф переходів автомата Мура з відсутнім переходом

```

when a1=> if x1='0' then nextState <= a2;
          elsif x1='1' and x2='0' then nextState <= a3;
          elsif x2='1' and x3='0' then nextState <= a4;
          end if;

```

Рис. 11. Фрагмент VHDL-моделі автомата Мура з відсутнім переходом

При моделюванні роботи даного автомата (рис. 12) на перший погляд все добре, але насправді змінна x_3 тут виявляється несуттєвою, як на рис. 3, в. На наборі $x_1, x_2, x_3=111$ автомат не має переходити в жодний стан, але система моделювання переводить його в стан a_4 . Так само при наборі умов $x_1, x_2, x_3=010$, автомат може перейти в два стани a_2 і a_4 , чого не повинно бути при правильній роботі автомата, але система

моделювання маскує таку ситуацію, переводячи автомат в стан a_2 . У цьому прикладі є і відсутність повноти, і наявність суперечності умов переходів, але на етапах аналізу синтаксису і моделювання це явно не проявляється.

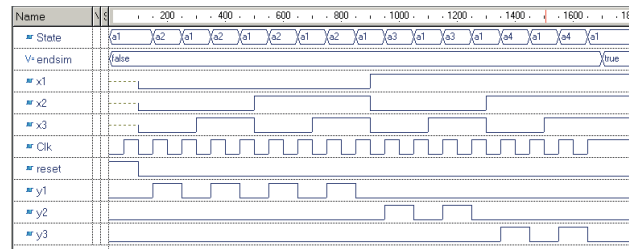


Рис. 12. Часові діаграми автомата Мура з відсутнім переходом

При синтезі цього автомата виникають проблеми. Критичної помилки не виникає, але з'являється попередження про появу чотирьох тригерів типу latch на додаток до чотирьох тригерів типу flip-flop: WARNING: Xst:737 – Found 4-bit latch for signal <NextState>. Latches may be generated from incomplete case or if statements. We do not recommend the use of latches in FPGA/CPLD designs, as they may lead to timing problems. Отже, замість двох тригерів для чотирьох станів синтезується 8 тригерів двох типів. Цього не повинно бути в коректно синтезованому автоматі.

На завершення розглянемо приклад графа переходів (рис. 13) з суперечливою умовою переходу із стану a_1 в стан a_4 : \bar{x}_2x_3 замість $x_1x_2\bar{x}_3$. Фрагмент VHDL-моделі цього автомата наведений на рис. 14.

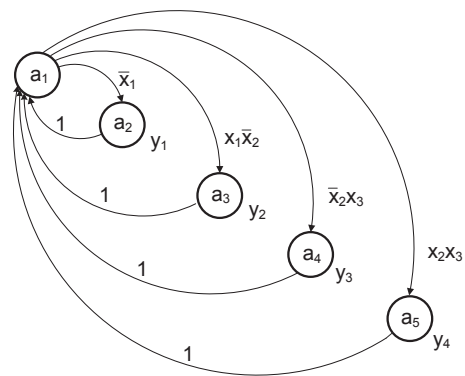


Рис. 13. Граф переходів автомата Мура з суперечливою умовою переходу

```

when a1=> if x1='0' then nextState <= a2;
          elsif x1='1' and x2='0' then nextState <= a3;
          elsif x2='0' and x3='1' then nextState <= a4;
          elsif x2='1' and x3='1' then nextState <= a5;
          end if;

```

Рис. 14. Фрагмент VHDL-моделі автомата Мура з суперечливою умовою переходу

При моделюванні роботи цього автомата (рис. 15) явно спостерігається тільки неможливість переходу в стан a_4 ні при яких наборах умов x_1, x_2, x_3 . Як і в попередньому прикладі, система моделювання маскує помилкову ситуацію, при якій створюються умови переходу автомата в декілька станів. Наприклад, при наборі умов $x_1, x_2, x_3 = 001$ автомат повинен перейти в два стани a_2 і a_4 , при наборі умов $x_1, x_2, x_3 = 101$ автомат повинен перейти в a_3 і a_4 . Також маскує помилкову ситуацію, при якій створюються умови, коли автомат не повинен переходити в жоден стан, наприклад, при наборі умов $x_1, x_2, x_3 = 110$. У даному прикладі також є відсутність повноти та наявність суперечності умов переходів, але на етапах аналізу синтаксису це ніяк не проявляється, а на етапі моделювання явно видно тільки те, що автомат ніколи не переходить в стан a_4 .

При синтезі теж виникають проблеми. З'являється попередження про появу п'яти тригерів типу latch на додаток до п'яти тригерів типу flip-flop: Xst:737 - Found 5-bit latch for signal <NextState>. Latches may be generated from incomplete case or if statements.

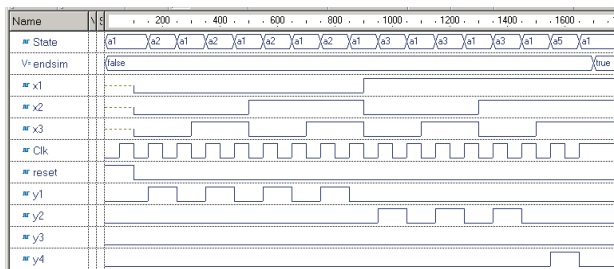


Рис. 15. Часові діаграми автомата Мура з суперечливою умовою переходу

Таким чином, показано, що проблеми, пов'язані з некоректними умовами, можуть проявитися тільки в процесі синтезу, тому перевірку умов переходів на несуперечність і повноту необхідно виконувати ще на етапі формування моделі графа переходів автомата.

4. Висновки

В результаті проведених досліджень було показано:

- 1) Функція умов переходів $f(x_1, x_2, \dots, x_n)$ несуперечлива, якщо вона ортогональна.
- 2) Ортогональна функція умов переходів $f(x_1, x_2, \dots, x_n)$ є повною, якщо її терми покривають усі набори x_1, x_2, \dots, x_n .
- 3) Число переходів із стану a_i з функцією умов переходів $f(x_1, x_2, \dots, x_n)$ має бути не менше $(n+1)$, де n – число змінних.

4) Число переходів із стану a_i з функцією умов переходів має бути рівне $(m+1)$, де m – число вершин дерева ортогональних рішень.

5) При невиконанні п. 1-4 висновків синтезується автомат з надмірним числом тригерів типу flip-flop та неприпустимою появою такого ж числа тригерів типу latch.

Наукова новизна полягає в тому, що запропоновано метод аналізу графа переходів кінцевого автомата на коректність умов переходів, що дозволяє отримати ненадмірну, коректну з точки зору запам'ятовуючої частини, схемну реалізацію при синтезі HDL-моделі кінцевого автомата.

Практична значущість: в силу особливостей роботи системи моделювання, відсутні переходи або суперечливі умови переходів на етапі перевірки синтаксису не фіксуються, на етапі моделювання вони можуть залишитися непоміченими і тільки після автоматизованого синтезу виявляються проблеми. Тому перевірка коректності умов переходів повинна перевірятися на етапі побудови моделі графа автомата. Це є актуальним в системах логічного управління, в порівнянні з пристроями мікропрограмного управління, де умови на графі переходів записуються по бінарному дереву умов у граф-схемі алгоритму (ГСА), що описує операційний автомат. У зв'язку з цим, ймовірність помилки значно менша в порівнянні з автоматами логічного управління, де ГСА не використовуються, як правило, а закон управління описується графом переходів.

Для аналізу функції умов переходів до 5 змінних можна використати карти Карно, але для більшого числа змінних і у разі автоматизації аналізу зручніше використати кубічне числення з представленням функцій умов переходів у вигляді багатовимірних векторів (кубів) [8].

Література: 1. Шкіль А.С. Автоматизированное проектирование систем логического управления с использованием шаблонов автоматного программирования / А.С. Шкіль, Э.Н. Кулак, И.В. Филиппенко, Д.Е. Кучеренко, М.В. Гога. // Радиоэлектроника та інформатика. 2018. №3. С. 75-81.

2. Рябинин И.А. Надежность и безопасность структурно-сложных систем / И.А. Рябинин. СПб.: Изд-во С.-Петербурга. ун-та, 2007. 276с. 3. Бибило П.Н. Синтез логических схем с использованием языка VHDL / П.Н. Бибило. М.: СОЛОН-Р, 2009. 384 с.: 4.

Шальто А.А. SWITCH-технология. Алгоритмизация и программирование задач логического управления / А.А. Шальто. СПб.: Наука, 1998. 628 с. 5. Баранов С.И. Синтез микропрограммных автоматов (граф схемы и автоматы) / С.И. Баранов. Л.: Энергия, 1979. – 232 с. 6. Закревский А.Д. Логические основы проектирования дискретных устройств / А.Д. Закревский, Ю.В. Поттосин, Л.Д. Черемисинова. М.: ФИЗМАТЛИТ, 2007. 592 с. 7. Шальто А.А. Новый

метод вычисления булевых формул / А.С.Ковалев, А.П.Лукиянова, А.А.Шалыто. [Электронный ресурс] Движение за открытую проектную документацию – Режим доступа: [www / URL: http://is.ifmo.ru/download/lbgmx.pdf](http://www.ifmo.ru/download/lbgmx.pdf). – Загл. с экрана. 8. Хаханов В.И. Техническая диагностика элементов и узлов персональных компьютеров / В.И. Хаханов – К.: ИЗМН, 1997. 308 с.

Transliterated bibliography:

1. *Shkil A.S.* Avtomatizirovannoe proektirovanie sistem logicheskogo upravlenija s ispol'zovaniem shablonov avtomatnogo programmirovanija / A.S. Shkil', Je.N. Kulak, I.V. Filippenko, D.E. Kucherenko, M.V. Goga. // Radioelektronika ta informatika. 2018. №3. S. 75-81.
2. *Rjabinin I.A.* Nadezhnost' i bezopasnost' strukturno-slozhnyh sistem / I.A. Rjabinin. SPb.: Izd-vo S.-Peterb. un-ta, 2007. 276s.
3. *Bibilo P.N.* Sintez logicheskikh shem s ispol'zovaniem jazyka VHDL / P.N. Bibilo. M.: SOLON-R, 2009. 384 s.:
4. *Shalyto A.A.* SWITCH-tehnologija. Algoritmizacija i programmirovanie zadach logicheskogo upravlenija / A.A. Shalyto. SPb.: Nauka, 1998. 628 s.
5. *Baranov S.I.* Sintez mikroprogrammnyh avtomatov (graf shemy i avtomaty) / S.I. Baranov. L.: Jenergija, 1979. 232 s.
6. *Zakrevskij A.D.* Logicheskie osnovy proektirovanija diskretnyh ustrojstv / A.D. Zakrevskij, Ju.V. Pottosin, L.D. Cheremisinova. – M.: FIZMATLIT, 2007. 592 s.
7. *Shalyto A.A.* Novyj metod vychislenija bulevykh formul / A.S. Kovalev, A.P. Luk'janova, A.A. Shalyto. [Jelektronnyj resurs] Dvizhenie za otkrytuju proektnuju dokumentaciju – Rezhim dostupa: [www / URL: http://is.ifmo.ru/download/lbgmx.pdf](http://www.ifmo.ru/download/lbgmx.pdf).
8. *Hahanov V.I.* Tehniceskaja divgnostika jelementov i uzlov personal'nyh komp'juterov / V.I. Hahanov. K.: IZMN, 1997. 308 s.

Поступила в редколлегию 11.03.2019

Рецензент: д-р техн. наук, проф. Кривуля Г.Ф.

Шкіль Олександр Сергійович, канд. техн. наук, доцент кафедри АПОТ ХНУРЕ. Наукові інтереси: діагностика цифрових систем, дистанційна освіта. Адреса: Україна, 61166, Харків, пр. Науки, 14, тел. 702-13-26.

Кулак Ельвіра Миколаївна, канд. техн. наук, доцент кафедри АПОТ ХНУРЕ. Наукові інтереси: автоматизоване проектування цифрових автоматів, тестопридатне проектування. Адреса: Україна, 61166, Харків, пр. Науки, 14, тел. 702-13-26.

Філіппенко Інна Вікторівна, канд. техн. наук, доцент кафедри АПОТ ХНУРЕ. Наукові інтереси: проектування цифрових пристроїв на базі мікроконтролерів, цифрові фільтри. Адреса: Україна, 61166, Харків, пр. Науки, 14, тел. 702-13-26.

Шакура Олексій Геннадійович, студент ХНУРЕ. Наукові інтереси: автоматизоване проектування цифрових автоматів. Адрес: Україна, 61166, Харків, пр. Науки, 14, тел. 702-13-26.

Фоменко Владислав Володимирович, студент ХНУРЕ. Наукові інтереси: верифікація моделей цифрових автоматів. Адреса: Україна, 61166, Харків, пр. Науки, 14, тел. 702-13-26.

Shkil Alexander Sergeevich, PhD, Associate Professor, Design Automation Department, KNURE. Scientific interests: diagnostics of digital systems, distance education. Address: Ukraine, 61166, Kharkiv, Nauka Avenue, 14, tel. 702-13-26.

Kulak Elvira Nikolaevna, PhD, Associate Professor, Design Automation Department, KNURE. Scientific interests: automated design of digital machines, HDL. Address: Ukraine, 61166, Kharkiv, Nauka Avenue, 14, tel. 702-13-26.

Filippenko Inna Victorovna, PhD, Associate Professor, Design Automation Department, KNURE. Scientific interests: design based on microcontrollers, digital filters. Address: Ukraine, 61166, Kharkiv, Nauka Avenue, 14, tel. 702-13-26.

Shakura Alexey Gennadievich, student, KNURE. Scientific interests: automated design of digital machines. Address: Ukraine, 61166, Kharkiv, Nauka Avenue, 14, tel. 702-13-26.

Fomenko Vladislav Vladimirovich, student, KNURE. Scientific interests: verification of FSM models. Address: Ukraine, 61166, Kharkiv, Nauka Avenue, 14, tel. 702-13-26.