

МІРОШНИК М. А., доктор техн. наук, професор, професор кафедри комп'ютерних систем та робототехніки, Харківський національний університет імені В. Н. Каразіна,

КУЛАК Е. М., канд. техн. наук, доцент, доцент кафедри автоматизації проектування обчислювальної техніки, Харківський національний університет радіоелектроніки,

КЛИМЕНКО Л. А., канд. техн. наук, доцент, доцент кафедри спеціалізованих комп'ютерних систем, Український державний університет залізничного транспорту,

МІРОШНИК А. М., PhD, ст. викл. кафедри автоматизації проектування обчислювальної техніки, Харківський національний університет радіоелектроніки



Метод аналізу помилок умов переходів станів у графових моделях кінцевих автоматів для систем автоматизованого синтезу

Анотація: Запропоновано формалізований метод, орієнтований на використання в компіляторах САПР, для виявлення помилок в умовах переходів у графових моделях керуючих автоматів. За допомогою цього підходу можна не лише виявляти стани, де умови переходів описані з помилками, але й точно визначати місце виникнення таких помилок. Метод має функціонувати на етапі компіляції початкової мовної моделі автомата, що дає змогу знаходити проблеми до синтезу схеми, тим самим знімаючи з проєктувальника потребу в ручному аналізі вже після отримання хибного результату автоматизованого синтезу.

Ключові слова: граф переходів, умови переходів, керуючий автомат, автоматизований синтез, HDL-модель, компілятор САПР.

Вступ

Область дослідження. Методи та моделі опису кінцевих автоматів для систем автоматизованого синтезу. HDL-моделі автоматів, компілятори САПР.

Аналіз наявних рішень

У сучасній практиці цифрового керування найчастіше використовують мікропрограмні автомати та автомати з часовим логічним керуванням [1-3]. Однією з актуальних проблем під час використання САПР (систем автоматизованого проектування) для синтезу керуючих автоматів є некоректне формування умов переходів станів [4].

Проблеми подібного характеру виявлено у відомих САПР, зокрема Xilinx ISE і Xilinx Vivado, на етапі автоматичного синтезу. Замість тригерів із динамічним керуванням, рекомендованих для використання в сучасних синхронних цифрових схемах, синтезують тригери зі статичним керуванням, застосування яких може призвести до появи часових збоїв у роботі пристрою.

Основною причиною таких ситуацій здебільшого є некоректно описані умови переходів у HDL-моделі автомата або повна відсутність деяких переходів між станами.

© МІРОШНИК М. А., КУЛАК Е. М., КЛИМЕНКО Л. А., МІРОШНИК А. М., 2025

Особливо це характерно для автоматів логічного керування, де граф переходів може бути сформований як первинне подання моделі, на відміну від мікропрограмних автоматів, у яких граф переходів формується формально після виконання розмітки граф-схеми операційного пристрою. У випадку мікропрограмних автоматів імовірність помилок в умовах переходів знижена, хоча й тут можливі втрати окремих переходів між станами керуючого автомата.

Однак на стадії синтаксичної перевірки компілятором такі помилки не виявлені. У результаті після некоректного синтезу виникає необхідність у ручному пошуку джерела помилки. Що ще суттєвіше, доводиться повторно виконувати синтез, що потребує значних витрат часу, особливо коли йдеться про великі моделі.

У роботі [4] подано результати дослідження впливу некоректно визначених умов переходів на якість синтезу керуючих автоматів. Було виявлено, що подібні помилки можуть спричинити появу небажаних тригерів із статичним керуванням замість очікуваних тригерів із динамічним керуванням.

Щоб уникнути подібних помилок на етапі проектування, необхідно впровадити метод перевірки правильності опису умов переходів ще під час побудови графової моделі або на етапі семантичного аналізу в рамках інструментарію САПР.

Граф переходів, що описує поведінку керуючого автомата, має бути і синтаксично, і семантично коректним. Під час перевірки синтаксису необхідно контролювати досяжність усіх станів, повноту опису, відсутність логічних суперечностей і замикальних контурів, що можуть спричинити нескінченні цикли [5].

Поняття ортогональності є важливим і активно використовують під час аналізу надійності керуючих систем, а також для декомпозиції логічних функцій у процесі синтезу цифрових схем [6, 7] і перевірки правильності графових моделей цифрових автоматів [5, 8].

У публікаціях [4, 9] було доведено, що якщо функція умов переходів зі стану є ортогональною, то вона є повною та несуперечливою. Це стало основою для ідеї створення інструменту, за допомогою якого компілятори САПР можуть виявляти потенційні логічні помилки в описі переходів ще до етапу схемного синтезу, забезпечуючи тим самим більшу надійність і ефективність автоматизованого проектування.

Мета дослідження

Метою дослідження є створення методу виявлення помилок у логічних умовах переходів між

станами в графових моделях керуючих автоматів на основі аналізу ортогональності функції умов, що відповідають переходам зі стану, який розглядають.

Постановка задачі

Об'єкт дослідження — графові подання керуючих автоматів. Предмет дослідження — вивчення формальних характеристик помилок, що виникають в умовах переходів у графовій структурі автомата з використанням таблиці термів ортогональної логічної функції та вектора помилок.

Завданням дослідження є розроблення методу побудови моделі для формалізованого опису правильних умов переходів; дослідження всіх можливих видів помилок, що виникають у ортогональних логічних функціях; вивчення формальних характеристик помилок в умовах переходів у графових структурах керуючих автоматів; виявлення закономірностей у типових помилках і їх упорядкування; створення алгоритму для автоматизованого виявлення помилок; проведення практичного тестування запропонованого методу на прикладних задачах.

Основний матеріал

Спершу розглянемо суть ортогоналізації логічних функцій, що визначають умови переходів.

Кожна графова модель переходів має відповідати як синтаксичним, так і семантичним вимогам. У процесі перевірки синтаксичної коректності графа переходів необхідно враховувати такі властивості, як досяжність, повнота, відсутність суперечностей, а також відсутність циклів, що породжують нескінченні переходи [7].

У випадку перевірки логічних функцій, які задають умови переходів у керуючому автоматі, необхідно переконатися в їхній повноті й логічній узгодженості (несуперечності).

На рис. 1 зображено фрагмент графа переходів, де вершина a_i має K вихідних дуг.

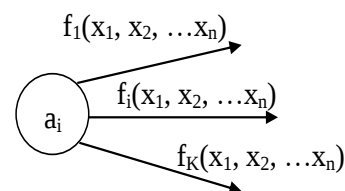


Рис. 1. Графічне відображення умов переходів у моделі керуючого автомата

Для кожної дуги задано логічний вираз, що описує умову переходу $f(x_1, x_2, \dots, x_n)$ у диз'юнктивній нормальній формі (ДНФ):

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

Повноту перевіряють для кожної його вершини шляхом дослідження логічних умов переходів, що відповідають усім дугам, які виходять з цієї вершини:

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

Повноту умов переходів визначають як охоплення всієї множини з 2^n термів, що відповідають булевим функціям переходів, де n — кількість вхідних змінних, які зумовлюють переходи з аналізованого стану. Тобто має бути виконана тотожність $f(x_1, x_2, \dots, x_n) = 1$. Несуперечливість забезпечена перевіркою ортогональності булевих виразів, які описують умови переходів із кожної вершини графа: булеві функції, що відповідають різним дугам, не повинні мати спільних термів, тобто для кожної пари функцій $\forall (f_g \cdot f_h = 0), g \neq h$ [2].

Диз'юнктивну нормальну форму (ДНФ) булевої функції вважають ортогональною, якщо її кон'юнктивні терми є взаємно ортогональними, тобто кожен конкретний набір значень вхідних змінних задовольняє лише одну елементарну кон'юнкцію. У термінах карти Карно це означає, що області, що відповідають істинності кон'юнкцій, не перетинаються. Прикладом такої ортогональної структури є досконала диз'юнктивна нормальна форма (ДДНФ), яка складається з повної множини взаємно ортогональних кон'юнктивних термів [4].

Додатковим інструментом для візуального аналізу ортогональності повних булевих функцій є подання у вигляді бінарних дерев рішень у формі графа схеми алгоритму (ГСА), як це показано на рис. 2.

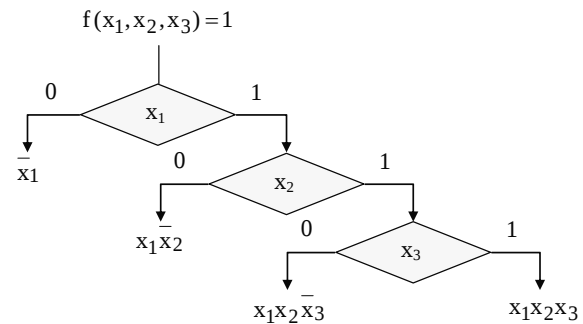


Рис. 2. Умови переходів для f ГСА

Коли функції умов переходів подано у вигляді бінарного дерева ортогональних рішень, існує можливість точно встановити необхідну кількість переходів. Якщо умови $x_1, x_2, \dots, x_n$ є незалежними між собою, то кількість таких переходів дорівнює $(m + 1)$, де m означає кількість умовних вершин дерева. При цьому слід зазначити, що вершини — це не змінні, оскільки одна й та сама змінна може фігурувати в декількох вершинах дерева.

Кількість переходів зі стану a_i , яка відповідає функції умов переходів, має бути більшою або дорівнювати $(n+1)$, де n — це кількість незалежних змінних.

На рис. 2 зображено дерево ортогональних рішень із лінійною структурою розгалуження, у якому відсутні повторювані умови (оскільки повторення не має додаткового смислового навантаження). Це дерево має три умовні вершини та чотири шляхи розгалуження, кожен із яких відповідає окремому терму ортогональної диз'юнктивної нормальної форми.

На рис. 3 подано граф переходів для функції f , побудований на основі ортогональних рішень. У подальшому для наочності та опису переходів використовуватимемо умовні позначення станів $a_1, a_2, \dots, a_K$ [4].

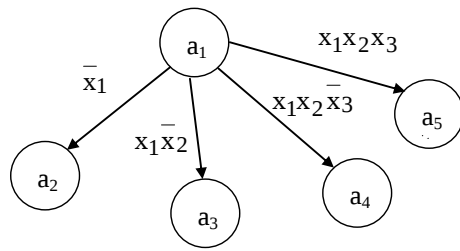


Рис. 3. Фрагмент графа переходів з умовами для f

Формула нижче відповідає функції f без помилок у виразі формули умов переходів ортогональної ДНФ:

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

У табл. 1 наведено подання функції f у вигляді окремих термів (стовпців).

Таблиця 1

Таблиця для функції f

№	$x_1 x_2 x_3$	$\overline{x_1}$	$x_1 \overline{x_2}$	$x_1 x_2 \overline{x_3}$	$x_1 x_2 x_3$	err	
0	000	1	-	-	-	1	-
1	001	1	-	-	-	1	-
2	010	1	-	-	-	1	-
3	011	1	-	-	-	1	-
4	100	-	1	-	-	1	-
5	101	-	1	-	-	1	-
6	110	-	-	1	-	1	-
7	111	-	-	-	1	1	-

Таблиця, призначена для оцінювання повноти і несуперечності умов переходів, включає множину змінних, набір термів ортогональної функції переходів, а також вектор помилок. Для кожного

терму обчислюють відповідні значення на всіх можливих наборах змінних функцій умов переходів.

Вектор помилок, позначений як err, подано у вигляді стовпця, де кожен елемент вказує кількість одиничних значень у рядку, що відповідає певному набору змінних. У табл. 1 цей вектор складається виключно з одиниць, що свідчить про відповідність функції f ортогональній диз'юнктивній нормальній формі (ДНФ) умов переходів.

На початковому етапі доцільно класифікувати всі можливі помилки за певними категоріями:

- відсутність терму або окремої змінної в термі;
- некоректна інверсія змінної або всього терму;
- помилкове використання або заміна змінних місцями.

Розглянемо випадок втрати терму.

Якщо був втрачений терм $\overline{x_1}$ у функції f, вона матиме такий вигляд:
 $f_1(x_1, x_2, x_3) = x_1 \overline{x_2} \vee x_1 x_2 \overline{x_3} \vee x_1 x_2 x_3$. Ця функція зображена в табл. 2.

Таблиця 2 Т

Таблиця для функції f₁

№	$x_1 x_2 x_3$	$x_1 \overline{x_2}$	$x_1 x_2 \overline{x_3}$	$x_1 x_2 x_3$	err	
0	000	-	-	-	0	+
1	001	-	-	-	0	+
2	010	-	-	-	0	+
3	011	-	-	-	0	+
4	100	1	-	-	1	-
5	101	1	-	-	1	-
6	110	-	1	-	1	-
7	111	-	-	1	1	-

У табл. 2 можна побачити, що в стовпці помилки err зникли одиниці, натомість з'явилися нулі. В останньому стовпці знаками «+» виділені ці рядки. Можна також побачити закономірність, що ці нулі

збігаються з нулями у другому стовпці таблиці, і їм відповідає змінна x_1 функції f_1 .

Ми тут опускаємо безліч прикладів, використаних для дослідження, але слід зазначити, що з втратою терму з'являються нулі в егг, і в проекції цих нулів на другий стовпець можна визначити формулу терму. Якщо в проекції змінна x_1 не змінює своє значення і дорівнює 0, це буде $\overline{x_1}$, якщо дорівнює 1, то буде x_1 , а якщо змінює значення, ця змінна не увійде в терм.

Далі розглянемо втрату змінних у термах.

Якщо були втрачені у функції f (у термі $x_1 x_2 x_3$) змінні $x_1 x_2$, вона матиме такий вигляд: $f_2(x_1 x_2 x_3) = \overline{x_1} \vee x_1 \overline{x_2} \vee x_1 x_2 \overline{x_3} \vee x_3$. Ця функція зображена в табл. 3.

Таблиця 3

Таблиця для функції f_2

№	$x_1 x_2 x_3$	$\overline{x_1}$	$x_1 \overline{x_2}$	$x_1 x_2 \overline{x_3}$	x_3	егг	
0	000	1	-	-	-	1	-
1	001	1	-	-	1	2	+
2	010	1	-	-	-	1	-
3	011	1	-	-	1	2	+
4	100	-	1	-	-	1	-
5	101	-	1	-	1	2	+
6	110	-	-	1	-	1	-
7	111	-	-	-	1	1	-

У таблиці з'являються двійки в стовпчику егг. Втрата змінної розширює простір, який займає терм у карті Карно і, як наслідок, у таблиці повноти. Займаний простір завжди збільшується вдвічі, крім випадків, коли помилка виникає в термі максимальної довжини. У табл. 3 видно, що кількість рядків із значенням 2 у егг. непарна, тому що помилка тут у термі максимальної довжини.

Проведений аналіз виявлених помилок дає змогу встановити, що значення елементів вектора помилки егг у рядках, де зафіксовано некоректності, дорівнюють або 0, або 2, однак ці значення не

співіснують одночасно в межах одного рядка. Зокрема, у випадках, коли у складі терму відсутня змінна, відповідна область, яку він охоплює на карті Карно, подвоюється, що й обумовлює появу значення 2 у векторі помилок. Водночас значення 0 у векторі егг свідчить про повну відсутність відповідного терму у функціональному описі переходів. Оскільки площа, яку займає терм на карті Карно, визначена кількістю змінних у логічній функції, то, маючи інформацію про кількість змінних, виявлених помилок і значення вектора помилок, можна зробити аналітичні вирази для визначення розміру терму, у якому допущено помилку.

Далі розглянемо інверсію терму.

Якщо була зроблена помилкова інверсія терму $\overline{x_1}$, функція матиме такий вигляд:

$f_3(x_1 x_2 x_3) = x_1 \vee x_1 \overline{x_2} \vee x_1 x_2 \overline{x_3} \vee x_1 x_2 x_3$. Ця функція зображена в табл. 4.

Т

Таблиця 4

Таблиця для функції f_3

№	$x_1 x_2 x_3$	$\overline{x_1}$	$x_1 \overline{x_2}$	$x_1 x_2 \overline{x_3}$	x_3	егг	
0	000	1	-	-	-	1	-
1	001	1	-	-	1	2	+
2	010	1	-	-	-	1	-
3	011	1	-	-	1	2	+
4	100	-	1	-	-	1	-
5	101	-	1	-	1	2	+
6	110	-	-	1	-	1	-
7	111	-	-	-	1	1	-

Аналізуючи дані, наведені в табл. 4, можна встановити, що у всіх восьми рядках вектор помилок егг має значення, відмінне від одиниці. При цьому у векторі зафіксовані як нульові, так і двійкові значення, причому їхня кількість є однаковою. Додатково, проаналізувавши координати, виявлено, що кожному рядку з нульовим значенням у векторі помилок відповідає рядок із двійковим значенням, для якого координата x_1 має протилежне значення.

Далі розглянемо переплутування змінних.

Якщо була зроблена помилкова заміна змінних x_1 на x_2 , x_2 на x_3 , x_3 на x_1 , функція матиме такий вигляд: $f_4(x_1x_2x_3) = \overline{x_2} \vee x_2 \overline{x_3} \vee x_2x_3\overline{x_1} \vee x_2x_3x_1$. Ця функція зображена в табл. 5.

Таблиця 5

Таблиця для функції f_4

№	$x_1x_2x_3$	$\overline{x_2}$	$x_2\overline{x_3}$	$x_2x_3\overline{x_1}$	$x_2x_3x_1$	егг	
0	000	1	-	-	-	1	-
1	001	1	-	-	-	1	-
2	010	-	1	-	-	1	-
3	011	-	-	1	-	1	-
4	100	1	-	-	-	1	-
5	101	1	-	-	-	1	-
6	110	-	1	-	-	1	-
7	111	-	-	-	1	1	-

У табл. 5 помилок не видно, адже вектор егг має одиничні значення у всіх рядках. Це трапилося тому, що всі терми мають відмінний вигляд від початкових термів. Слід зазначити, що переплутування змінних – це дуже складний клас помилок, і в деяких виразах виявити помилку неможливо.

У наступному прикладі була зроблена помилкова заміна змінної x_1 на x_2 у термі $\overline{x_1}$. У цьому випадку функція матиме такий вигляд: $f_5(x_1x_2x_3) = \overline{x_2} \vee x_1 \overline{x_2} \vee x_1x_2\overline{x_3} \vee x_1x_2x_3$. Ця функція зображена в табл. 6.

Таблиця 6

Таблиця для функції f_5

№	$x_1x_2x_3$	$\overline{x_2}$	$x_1\overline{x_2}$	$x_1x_2\overline{x_3}$	$x_1x_2x_3$	егг	
0	000	1	-	-	-	1	-
1	001	1	-	-	-	1	-
2	010	-	-	-	-	0	+
3	011	-	-	-	-	0	+
4	100	1	1	-	-	2	+
5	101	1	1	-	-	2	+
6	110	-	-	1	-	1	-
7	111	-	-	-	1	1	-

Проаналізувавши дані, наведені в табл. 6, встановлено, що, як і у випадках помилок, пов'язаних із втратою інверсії, кількість нульових значень у векторі помилок дорівнює кількості двійкових значень.

Розглядаючи окрему помилку, можна зіставити відповідні пари координат із нульовими та двійковими значеннями: 010 – 100, 011 – 101. Аналіз показує, що кожна така пара відрізняється за двома координатами, тоді як для одиничної помилки, пов'язаної лише із втратою інверсії, відмінність зафіксовано за однією координатою.

З метою спрощення процесу виявлення помилок запропоновано сформулювати метод, який передбачає послідовне виконання таких кроків.

Крок 1. Аналіз вектора помилок.

За наявності однієї помилки у функції вектор помилок може містити значення 0, 1 або 2. При цьому рядок зі значенням 1 свідчить про те, що у відповідній комбінації змінних помилка себе не проявляє. Отже, у векторі помилок можливі такі ситуації:

1) усі значення у векторі егг дорівнюють 1. Такий випадок вказує на відсутність виявлених помилок. Проте, як показано на прикладі функції f_4 , зі взаємною заміною змінних значення вектора помилок можуть залишатися рівними 1 у всіх рядках, незважаючи на наявність помилки. Це пояснюють тим, що утворюється нова функція, яка формально, за умовами, описана коректно. У межах цієї роботи випадки маскованих помилок не розглянуто, тому значення 1 у векторі помилок будемо трактувати як відсутність виявленої помилки для відповідного набору змінних;

2) усі значення у векторі егт дорівнюють 0, без урахування рядків із 1. Це свідчить про повне пропущення відповідного терму. Як показано на прикладі функції f_1 , у всіх випадках вектор помилок набував значення 0 у рядках, де була виявлена помилка, що відповідало повній відсутності терму у функції.

Далі для аналізу слід перейти до кроку 2 випадку 1;

3) усі значення у векторі егт дорівнюють 2, за винятком рядків зі значенням 1. Такий випадок свідчить про збільшення області покриття відповідного терму в таблиці аналізу повноти й несуперечності умов переходів. Це може відбуватися внаслідок пропуску змінної у складі терму. Аналогічні випадки були проаналізовані на прикладі функції f_2 . Для продовження аналізу необхідно перейти до кроку 2, випадку 1;

4) кількість рядків зі значенням 0 у векторі помилок дорівнює кількості рядків зі значенням «2». Таку ситуацію можна спостерігати на прикладі функції f_3 . За таких обставин неможливо однозначно визначити характер помилки без додаткового аналізу координат, у яких вона проявляється.

Слід зазначити, що наразі аналізують випадки, пов'язані з наявністю лише однієї помилки. До множинних помилок належать також випадки, де мала місце інверсія або перестановка змінних у межах одного терму, розмір якого перевищує одну змінну.

На наступному етапі необхідно попарно зіставити координати рядків із вектором помилок, що мають значення 0 та 2. Зазвичай відповідні пари рядків розташовані поряд у таблиці.

Коли існує некоректна інверсія змінних, координати рядків зі значенням 0 у векторі егт відрізняються від відповідних координат рядків зі значенням 2 лише за однією змінною. Натомість у випадках переплутування змінних ці відмінності охоплюють дві змінні. Відповідно на етапі 2 необхідно обрати правильний сценарій аналізу — випадок 2 або 3;

5) кількість рядків зі значенням 0 не дорівнює кількості рядків зі значенням 2. Подібні ситуації не були виявлені на розглянутих прикладах, однак їх наявність свідчить про існування декількох помилок різного характеру.

Крок 2. Аналіз кількості рядків із відмінними від 1 значеннями у векторі помилок.

На цьому етапі розглядають такі ситуації:

1) у випадку пропускання терму, маючи інформацію про кількість рядків, у яких вектор помилок має значення, відмінне від 1, можна визначити розмір пропущеного терму за допомогою формули

$$s = n - \log_2 R, \quad (1)$$

де R — кількість відмінних зі значенням, яке відрізняється від 1 у векторі егт;

n — загальна кількість змінних ($x_1, x_2 \dots x_n$);

s — довжина повністю пропущеного терму.

Після цього слід перейти до кроку 4 та розглядати випадок 1.

2) пропускання змінної у складі терму. Після аналізу кількості відмінних зі значенням 1 на прикладах функцій f_2 та f_1 можна встановити, що ця кількість може бути і парною, і непарною. Така ситуація вказує на довжину терму, у якому було допущено помилку.

Непарна кількість відмінних значень вказує на те, що змінна була пропущена в термі, який займав мінімальну площу в таблиці або на карті Карно, тобто одну клітинку або один рядок. У такому випадку початковий терм без помилок містив n змінних. Кількість пропущених змінних можна визначити за допомогою формули

$$a = \log_2 (R + 1), \quad (2)$$

де R — кількість відмінних зі значенням, яке відрізняється від 1 у векторі егт;

a — кількість змінних, пропущених у складі терму.

Парна кількість значень 2 у векторі егт ускладнює процес аналізу, оскільки стає неможливим безпосередньо визначити довжину терму, у якому допущено помилку. Проте розмір терму можна обчислити за допомогою формули

$$s = n - \log_2 (2^{(\log_2 R)+1} - R), \quad (3)$$

де R — кількість відмінних зі значенням, яке відрізняється від 1 у векторі егт;

n — загальна кількість змінних ($x_1, x_2 \dots x_n$);

s — довжина повністю пропущеного терму.

А кількість пропущених змінних можна обчислити за допомогою формули

$$a = \log_2 (2^{(\log_2 R)+1}) - \log_2 (2^{(\log_2 R)+1} - R), \quad (4)$$

де R — кількість відмінних зі значенням, яке відрізняється від 1 у векторі егт;

n — загальна кількість змінних ($x_1, x_2 \dots x_n$);

a — кількість змінних, пропущених у складі терму.

Необхідно враховувати, що у формулах (3) і (4) у виразі $\log_2 R$, який знаходиться в показниковій частині основи 2, для коректних розрахунків слід

відкидати всю дробову частину. Для продовження аналізу необхідно перейти до кроку 3 випадку 1;

3) у разі некоректної інверсії змінних, як це проілюстровано на прикладі f_2 , спостерігають, що обидва варіанти — інверсія окремих змінних або всіх змінних у термі — призводять до аналогічного результату: терм змінює своє положення на карті Карно та утворюється рівна кількість значень 0 та 2 у векторі егг.

Відповідно у формулі (1) необхідно поділити кількість значень, відмінних від 1, на 2, що дає змогу вивести формулу для визначення довжини терму, у якому була допущена помилка:

$$s = n - \log_2 \frac{R}{2}, \quad (5)$$

де R — кількість відмінних зі значенням, яке відрізняється від 1 у векторі егг;

n — загальна кількість змінних ($x_1, x_2 \dots x_n$);

s — довжина повністю пропущеного терму.

Після цього слід перейти до кроку 3 випадку 2;

4) якщо переплутані змінні. У випадку переплутування змінних функція переходів набуває іншого вигляду. За таких умов помилка не проявляється у всіх рядках, що потребує зменшення розміру терму на одиницю. Відповідно довжину терму у випадках переплутування змінних можна визначити за формулою

$$s = n - \log_2 \frac{R}{2} - 1, \quad (6)$$

де R — кількість відмінних зі значенням, яке відрізняється від 1 у векторі егг;

n — загальна кількість змінних ($x_1, x_2 \dots x_n$);

s — довжина повністю пропущеного терму.

Далі слід йти до кроку 3 випадку 2.

Крок 3. Аналіз рядків щодо існування термів із помилкою.

Маючи визначену довжину терму, у якому допущено помилку, можна проаналізувати терми, у яких потенційно могла виникнути помилка, і виключити ті варіанти, що не відповідають встановленим розмірним критеріям. Залежно від характеру виявленої помилки умови відбору термів дещо відрізнятимуться:

1) у разі пропускання змінної в термі, якщо кількість відмінних від 1 значень у векторі егг є непарною, розмір терму, де відбулась помилка, дорівнює n . Далі необхідно проаналізувати кожен із R рядків, у яких була зафіксована помилка, визначити відповідні змінні та методом виключення відкинути ті терми, розмір яких не відповідає умові $(n - a)$.

У випадку парної кількості змінних умова аналізу ускладнюється необхідністю окремого визначення розміру терму з помилкою. Отже, розмір аналізованих термів має дорівнювати $(s - a)$. Усі інші терми мають бути виключені з подальшого розгляду. Крім того, терм має бути присутнім у всіх рядках, де значення вектора помилок дорівнює 2;

2) у випадках некоректної інверсії та переплутування змінних умова аналізу є аналогічною: слід розглядати лише ті рядки, де значення вектора помилок дорівнює 2, і виключити всі терми, розмір яких не відповідає розрахованому на кроці 2 значенню s . При цьому варто враховувати, що, хоча підхід для визначення довжини терму у випадках некоректної інверсії та переплутування змінних схожий, відповідні формули розрахунку мають певні відмінності.

Крок 4. Аналіз позицій у рядках, де були знайдені помилки.

Розглянемо детально такі випадки:

1) якщо було втрачено цілий терм. У разі виявлення помилки, пов'язаної з втратою терму, слід проаналізувати всі координати рядків, для яких значення вектора егг дорівнює 0. Потрібно вибрати лише ті координати, символи яких залишилися незмінними, тобто не були модифіковані. Такий терм вважають втраченим. При цьому, якщо у відповідній координаті стоїть значення 1, змінну в термі слід записати без інверсії; якщо ж значення дорівнює -1 з інверсією;

2) якщо було втрачено змінну в термі. У цьому випадку аналізують координати рядків, де помилка не була виявлена, а також для термів, які не були відхилені на попередньому кроці (крок 3). Необхідно проаналізувати, у яких рядках зі значенням 1 вектора егг присутні ці терми, і попарно порівняти їхні координати з рядками, де вектор егг має відмінне від 1 значення. Якщо виявлено відмінність лише в одній змінній, можна зробити висновок про втрату саме цієї змінної. Після цього потрібно звернути увагу на значення цієї змінної в координатах рядків, для яких значення вектора егг дорівнює 1: якщо координата має значення 1, змінну в термі записують без інверсії, якщо 0 — з інверсією.

Однак можливі ситуації, коли на кроці 3 не вдається однозначно відкинути всі терми, окрім одного. У таких випадках неможливо остаточно встановити, у якому саме термі була допущена помилка. Розглянемо окремо такі випадки:

1) якщо з'явилася неправильна інверсія.

Порівнюючи попарно позиції рядків із відповідними значеннями вектора егг 2 та 0, можна виявити, у якій змінній сталася помилка. Далі, проаналізувавши значення цієї змінної в рядках із вектором егг 0, можна визначити правильне значення змінної: якщо координата дорівнює 1, змінну в термі слід записати без інверсії; якщо координата дорівнює 0 — з інверсією;

2) якщо змінні переплутані.

У випадках переплутування змінних у відповідних парах рядків зі значеннями вектора егг 2 та 0 відмінність спостерігатимуть у двох координатах. Для виявлення помилки слід проаналізувати, які рядки займають терми, що залишилися після кроку 3, і визначити спільну частину в координатах рядків із вектором егг 1 і рядків зі значенням 0. Ця спільна частина і буде відповідати терму без помилки.

Висновки

У результаті проведених досліджень створено модель, що описує коректні умови для переходів; досліджено формальні характеристики помилок в умовах переходів у графах переходів автоматів; упорядковано закономірності серед зафіксованих помилок; запропоновано метод виявлення помилок; проведено експериментальну перевірку запропонованого методу на конкретних прикладах.

Наукова новизна: запропоновано формалізований підхід для пошуку помилок в умовах переходів у графах переходів керуючих автоматів. За допомогою цього підходу можна не лише виявити стани з некоректно заданими умовами переходів, але й точно визначити місця виникнення таких помилок.

Практична значущість: за розробленим методом можна суттєво скоротити час на пошук причин некоректного автоматизованого синтезу автоматів у випадках некоректних умов переходів станів. Завдяки можливості автоматичного виявлення компілятором помилок HDL-моделі їх можна усунути до виконання синтезу, що знімає з проєктувальника потребу в трудомісткому ручному аналізі вже після отримання помилкового результату. Метод розрахований на використання в системах моделювання та САПР.

Список використаних джерел

1. Кривуля Г. Ф., Рябенський В. М., Рязанцев О. І. Комп'ютерна схемотехніка: підручник. Луганськ : СНУ ім. В. Даля, 2009. 744 с.
2. Борисенко О. А. Цифрова схемотехніка: підручник. Суми : Сумський державний університет, 2016. 200 с.
3. Volnei A. P. Finite state machines in hardware : theory and design (with VHDL and SystemVerilog). The MIT Press Cambridge, 2013. 349 p.
4. Аналіз коректності графів переходів керуючих автоматів при побудові hdl-моделей для автоматизованого синтезу / О. С. Шкіль, Е. М. Кулак, І. В. Філіппенко та ін. *Радиоелектроніка и інформатика*. 2019. № 1 [84]. С. 28-34.
5. Рудий Т. В., Паранчук Я. С., Сеник В. В. Алгоритмізація та програмування : навч. посіб. Ч. 1. Структурне програмування. Львів :

Львівський державний університет внутрішніх справ, 2023. 240 с.

6. Мірошник М. А., Курцев М. С. Автоматизація проєктування вбудованих систем і програмних засобів на ПЛІС мовою опису апаратури : навч. посіб. Харків : УкрДУЗТ, 2021. 332 с.
7. Мірошник М. А., Клименко Л. А., Корольова Я. Ю. Технології та автоматизація проєктування цифрових пристроїв складних комп'ютерних систем на ПЛІС: навч. посіб. Харків: УкрДУЗТ, 2021. 220 с.
8. Автоматизированное проектирование систем логического управления с использованием шаблонов автоматного программирования / А. С. Шкіль, Э. Н. Кулак, И. В. Филиппенко и др. *Радиоелектроніка та інформатика*. 2018. № 3. С. 75-81.
9. Кораблев Н. М., Кулак Э. Н., Шакура А. Г. Подсистема синтеза конечных микрокомандных автоматов. *Матеріали ІХ Міжнар. наук.-техн. конф. «Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління»*, Баку – Харків – Жиліна, 11-12 квітня 2019 р. 43 с.
10. Кулак Е. М., Погудін О. О., Кулак Г. К. Метод пошуку помилок в умовах переходів графових моделей керуючих автоматів. *Тези доповідей VIII Міжнар. наук.-техн. конф. «Проблеми інформатизації»*, Черкаси – Харків – Баку – Бельско-Бяла, 26-27 листопада 2020 р. С. 53.

Мірошник М., Кулак Е., Клименко Л. А., Мірошник А. Метод аналізу помилок умов переходів станів у графових моделях конечних автоматів для системи автоматизованого синтезу. У сучасній практиці цифрового керування найбільше використовують мікропрограми автоматів та автоматів з часовим логічним керуванням. Однією з актуальних проблем під час використання систем автоматизованого проєктування для синтезу керуючих автоматів є некоректне формування умов переходів станів. Проблеми подібного характеру виявляють у відомих САПР, зокрема Xilinx ISE і Xilinx Vivado, на етапі автоматичного синтезу. Замість тригерів із динамічним керуванням, рекомендованих для використання в сучасних синхронних цифрових схемах, синтезують тригери зі статичним керуванням, застосування яких може призвести до появи часових збоїв у роботі пристрою.

Основною причиною таких ситуацій здебільшого є некоректно описані умови переходів у HDL-моделі автомата або повна відсутність деяких переходів між станами. Особливо це характерно для автоматів логічного управління, де граф переходів формується як первинне подання моделі.

Однак на стадії синтаксичної перевірки компілятором такі помилки не виявлені. У результаті після некоректного синтезу виникає необхідність у ручному пошуку джерела помилки. Що ще суттєвіше,

доводиться повторно синтезувати, що потребує значних витрат часу, особливо коли йдеться про великі моделі.

Запропоновано формалізований метод, орієнтований на використання в компіляторах САПР, для виявлення помилок в умовах переходів у графових моделях керуючих автоматів. Цей підхід дає змогу не лише виявляти стани, де умови переходів описані з помилками, але й точно визначати місце виникнення таких помилок. Метод має функціонувати на етапі компіляції початкової мовної моделі автомата, що дає змогу знаходити проблеми до синтезу схеми, тим самим знімаючи з проєктувальника потребу в ручному аналізі вже після отримання хибного результату автоматизованого синтезу.

Ключові слова: граф переходів, умови переходів, керуючий автомат, автоматизований синтез, HDL-модель, компілятор САПР.

Miroshnyk M., Kulak E., Klymenko L. A., Miroshnyk A. Errors analysis method of state transitions conditions in FSM graph models for automated synthesis systems. In contemporary digital control practice, microprogrammed control units and finite state machines with temporal logic control are among the most widely used architectures. One of the pressing challenges in applying computer-aided design (CAD) systems for the synthesis of control automata is the incorrect specification of state transition conditions. Such issues are observed in well-known CAD environments, including Xilinx ISE and Xilinx Vivado, particularly at the stage of automatic synthesis. Instead of flip-flops with dynamic control – which are recommended for modern synchronous digital circuits—the tools tend to synthesize flip-flops with static control, whose application may lead to timing faults in device operation.

The primary cause of such errors typically lies in improperly defined transition conditions in the HDL (hardware description language) model of the automaton or in the complete omission of certain state transitions. This problem is especially common in logic-controlled automata, where the transition graph is constructed as the initial representation of the model.

However, such issues are not detected at the syntax-checking stage of compilation. As a result, incorrect synthesis necessitates manual error tracing. More critically, the synthesis process must be repeated, which significantly increases development time, particularly when dealing with large-scale models.

To address this issue, a formalized method is proposed for integration into CAD compilers, aimed at detecting errors in transition conditions within graph-

based models of control automata. This approach not only enables the identification of states with erroneous transition specifications but also provides precise localization of such errors. The method is designed to operate during the compilation stage of the initial language model of the automaton, thereby facilitating the identification of design flaws prior to circuit synthesis. This alleviates the designer's burden of manual debugging following an unsuccessful automated synthesis process.

Keywords: transition graph, transition conditions, control automaton, automated synthesis, HDL-model, CAD compiler.

Мірошник Марина, д-р техн. наук, професор, професор кафедри комп'ютерних систем та робототехніки, Харківський національний університет імені В. Н. Каразіна, пл. Свободи, 4, Харків, 61022.

Miroshnyk Maryna, Doctor of Technical Sciences, Professor, Professor computer systems and robotics department, V. N. Karazin Kharkiv National University, Svobody Sq., 4, Kharkiv, Ukraine, 61022. E-mail: m.miroshnyk@karazin.ua, <https://orcid.org/0000-0002-2231-2529>.

Кулак Ельвіра, канд. техн. наук, доцент, доцент кафедри автоматизації проєктування обчислювальної техніки, Харківський національний університет радіоелектроніки, пр. Науки, 14, Харків, Україна, 61166. E-mail: elvira.kulak@nure.ua, <https://orcid.org/0000-0002-8441-5187>.

Kulak Elvira, PhD, Associate Professor, Associate Professor of Design Automation Department, Kharkiv National University of Radio Electronics, Nauka Avenue, 14, Kharkiv, Ukraine, 61166. E-mail: elvira.kulak@nure.ua, <https://orcid.org/0000-0002-8441-5187>.

Клименко Любов, канд. техн. наук, доцент, доцент кафедри спеціалізованих комп'ютерних систем, Український державний університету залізничного транспорту, майдан Феєрбаха, 7, Харків.

Klymenko Liubov Candidate of Technical Sciences, Associate Professor, Associate Professor of the Department of Specialized Computer Systems of the Ukrainian State University of Railway Transport.

E-mail: klymenko.liubov73@gmail.com
<https://orcid.org/0000-0002-8252-7425>.

Мірошник Анатолій, канд. техн. наук, старш. викл. кафедри автоматизації проєктування обчислювальної техніки, Харківський національний

університет радіоелектроніки, пр. Науки, 14, Харків,
Україна, 61166. E-mail: anatolii.miroshnyk@nure.ua,
<https://orcid.org/0000-0001-5702-9611>.

Miroshnyk Anatolii. PhD, of Design Automation
Department, Kharkiv National University of Radio
Electronics, Nauka Avenue, 14, Kharkiv, Ukraine, 61166.
E-mail: anatolii.miroshnyk@nure.ua,
<https://orcid.org/0000-0001-5702-9611>.