

МЕТОДИ ВЕРИФІКАЦІЇ ТЕМПОРАЛЬНИХ ВЛАСТИВОСТЕЙ ЦИФРОВИХ АВТОМАТІВ
 ПШЕНИЧНИЙ К.Ю., ХАХАНОВА Г.В.

Для верифікації HDL-моделей пристроїв реального часу пропонується застосовувати механізм властивостей та асерцій мови System Verilog. Обґрунтовується доцільність механізму властивостей для більш вичерпного та ретельного тестування. Застосування цих підходів ілюструється конкретним прикладом.

Ключові слова: System Verilog, formal verification, functional verification, System Verilog assertions, assertion, property, functional coverage.

1. Вступ

Відомо, що пристрої логічного управління, побудовані на основі автоматного шаблону, функціонують в автоматному часі. Автоматним часом є дискретні відрізки часу, за які автомат переходить з одного стану в інший. Тривалість автоматного такту, як правило, визначається частотою синхросигналу. Проте, з іншого боку, пристрої логічного керування являють собою системи керування реального часу, і переходи між станами визначаються часовими параметрами алгоритму функціонування.

Для внесення реального часу в опис структурного автомату використовується розширена функція переходів (1), в якій аргументом виступає реальний час [3]:

$$Z(t + 1) = f(X(t), Z(t), T), \quad (1)$$

де $X(t)$ – функція входів; $Z(t)$ – стан автомата в поточному такті; T – поточний такт.

Граф, який описує функціонування автомата реального часу, називається темпоральним графом переходів (рис. 1). В такому графі кожна вершина має затримку T , впродовж якої автомат перебуває в даному стані. Затримку в кожній вершині темпорального графа реалізують за допомогою переходу в той же самий час впродовж певного часу. В ПЛІС це реалізується за допомогою лічильників, в мікроконтролерах – за допомогою таймерів з перериванням.

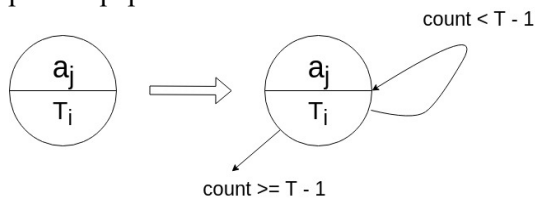


Рис. 1. Реалізація затримок в темпоральному графі автомата Мура

Мета дослідження – розробка та обґрунтування верифікаційних моделей на базі асерцій для HDL-моделей пристроїв реального часу.

Задачі:

- аналіз моделювання реального (метричного) часу на рівні HDL-опису;

- розробка асерційних моделей для верифікації темпоральних властивостей;
- порівняння розроблених моделей.

2. Два напрями верифікації

Вичерпним джерелом функціональних вимог є специфікація. На основі специфікації інженер створює тестові набори. Класичний підхід верифікації полягає у написанні тесту, моделюванні та аналізі результатів. У разі виявлення неправильної поведінки інженер аналізує часові діаграми та шукає потенційне місце помилки в HDL-описі. Головним недоліком такого підходу є час, який інженер має витратити на виявлення, пошук та виправлення несправності. Для значного зменшення цього часу існує механізм властивостей (property). Головна ідея даного підходу полягає у явному вираженні певних характеристик дизайну за допомогою спеціальних мов (SVA, PSL). Такі характеристики є правилами коректного функціонування пристрою. У разі їх порушення середовище моделювання видає відповідне повідомлення, що містить час та опис порушеної властивості.

3. Приклад автоматної моделі пристрою реального часу

Розглянемо модель пристрою для логічного керування дорожнім світлофором. На рис. 2 представлено граф переходів даного пристрою керування.

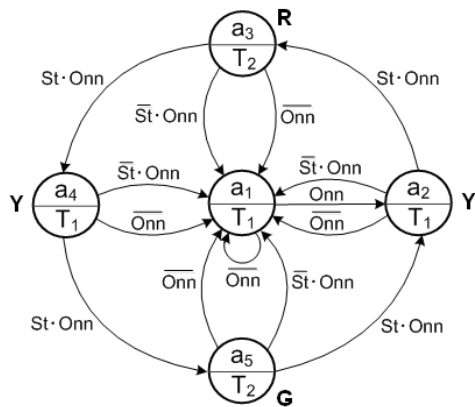


Рис. 2. Граф переходів автомата для керування дорожнім світлофором

Інтерфейс даного пристрою складається з двох множин сигналів:

- множина вхідних сигналів $\{Onn, St\}$, де $Onn \{0, 1\}$ – включення світлофора, $St \{0, 1\}$ – сигнал запуску стандартного циклу роботи;
- множина вихідних сигналів $\{R, Y, G\}$, де R – сигнал включення червоного світла, Y – сигнал включення жовтого світла, G – сигнал включення зеленого світла.

Пристрій має два цикли роботи:

- нічний (аварійний) – блимання жовтим кольором з інтервалом $T1$;

– стандартний цикл роботи {Y - R - Y - G -Y} із затримкою в станах R та G T2 тактів, а в стані Y – T1.

Головною особливістю HDL-реалізації автоматних моделей реального часу є наявність лічильників. Дані лічильники підраховують кількість тактів, впродовж яких автомат перебуває в певному стані. Для даної моделі T1 дорівнює 1 такту, T2 – 5 тактам (рис. 3).

```
reg [2:0] count, count1;
parameter T1 = 3'b010;
parameter T2 = 3'b101;
```

Рис. 3. Декларація лічильників та затримок на мові SystemVerilog

При переходах між станами необхідно враховувати значення лічильників. Дана перевірка виражає умову реального часу. На рис. 4 наведено приклад реалізації такого переходу між станами a2 та a3.

```
a2: begin
  if(count < T1 - 1) begin
    nextState = a2;
    count1 = count1 + 1'b1;
  end
  else if(!onn || !st) begin
    nextState = a1;
    count1 = 3'b000;
  end
  else
  begin
    nextState = a3;
    count1 = 3'b000;
  end
end
```

Рис. 4. Приклад реалізації переходів між двома станами з урахуванням значень лічильника

4. Верифікація темпоральних характеристик за допомогою миттєвих асерцій

Верифікація за допомогою миттєвих асерцій полягає у перевірці значення регістру стану впродовж певного часу затримки. Впродовж цього часу (певної кількості тактів) регістр має зберігати одне значення. На рис. 5 наведено приклад реалізації такої перевірки на мові SystemVerilog.



Рис. 5. Послідовність логічних подій для перевірки темпоральних властивостей автомата

Під час кожного переднього фронту синхросигналу необхідно перевірити, що автомат перебуває у бажаному стані та внутрішній лічильник перебуває у початковому стані. Це перевірка показує,

що автомат щойно перейшов в даний стан. Далі необхідно перевірити, що стан не змінюється впродовж N-1 тактів. Для цього використовується конструкція assert.

У разі порушення даної умови середовище моделювання видасть відповідне повідомлення.

Даний метод має такі недоліки:¶

- неявність вираження властивості дизайну;
- неможливість використання такого підходу з формальними середовищами верифікації;¶
- відсутність способу перевірки активізації верифікаційного коду (виконання умови конструкції if).

5. Верифікація темпоральних властивостей за допомогою паралельних асерцій

Бажану темпоральну властивість можна описати натуральною мовою так: якщо автомат перейшов зі стану a1 в стан a2, то він має перебувати у стані a2 N-1 такт та перейти у стан a3. Іншими словами, властивість складається з трьох логічних частин: передумова, цикл та наслідок.

На рис. 6 представлено три складові перевірки темпоральної властивості між станами a3 та a4 на мові SystemVerilog Assertions (SVA).

```
sequence Red2YellowPrecondition;
  ($past(state) == a2) && (state == a3);
endsequence
|
sequence Red2YellowStay;
  (state == a3) [*4];
endsequence

sequence Red2YellowPostCondition;
  state == a4;
endsequence
```

Рис. 6. Складові темпоральної властивості на мові SVA

Red2YellowPrecondition – послідовність, що описує передумову властивості. Вбудована функція \$past повертає значення регістру state на попередньому такті. Для даної властивості попереднім станом має бути стан a2.

Red2YellowStay – послідовність, що описує затримку у стані a3. Оператор [*4] вказує, що вираз (state == a3) має повертати логічне значення “1” впродовж 4 тактів.

Red2YellowPostCondition – послідовність, що описує наступний стан автомата після затримки.¶ Далі необхідно об’єднати описані вище послідовності у конструкцію property за допомогою спеціальних операторів. Приклад такої конструкції наведено на рис. 7.

```
property Red2Yellow;
  @(posedge clk) disable iff (!onn || !st || reset)
  Red2YellowPrecondition | =>
  Red2YellowStay ##1
  Red2YellowPostcondition;
endproperty
```

Рис 7. Конструкція property для перевірки темпоральних властивостей

Конструкція property Red2Yellow встановлює відношення між трьома послідовностями. Під час кожного переднього фронту синхросигналу *clk* перевіряється послідовність-передумова Red2YellowPrecondition. Якщо ця умова виконалася – під час кожного наступного переднього фронту синхросигналу відбувається перевірка затримки у стані a3 за допомогою послідовності Red2YellowStay. Останньою послідовністю у цій властивості є перехід у стан a4 – послідовність Red2YellowPostcondition.

На рис. 8 різними кольорами позначено момент обчислення певної послідовності, а саме:

червоним – Red2YellowPrecondition,

зеленим – Red2YellowStay,

синім – Red2YellowPostcondition.

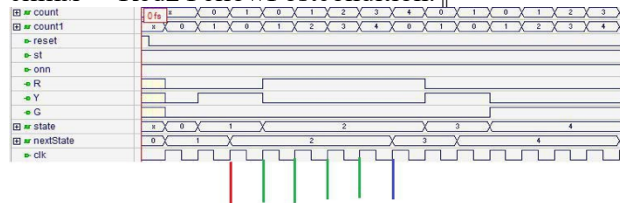


Рис. 8. Моменти обчислення трьох складових властивостей

Варто зауважити, що ця властивість описує стандартний цикл роботи автомата, тобто коли сигнали *om* та *st* мають логічне значення ‘1’. Окрім того, автомат у будь-який момент може бути встановлений у початкове положення за допомогою сигналу скидання *reset*. Усі ці умови виражаються за допомогою конструкції *disable iff*. Якщо виконується логічна умова у тілі *disable iff*, обчислення усіх послідовностей буде завершено.

Далі необхідно задати перевірку властивості. Оскільки властивість носить темпоральний характер, використовується механізм паралельних асерцій (*concurrent assertion*). Окрім ретельної верифікації, асерції можуть надавати вичерпну інформацію щодо покриття функціональності під час тестування. Для цієї задачі використовується механізм *cover property* (рис. 9), який надасть таку інформацію кількості:

- спроб вирахувати властивість;
- успішних проходжень властивості;
- невдалих проходжень властивості;
- “вакуумних успіхів” властивості (не було активовано послідовність Red2YellowPrecondition).

```
assert property (Red2Yellow);
cover property (Red2Yellow);
```

Рис. 9. Дефініція паралельної асерції

Механізм паралельних асерцій дає такі переваги:

- явне вираження властивості дизайну завдяки прив’язці до подій синхросигналу як засобу вираження часу;
- можливість використання як з динамічними, так і з формальними середовищами верифікації;

– можливість проаналізувати статистику спрацювань верифікаційного коду за допомогою інформації про покриття.

Висновки

Розглянуто та проведено дослідження методів верифікації темпоральних характеристик моделей пристроїв реального часу.

Запропоновано та обґрунтовано апарат темпоральних властивостей як засіб вираження часових вимог. Доведено, що використання механізму паралельних асерцій дає такі переваги: 1) зменшення часу локалізації помилок у HDL-описі; 2) можливість використання даного підходу як з динамічними, так і з формальними середовищами верифікації; 3) наявність механізму перевірки активізації верифікаційного коду.

Література:

1. Harry D. Foster, Adam C. Krohnik, David J. Lacey. Assertion-Based Design. New York: Springer US, 2005. 390 p.
2. Srikanth Vijayaraghavan, Meeyappan Ramanathan. A Practical Guide for SystemVerilog Assertions. New York: Springer US, 2005. 333 с.
3. Шкіль А.С., Кулак Э.Н., Филиппенко И.В., Кучеренко Д.Е., Гога М.В. Автоматизированное проектирование систем логического управления с использованием шаблонов автоматного программирования // Радиоэлектроника и информатика. 2018. №3 С. 75-81.

Transliterated Bibliography:

1. Harry D. Foster, Adam C. Krohnik, David J. Lacey. Assertion-Based Design. New York: Springer US, 2005. 390 p.
2. Srikanth Vijayaraghavan, Meeyappan Ramanathan. A Practical Guide for SystemVerilog Assertions. New York: Springer US, 2005. 333 p.
3. Shkil A.S., Kulak E.N., Filippenko I.V., Kucherenko D.E., Goga M.V. Avtomatizirovannoe proektirovanie sistem logicheskogo upravleniya s ispolzovaniem shablonov avtomatnogo programmirovaniya // Radioelectronics & Informatics. 2018. #3 P. 75-81.

Надійшла до редколегії 11.09.2019

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

Пшеничний Кирило Юрійович, магістрант кафедри АПОТ ХНУРЕ. Наукові інтереси: проектування та верифікація цифрових систем. Адреса: Україна, 61166, Харків, пр. Науки, 14. E-mail: kyrylo.pshenychnyi@nure.ua

Хаханова Ганна Володимирівна, канд. техн. наук, доцент кафедри АПОТ ХНУРЕ. Наукові інтереси: стиснення та відновлення двійкової інформації. Адреса: Україна, 61166, Харків, пр. Науки, 14. e-mail: anna.hahanova@nure.ua

Pshenichny Kirill Yuryevich, Master Student, Design Automation Department, NURE. Scientific Interests: design and verification of digital systems. Address: Ukraine, 61166, Kharkov, Nauky Ave, 14, e-mail: kyrylo.pshenychnyi@nure.ua

Khakhanova Anna Vladimirovna, Ph.D., Associate Professor, Design Automation Department, NURE. Scientific Interests: compressed and binary information recovery. Address: Ukraine, 61166, Kharkov, Nauky Ave., 14, e-mail: anna.hahanova@nure.ua