

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерної інженерії та управління _____
Кафедра _____ Автоматизації проектування обчислювальної техніки _____
Рівень вищої освіти _____ другий (магістерський) _____
Спеціальність _____ 123 – Комп'ютерна інженерія _____
Тип програми _____ Освітньо-професійна _____
Освітня програма _____ Спеціалізовані комп'ютерні системи _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« _____ » _____ 20 ____ р.

**ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ**

студентові Малахову Микиті Володимировичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи Автоматні моделі в системах управління реального часу на _____
на технологічній основі ПЛІС _____

затверджена наказом по університету від 04 листопада 2019 р. № 1624 Ст 2.
Термін подання студентом роботи до екзаменаційної комісії _____ 20__ р. 3.

Вихідні дані до роботи _____

САПР Xilinx ISE _____

Модель часового автомата _____

Системи управління реального часу _____

Програмовані логічні інтегральні схеми _____

4. Перелік питань, що потрібно опрацювати в роботі

_____ Аналіз моделей часових _____
автоматів _____

Аналіз моделі структурного часового автомата Мура _____

Аналіз апаратного підходу до реалізації систем реального часу _____

Розробка VHDL-моделей часового автомата Мура

Аналіз результатів синтезу та моделювання

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів)

21 слайд

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання	02.09.2019-08.09.2019	
2	Аналіз предметної області	09.09.2019-15.09.2019	
3	Аналіз джерел з проблемної галузі	16.09.2019-29.09.2019	
4	Розробка моделі структурного часового автомата Мура	30.09.2019-13.10.2019	
5	Аналіз апаратного підходу до реалізації систем реального часу	14.10.2019-20.10.2019	

6	Розробка VHDL-опису пристрою керування	21.10.2019-27.10.2019	
7	Моделювання розроблених VHDL-моделей	28.10.2019-03.11.2019	
8	Оформлення пояснювальної записки	04.11.2019-01.12.2019	
9	Оформлення графічного матеріалу	02.12.2019-08.12.2019	
10	Перевірка виконаного проекту керівником	09.12.2019-15.12.2019	

Дата видачі завдання

Студент _____

(підпис)

Керівник роботи _____

(підпис)

доц. Шкіль О.С. _____

(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка містить 86 сторінок, 64 рисунків, 15 джерел за переліком посилань.

ЧАСОВИЙ АВТОМАТ, СИСТЕМА КЕРУВАННЯ РЕАЛЬНОГО ЧАСУ, VHDL, ШАБЛОН, FPGA, XILINX ISE, АВТОМАТ МУРА

Метою роботи є розробка методу побудови опису на HDL для систем реального часу, що може бути синтезовано в програмовану логічну інтегральну схему.

У роботі досліджено особливості моделі часового автомата. Розглянуто програмно-апаратний і апаратний підходи до реалізації систем реального часу.

Запропонована модель часового автомата Мура. Розроблені VHDL-моделі системи керування дорожнім світлофором. Проаналізовані результати моделювання та синтезу.

ABSTRACT

The explanatory note contains 86 pages, 64 figures, 15 reference sources.

TIMED AUTOMATA, REAL-TIME CONTROL SYSTEM, VHDL, TEMPLATE, FPGA, XILINX ISE, MOORE FINITE STATE MACHINE

The purpose of the work is to develop a method for constructing an HDL description for real-time systems that can be synthesized into a programmable logic integrated circuit.

The features of the timed automata model are investigated in the paper. Softwarehardware and hardware approaches to real-time systems realization are viewed. A model of Moore timed finite state machine is proposed. VHDL models of traffic light control system are developed. The results of modeling and synthesis are analyzed.

ЗМІСТ

ВСТУП.....	8
1 ДІАГРАМИ СТАНІВ.....	9
2 ЧАСОВИЙ АВТОМАТ.....	12

3 РЕГІОННИЙ АВТОМАТ.....	15
4 ЧАСОВИЙ АВТОМАТ З ЄДИНИМ ТАЙМЕРОМ.....	27
5 ЧАСОВИЙ РОЗШИРЕНИЙ АВТОМАТ.....	37
6 ЧАСОВИЙ АВТОМАТ З ТАЙМАУТАМИ І ЧАСОВИМИ ОБМЕЖЕННЯМИ.....	42
7 СТРУКТУРНИЙ ЧАСОВИЙ АВТОМАТ МУРА.....	47
8 АПАРАТНИЙ ПІДХІД ДО РЕАЛІЗАЦІЇ СИСТЕМ УПРАВЛІННЯ РЕАЛЬНОГО ЧАСУ.....	52
9 ПОСТАНОВКА ТЕХНІЧНОГО ЗАВДАННЯ.....	56
10 VHDL-МОДЕЛІ ЧАСОВОГО АВТОМАТА МУРА.....	60
10.1 Трьохпроцесна модель.....	62
10.2 Однопроцесна модель.....	65
10.3 Двопроцесна модель.....	66
11 РЕЗУЛЬТАТИ СИНТЕЗУ.....	68
12 РЕЗУЛЬТАТИ ВЕРИФІКАЦІЇ МОДЕЛЕЙ.....	71
12.1 Моделювання до імплементації.....	71
12.2 Моделювання після імплементації на FPGA.....	78
12.2.1 Трьохпроцесна модель.....	78
12.2.2 Двопроцесна модель.....	82
12.2.3 Однопроцесна модель.....	84
ВИСНОВКИ.....	86
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	89
Додаток А Графічний матеріал атестаційної роботи.....	
..... Ошибка! Закладка не определена.	
Додаток Б VHDL-опис системи керування дорожнім світлофором	Ошибка!
Закладка не определена.	
Додаток В Матеріали публікації.....	Ошибка!

Закладка не определена.

ВСТУП

Під системою управління реального часу розуміють систему, коректність функціонування якої залежить не лише від логічних значень управляючих дій, але і від часу, протягом якого ці дії виконуються. Таким чином, в системах реального часу завдання мають бути виконані в межах заданого проміжку часу.

Прикладами задач управління, що виконуються системами реального часу можуть вважатися задачі управління складальним цехом, освітленням чи опаленням приміщень, дорожнім рухом та інші.

Поширеним підходом до проектування системи реального часу є програмно-апаратний підхід. При програмно-апаратному підході для заданої апаратури обчислювальних пристроїв, яка може являти собою, наприклад, мікропроцесор, мікроконтролер, програмований логічний контролер формується програма, що описує алгоритм функціонування пристрою.

Досліджуються можливості використання апаратного підходу до реалізації таких систем, який передбачає розробку HDL-опису системи й синтез опису в ПЛІС або ASIC за допомогою засобів системи автоматизованого проектування. Основні труднощі, що виникають при такому підході - це необхідність розробки інтерфейсу вводу-виводу та забезпечення необхідних часових параметрів.

Мета роботи - розробка методу побудови опису на HDL для систем реального часу, що може бути синтезований в програмовану логічну інтегральну схему.

Актуальність даної роботи пов'язана з певними недоліками програмноапаратного підходу до реалізації систем реального часу:

- порівняно низька швидкодія розроблюваних систем;

- значні обмеження на алгоритм функціонування пристрою, що накладаються архітектурою обраного апаратного забезпечення.

Поставлені наступні завдання.

1. Дослідження часового автомата як моделі систем реального часу.
 2. Розробка структурної моделі часового автомата Мура.
 3. Аналіз особливостей апаратного підходу до реалізації пристроїв.
 4. Розробка VHDL-шаблонів опису систем реального часу.
 5. Аналіз результатів синтезу та верифікації розроблених VHDL-моделей.
- Об'єктом дослідження є моделі систем реального часу в системі автоматизованого проектування.

Предметом дослідження є застосування моделі часового автомата при проектуванні систем управління реального часу на VHDL.

Технічне завдання полягає в розробці VHDL-моделей системи управління дорожнім світлофором з використанням САПР на основі моделі часового автомата, що можуть бути синтезовані в програмовану логічну інтегральну схему.

1 ДІАГРАМИ СТАНІВ

В [1] розглядається проблема специфікації складних реактивних систем. Реактивна система, на відміну від трансформаційної, характеризується тим, що її функціонування визначається реакцією на зовнішні й внутрішні події. Сутність

проблеми полягає в труднощі створення наочного опису поведінки реактивної системи, який водночас був би достатньо детальним і формальним і дозволяв би використовувати автоматичне моделювання.

У підходу, заснованого на використанні стандартних графів переходів (state diagram) існують певні недоліки. До таких недоліків відноситься експоненціальне зростання числа станів. Крім того, наголошується, що стандартні графи переходів не підходять для складних специфікацій, важкі для читання і покрокового уточнення.

В [1] для опису поведінки реактивних систем пропонується використовувати діаграми станів (statechart), які розширюють стандартні графи переходом в трьох аспектах: ієрархічності, паралелізму та комунікації. Стверджується, що діаграма станів є наочним і економним засобом опису складного алгоритму реактивного функціонування.

При опису систем управління реального часу часто виникає необхідність обмежити час перебування системи в певному стані. Таким чином, існує необхідність в графічному позначенні, яке б пов'язувало стан з часовим параметром.

Для цієї мети визначається, що стан може мати верхню і нижню межі часу перебування. Нижня межа позначає, що події не можуть вивести систему з даного стану, поки час перебування в даному стані не досяг цієї межі. Також вводиться спеціальна спільна подія timeout, що активує перехід зі стану при знаходженні системи в цьому стані певний час, який визначається верхньою межею.

На рисунку 1.1 зображено приклад використання такої нотації. Спеціальний символ у вигляді ламаної (squiggle) позначає, що стани А, В мають верхні межі часу перебування: 5 і 3 секунди відповідно.

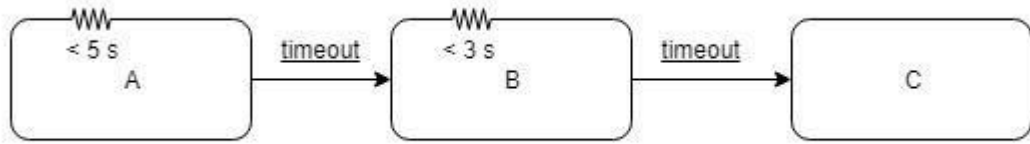


Рисунок 1.1 - Діаграма станів з таймаутами

Діаграма станів представляє керуючу частину системи, що відповідає за прийняття рішень, які залежать від часу і визначають поведінку системи. Керуюча частина впливає на поведінку інших компонентів шляхом генерації подій і зміни значень умов. Виділяється два типи керуючих сигналів: дія (action) і діяльність (activity).

Дія - це подія, час існування якої в ідеалі нульовий. Діяльність співвідноситься з дією подібно до того, як умова співвідноситься з подією. Діяльність завжди займає певний час, наприклад, виконання тривалих обчислень, тоді як дії є миттєвими.

В [1] для кожної діяльності X вводиться дві спеціальні події: $start(X)$ і $stop(X)$, які позначають початок і кінець часу існування діяльності X . Крім того, для кожної діяльності вводиться спеціальна умова $active(X)$, значення якої залежить від того, чи виконується діяльність X в даний момент.

Для коректного опису таких дій і діяльностей за допомогою діаграми станів необхідно скласти фізичний і функціональний опис системи, що встановлює функції, які підтримуються даною системою.

2 ЧАСОВИЙ АВТОМАТ

В [2, 3] розглядається метод перевірки моделі (model checking) на основі часового автомата (timed automata). Стверджується, що методи перевірки моделей, які не допускають явного моделювання часу, не підходять для аналізу систем управління реального часу, коректність функціонування яких залежить від величин різних затримок. До таких моделей належать LTL-модель (лінійна темпоральна логіка), STL-модель (темпоральна логіка розгалуженого часу). Відповідно, пропонується використовувати часові автомати (timed automata) як формальну нотацію для моделювання поведінки систем управління реального часу.

Щоб описати поведінку системи реального часу, розглядається граф переходів автомата, доповнений кінцевою множиною таймерів, що приймають дійсні значення. Вершини графа називають позиціями, а ребра називають переходами. Таймери можуть бути скинуті в нуль одночасно з будь-яким переходом. У будь-який момент свідчення таймера рівні часу, що минув з моменту останнього скидання. З кожним переходом пов'язано часове обмеження (clock constraint, в [4] використовується запобіжник), яке означає, що даний перехід може бути здійснений тільки в тому випадку, якщо поточні значення таймерів задовольняють цьому обмеженню. З кожною позицією пов'язане обмеження на таймери, зване інваріантом; система може перебувати в даній позиції, доти виконується її інваріант.

В [3] часовий автомат визначається як кортеж $(L, L_0, \Sigma, X, I, E)$, який складається з наступних елементів:

- L - кінцева множина позицій,
- L_0 - множина початкових позицій,

- Σ - кінцева множина міток,
- X - кінцева множина таймерів,
- I - відображення, що пов'язує позицію з деяким часовим обмеженням,
- E - множина переходів.

Перехід визначається як кортеж $(s, a, \varphi, \lambda, s')$ і відповідає зміні позиції s на позицію s' у відповідь на подію a , φ - це часове обмеження, яке визначає, коли даний перехід може бути виконаний, λ - це множина таймерів, які при виконанні переходу будуть скинуті в нуль.

Приклад часового автомата з часовим обмеженням на переході й приклад запуску даного автомата, представлений на рисунку 2.1. Перехід може бути здійснений, тільки якщо значення таймеру x перевищує 5. Автомат може знаходитися в позиції s необмежений час через те, що відповідний інваріант завжди виконується.

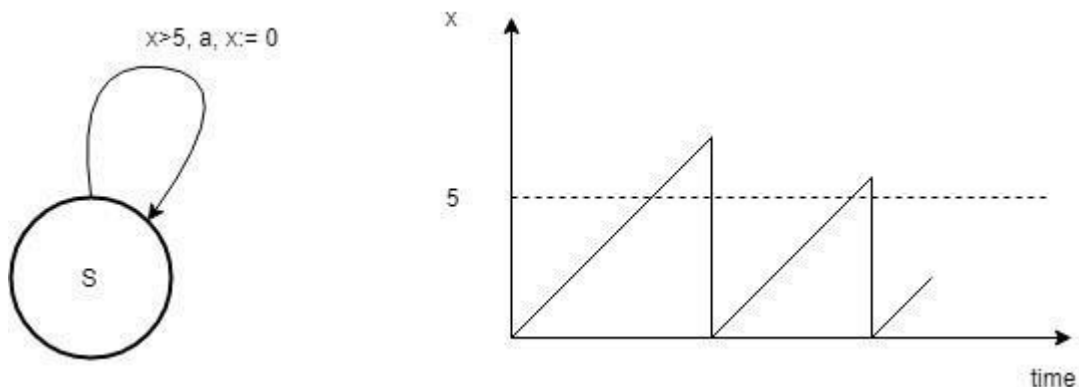


Рисунок - 2.1 - Приклад часового автомата з часовим обмеженням на переході

Приклад часового автомата з інваріантом представлений на рисунку 2.2. Автомат може знаходитися в позиції s , поки значення таймера x не перевищує 5. Перехід повинен бути виконаний в проміжок, коли $4 < x < 5$.

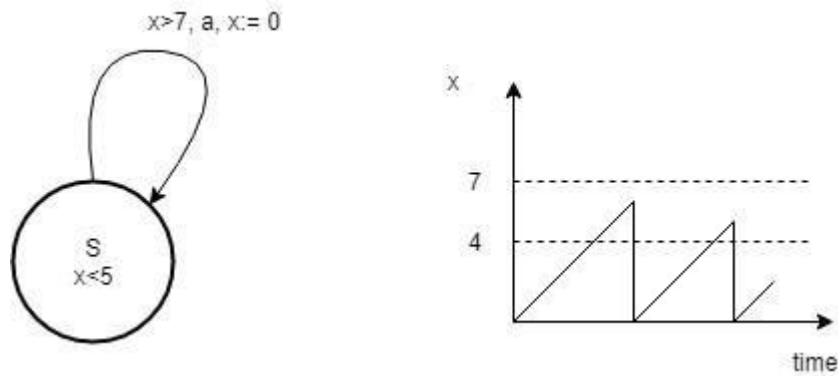


Рисунок 2.2 - Приклад часового автомата з інваріантом

Часова оцінка (clock interpretation, в [4] - показники таймерів) призначає дійсне значення кожному таймеру; тобто це відображення множини таймерів на множину невід'ємних дійсних чисел. Часова оцінка задовольняє деякому часовому обмеженню, якщо вираз, що відповідає обмеженню, є істинним при значеннях, відповідних даній оцінці.

Для опису часового автомата може бути побудована система переходів. Система переходів S_A часового автомата A визначається як кортеж $(Q, Q^0, \Sigma, \rightarrow)$, що складається з наступних елементів:

- Q - множина станів;
- Q^0 - початковий стан;
- Σ - множина подій;
- \rightarrow - множина переходів.

Станом такої системи переходів для часового автомата буде пара елементів (s, v) , яка складається з позиції часового автомата s і часової оцінки v , яка задовольняє інваріант $I(s)$. Стан системи переходів буде початковим, якщо відповідна позиція автомата є ініціальною, а часова оцінка ставить у відповідність усім таймерам нульові значення. Можна виділити два типи переходів в системі переходів: просування часу на деяке значення (в [4] - перехід за затримкою) і зміна позиції часового автомата у відповідь на подію (в [4] - перехід за дією).

3 РЕГІОННИЙ АВТОМАТ

Позиція s часового автомата A досяжна, якщо деякий стан q системи переходів S_A є досяжним і містить позицію s . Перевірка аспектів безпеки систем реального часу може бути заснована на перевірці досяжності станів часового автомата. Оскільки число станів системи переходів S_A часового автомата нескінченно, щоб вирішити задачу досяжності пропонується ввести відношення еквівалентності \sim на часових оцінках.

При введенні такого відношення еквівалентності, необхідно, щоб число класів еквівалентності - регіонів (в [4] - часових областей) - було кінцевим. Крім того, необхідно, щоб для кожної часової оцінки, що належить одному класу, множина виконуваних інваріантів і часових обмежень була однаковою.

Розглянемо ідеї, що лежать в основі вимог до часових оцінок, за якими виконується побудова регіонів [5].

Для часової оцінки v і таймера x використовується наступна нотація: $v(x) = [x] + fr(x)$, де $[x]$ - це ціла частина значення таймера, а $fr(x)$ - дрібна частина.

Розглянемо часовий автомат на рисунку 3.1 і дві часові оцінки: $v(x) = 11.1$ і $v'(x) = 12.8$. Слід зауважити, що $c_x = 11$ - це максимальне значення, з яким порівнюється значення таймера x . У станах з часовою оцінкою $v(x)$ і $v'(x)$ всі часові обмеження виконуються тому, що $v(x) > c_x$ і $v'(x) > c_x$.

Таким чином, часові оцінки v і v' знаходяться в одному регіоні $v \sim v'$, якщо $v(x) > c_x$ і $v'(x) > c_x$ для всіх $x \in C$.

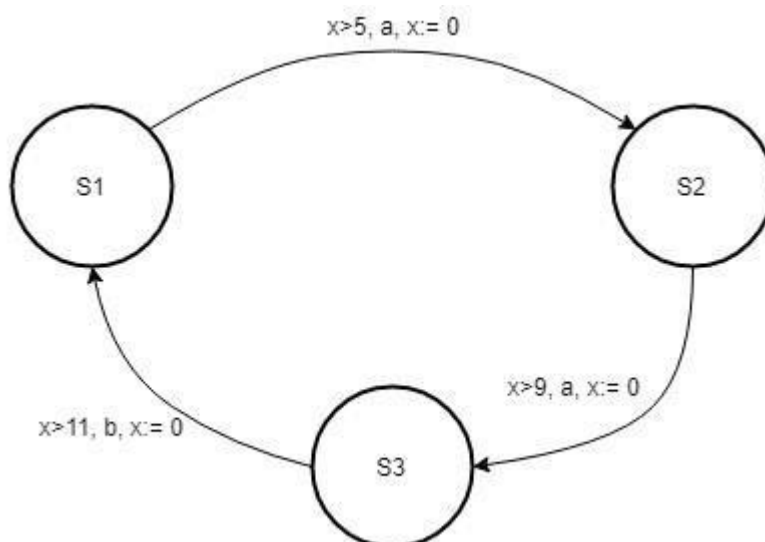


Рисунок 3.1 - Часовий автомат з $c_x = 11$

Розглянемо часовий автомат на рисунку 3.2 і дві часові оцінки: $v(x) = 2.2$ і $v'(x) = 2.9$. Слід зауважити, що часові обмеження й інваріанти задаються в цілих числах. У станах з часовою оцінкою $v(x)$ і з часовою оцінкою $v'(x)$ виконується часове обмеження на переході від $s1$ в $s2$ і не виконується часове обмеження на переході від $s2$ в $s1$, оскільки $2 \leq v(x) < 3$ і $2 \leq v'(x) < 3$. Таким чином, можна вважати, що дробові частини часових оцінок несуттєві.

Часові оцінки v і v' знаходяться в одному регіоні $v \sim v'$, якщо $[v(x)] = [v'(x)]$ для всіх $x \in C$.

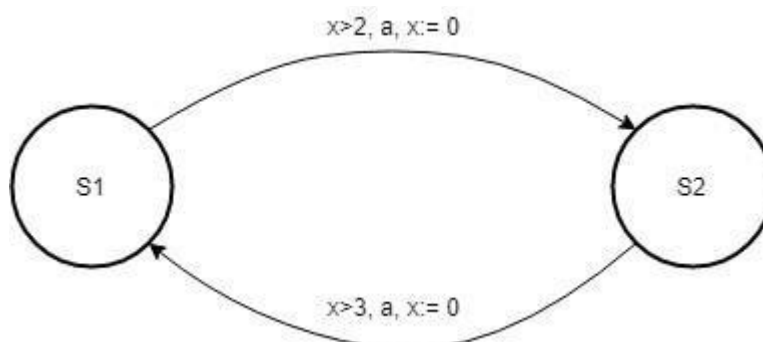


Рисунок 3.2 - Часовий автомат з $[v(x)] = [v'(x)]$

Розглянемо часовий автомат на рисунку 3.3 і два стани: $s = (s1, v)$ і $s' = (s1, v')$ при $v(x) = 2.1$, $v'(x) = 2.9$, $v(y) = 2.9$ і $v'(y) = 2.2$. По закінченню часу 0.5 в першому випадку автомат зі стану s перейде в стан s'' з $v(x) = 2.6$ і $v(y) = 3.4$, таким чином, перехід з $s1$ в $s3$ буде активним. У другому випадку для того, щоб виконати перехід з $s1$ в $s3$ необхідно очікування автомата в позиції $s1$ протягом 0.8, але через інваріант таке очікування неможливе. Таким чином, порядок дрібних частин різних таймерів часових оцінок є важливим. $v \sim v'$, якщо $fr(v(x)) \leq fr(v(y))$ і $fr(v'(x)) \leq fr(v'(y))$ для всіх $x, y \in C$ при $v(x) \leq c_x, v(y) \leq c_y$.

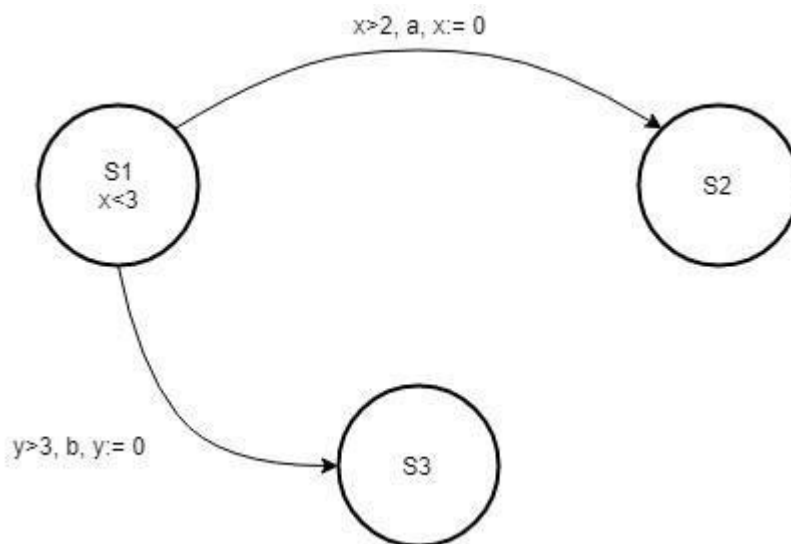


Рисунок 3.3 - Часовий автомат із таймерами x, y

Розглянемо часовий автомат на рисунку 3.4 і два стани: $s = (s0, v)$ і $s' = (s0, v')$ при $v(x) = 3.0$, $v'(x) = 3.1$. У першому випадку перехід зі стану $s1$ в $s2$ допустимий, тоді як у другому випадку блокований. Таким чином, якщо дрібна частина дорівнює нулю, то необхідний додатковий регіон.

Часові оцінки v і v' знаходяться в одному регіоні $v \sim v'$, якщо $\text{fr}(v(x)) = 0$ і $\text{fr}(v'(x)) = 0$ для всіх $x \in C$ при $v(x) \leq c_x$.

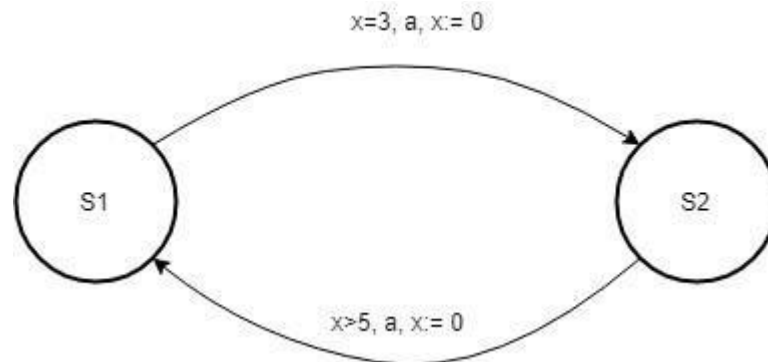


Рисунок 3.4 - Часовий автомат з обмеженням на переході $x = 3$

Розглянемо часовий автомат на рисунку 3.5 і два стани: $s = (s1, v)$ і $s' = (s1, v')$ при $v(x) = 1.4$, $v'(x) = 1.4$, $v(y) = 2.4$ і $v'(y) = 2.5$. По закінченню часу 0.6 в першому випадку автомат зі стану s перейде в стан s'' з $v(x) = 2.0$ і $v(y) = 3.0$, таким чином, обидва переходи зі стану $s1$ в $s2$ і $s3$ активні. У другому випадку для того, щоб виконати перехід з $s1$ в $s2$ необхідно очікування автомата в позиції $s1$ протягом 0.6, але через інваріант, що обмежує значення таймера y , таке очікування неможливо. Таким чином, якщо дробові частини значень різних таймерів часових оцінок рівні, то необхідно виділити окремий регіон.

$v \sim v'$, якщо $\text{fr}(v(x)) = \text{fr}(v(y))$ і $\text{fr}(v'(x)) = \text{fr}(v'(y))$ для всіх $x, y \in C$ при $v(x) \leq c_x, v(y) \leq c_y$.

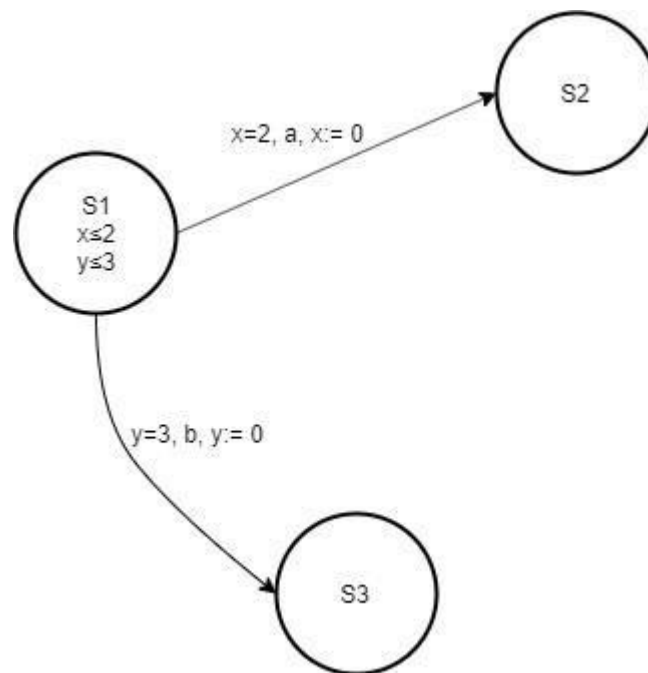


Рисунок 3.5 - Часовий автомат з обмеженнями на переходах $x = 2$, $y = 3$

Таким чином, $v \sim v'$, якщо виконуються всі п'ять умов:

- $v(x) > c_x$, якщо $v'(x) > c_x$, для всіх $x \in C$,
- $[v(x)] = [v'(x)]$ для всіх $x \in C$ при $v(x) \leq c_x$,
- $\text{fr}(v(x)) \leq \text{fr}(v(y))$, якщо $\text{fr}(v'(x)) \leq \text{fr}(v'(y))$ для всіх $x, y \in C$

при $v(x) \leq c_x$, $v(y) \leq c_y$,

- $\text{fr}(v(x)) = 0$, якщо $\text{fr}(v'(x)) = 0$ для всіх $x \in C$ при $v(x) \leq c_x$,
- $\text{fr}(v(x)) = \text{fr}(v(y))$, якщо $\text{fr}(v'(x)) = \text{fr}(v'(y))$ для всіх $x, y \in C$

при $v(x) \leq c_x$, $v(y) \leq c_y$.

Розглянемо приклад послідовної побудови регіонів при $C = \{x, y\}$, $c_x = 1$, $c_y = 3$.

Вимога того, що $v(x) > c_x$, якщо $v'(x) > c_x$, для всіх $x \in C$, призводить до 4 класів еквівалентності: $[0 \leq x \leq 1, y > 3]$, $[0 \leq x \leq 1, 0 \leq y \leq 3]$, $[x > 1, y > 3]$, $[x > 1,$

$0 \leq y \leq 3$]. Дані регіони зображені на рисунку 3.6.

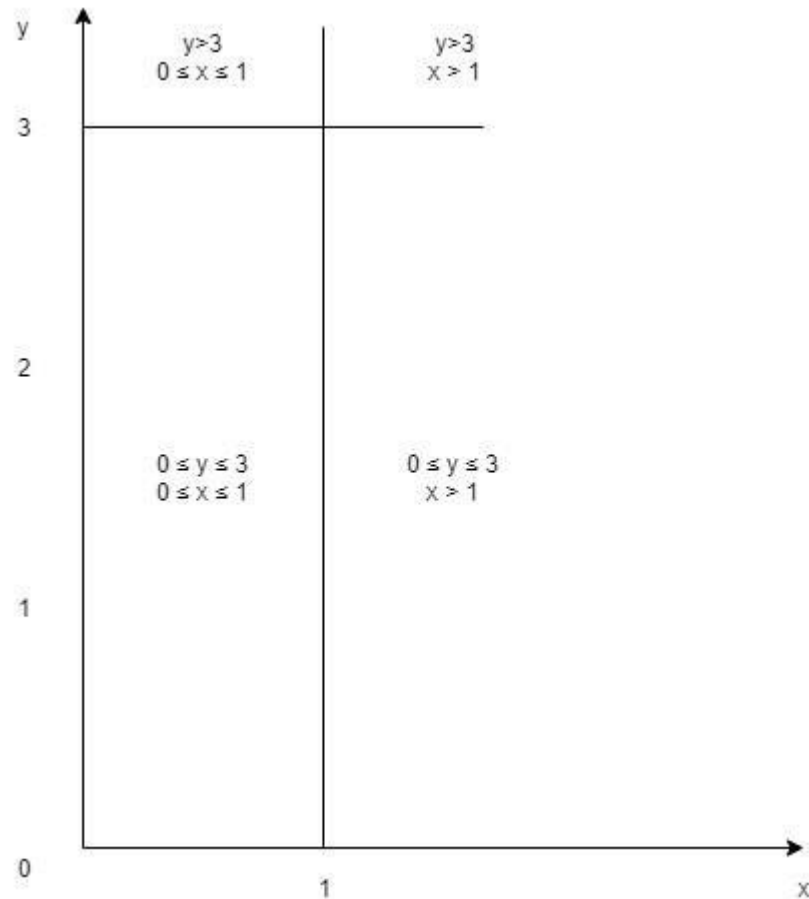


Рисунок 3.6 - Процес розбиття на регіони для виконання вимоги $v(x) > c_x$, якщо $v'(x) > c_x$, для всіх $x \in C$

Вимога того, що $[v(x)] = [v'(x)]$ для всіх $x \in C$ при $v(x) \leq c_x$, призводить, наприклад, до розбиття регіону $[0 \leq x \leq 1, 0 \leq y \leq 3]$ на наступні регіони: $[0 \leq x < 1, 0 \leq y < 1]$, $[0 \leq x < 1, 1 \leq y < 2]$, $[0 \leq x < 1, 2 \leq y < 3]$, $[x = 1, y = 3]$, $[x = 1, 0 \leq y < 1]$, $[x = 1, 1 \leq y < 2]$, $[x = 1, 2 \leq y < 3]$, $[x = 1, y = 3]$. В результаті таких розділень буде отримано 15 регіонів. Дані регіони зображені на рисунку 3.7.

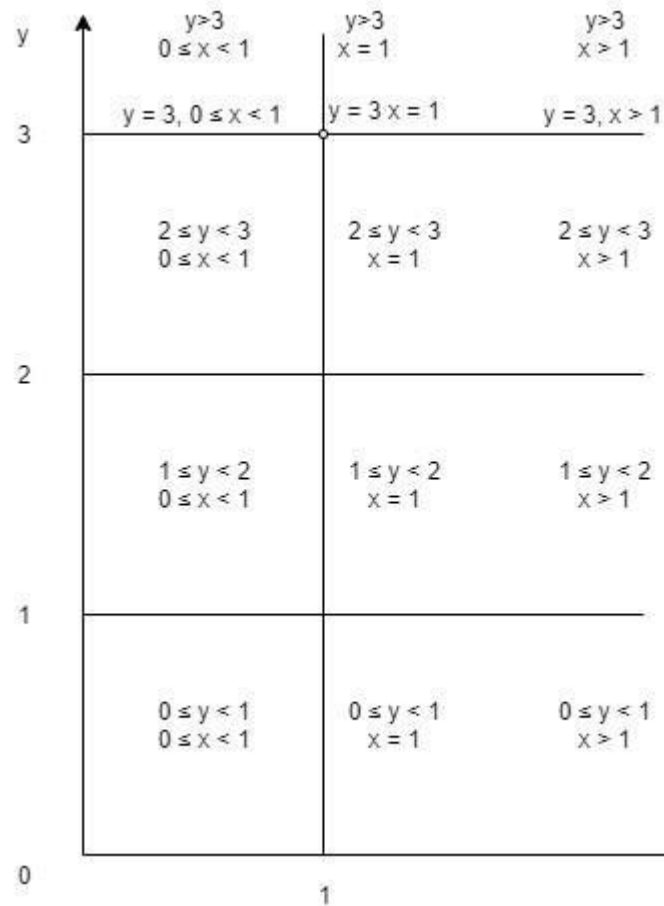


Рисунок 3.7 - Процес розбиття на регіони для виконання вимоги $[v(x)] = [v'(x)]$ для всіх $x \in C$ при $v(x) \leq c_x$

Вимога того, що $\text{fr}(v(x)) \leq \text{fr}(v(y))$, якщо $\text{fr}(v'(x)) \leq \text{fr}(v'(y))$ для всіх $x, y \in C$ при $v(x) \leq c_x, v(y) \leq c_y$, призводить, наприклад, до розбиття регіону $[0 \leq x < 1, 1 \leq y < 2]$ на наступні регіони: $[0 \leq x < 1, 1 \leq y < 2, \text{fr}(y) \geq \text{fr}(x)]$ і $[0 \leq x < 1, 1 \leq y < 2, \text{fr}(y) < \text{fr}(x)]$. Процес такого розбиття зображений на рисунку 3.8. В результаті таких розділень буде отримано 18 регіонів.

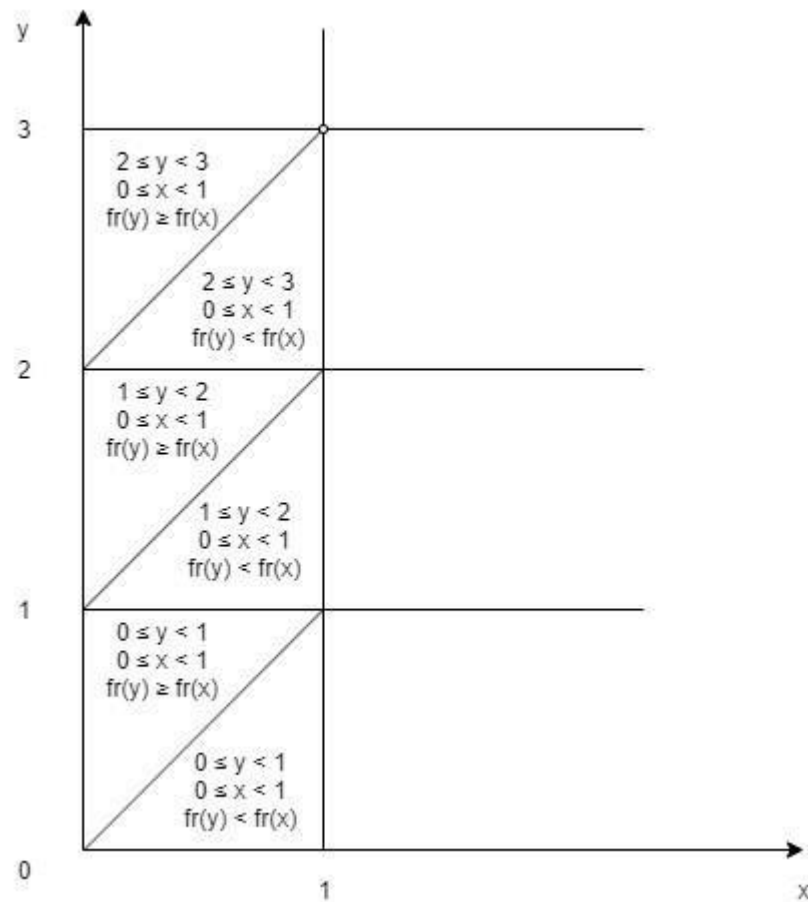


Рисунок 3.8 - Процес розбиття на регіони для виконання вимоги про порядок дрібних частин значень таймерів

Вимога того, що $fr(v(x)) = 0$, якщо $fr(v'(x)) = 0$ для всіх $x \in C$ при $v(x) \leq c_x$, призводить, наприклад, до розбиття регіону $[0 \leq x < 1, 1 \leq y < 2, fr(y) < fr(x)]$ на наступні регіони: $[x = 0, y = 1]$, $[x = 0, 1 < y < 2]$, $[0 < x < 1, y = 1]$, $[0 < x < 1, 1 < y < 2, fr(y) < fr(x)]$. Процес такого розбиття зображений на рисунку 3.9. В результаті таких розділень буде отримано 35 регіонів.

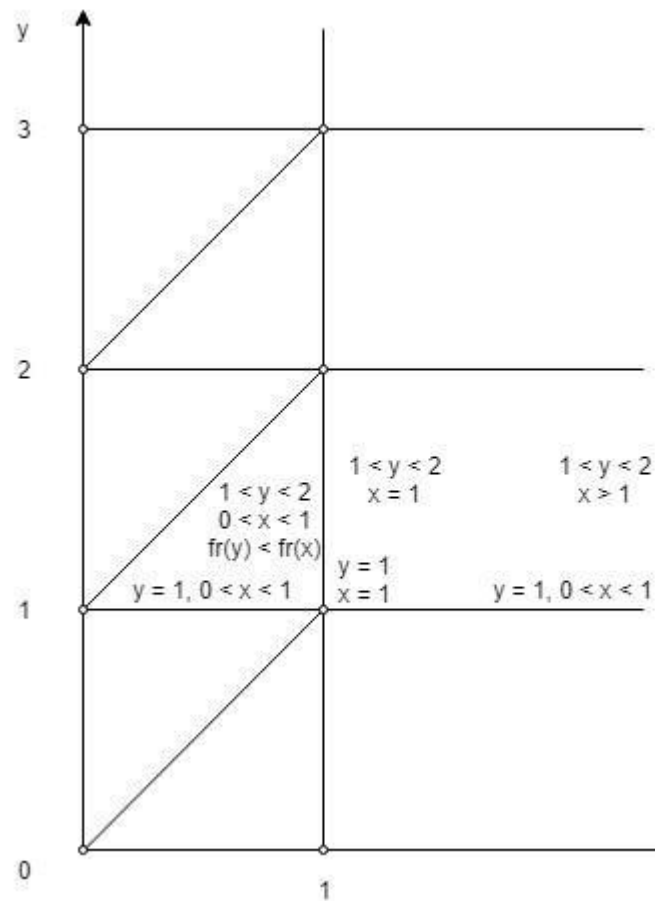


Рисунок 3.9 - Процес розбиття на регіони для виконання вимоги $\text{fr}(v(x)) = 0$, якщо $\text{fr}(v'(x)) = 0$ для всіх $x \in C$ при $v(x) \leq c_x$

Вимога того, що $\text{fr}(v(x)) = \text{fr}(v(y))$, якщо $\text{fr}(v'(x)) = \text{fr}(v'(y))$ для всіх $x, y \in C$ при $v(x) \leq c_x, v(y) \leq c_y$, призводить, наприклад, до розбиття регіону $[0 < x < 1, 1 < y < 2, \text{fr}(y) \geq \text{fr}(x)]$ на наступні регіони: $[0 < x < 1, 1 < y < 2, \text{fr}(y) > \text{fr}(x)]$, $[0 < x < 1, 1 < y < 2, \text{fr}(y) = \text{fr}(x)]$. Процес розбиття представлений на рисунку 3.10. Кінцева кількість класів еквівалентності - 38, до них відносяться 8 кутових точок, 11 відкритих областей і 19 прямокутних ділянок.

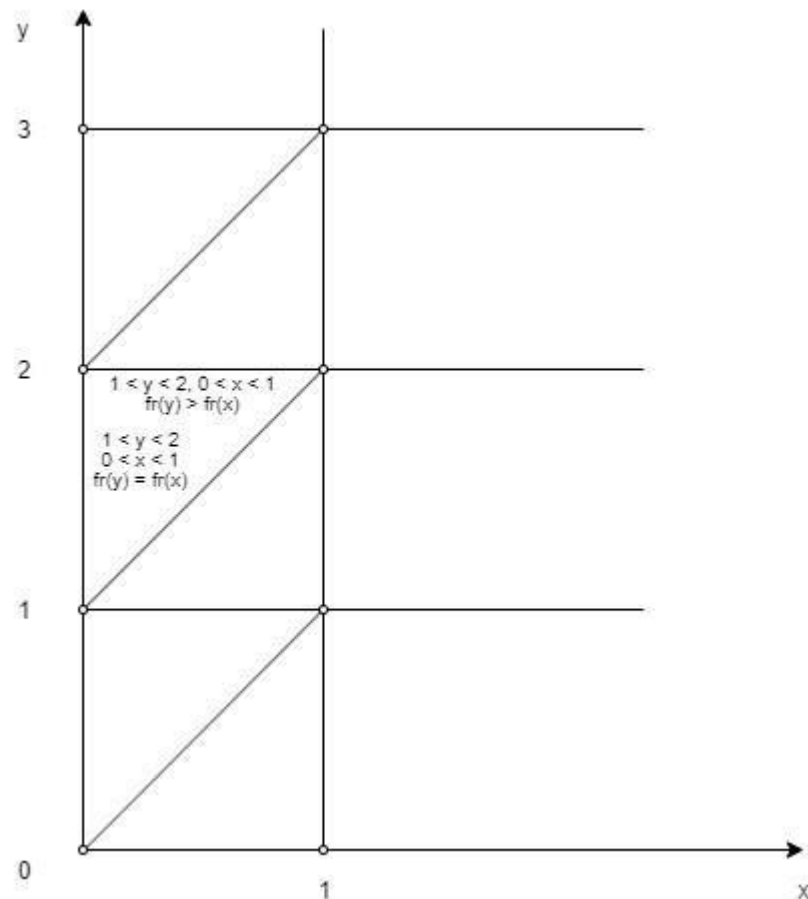


Рисунок 3.10 - Процес розбиття на регіони для виконання вимоги про рівність дрібних частин значень таймерів

Розглянемо дві часові оцінки v і v' , що належать одному регіону $[0 < x < 1, 1 < y < 2, fr(y) \geq fr(x)]$ на рисунку 3.11. Можна помітити, що приплині часу автомат, незалежно від того, чи має він часову оцінку v або v' , буде проходити одні й ті ж регіони, оскільки зміна часової оцінки представляється у вигляді відрізка прямої, що має кутовий коефіцієнт $k = 1$. Розглянемо перехід між позиціями автомата. Оскільки для довільної часової оцінки всередині регіону виконується одна і та ж множина інваріантів і часових обмежень, то і множина допустимих переходів усередині регіону однакова. При переході між позиціями у відповідь на подію може відбутися скидання таймера в 0 , така зміна часової оцінки представляється проекцією на вісь, яка відповідає даному таймеру. І отже, автомат, незалежно від того, чи має він часову оцінку v або v' , при переходах між станами може потрапити в той самий регіон.

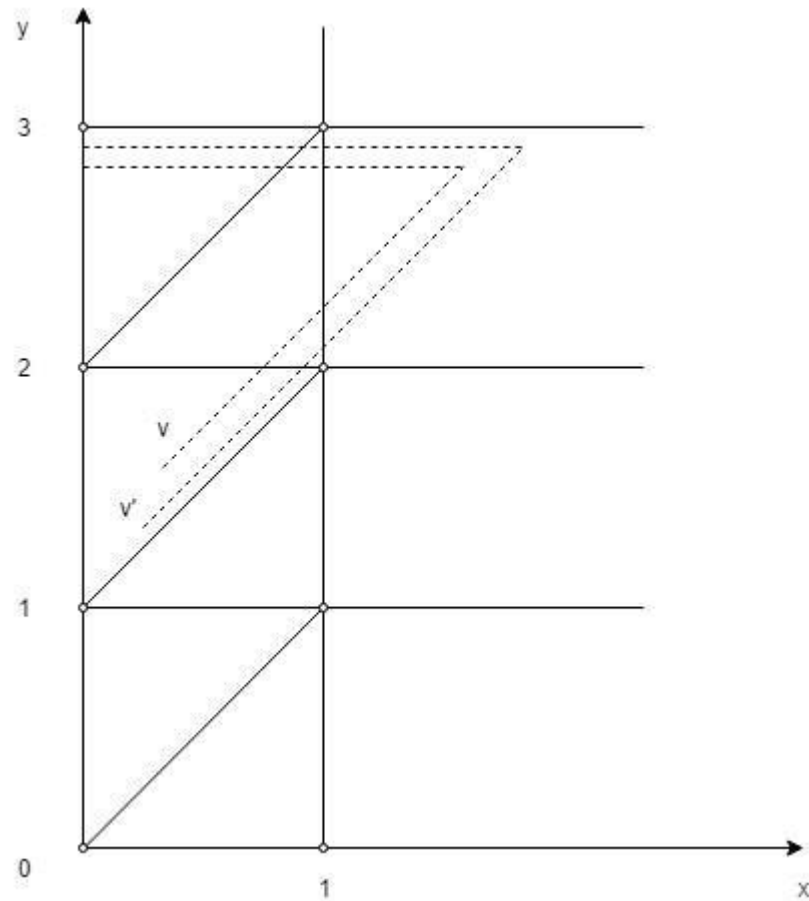


Рисунок 3.11 - Зміна часової оцінки часового автомата

На основі процедури розбиття часових оцінок на класи еквівалентності для часового автомата може бути отриманий регіонний автомат. Регіонний автомат - це система переходів, яка містить кінцеву множину станів-регіонів, початковий стан і відношення переходів. За рахунок того, що число регіонів кінцеве і множина виконуваних інваріантів і часових обмежень однакова для кожної часової оцінки, що належить одному регіону, перевірку моделі часового автомата з нескінченним числом станів можна звести до перевірки відповідного регіонного автомата.

Розглянемо часовий автомат на рисунку 3.12. Початкова позиція даного автомата - s_0 , $c_x = 1$, $c_y = 1$.

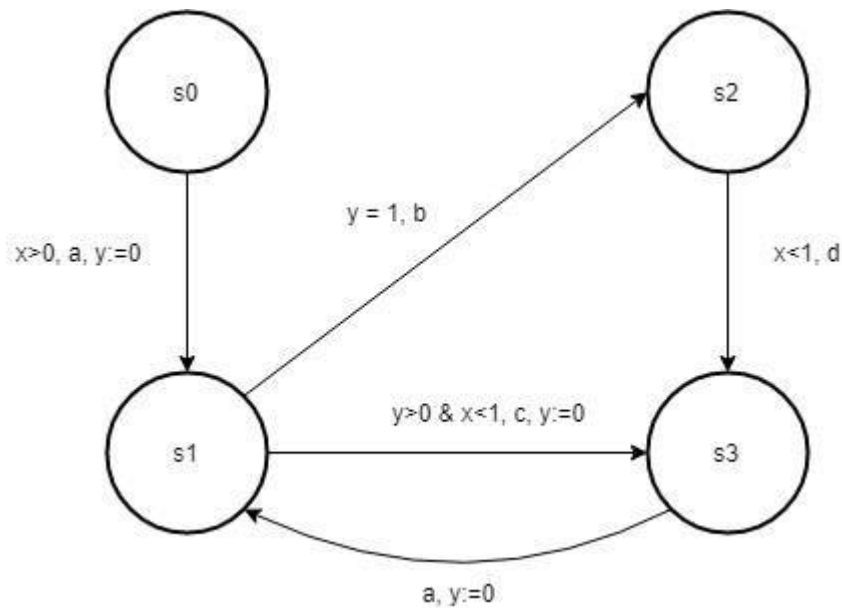


Рисунок 3.12 - Часовий автомат

Автомат з позиції s_0 може перейти тільки в позицію s_1 , при цьому таймер y буде скинутий в 0. Таким чином, в залежності від моменту появи події a і, відповідно, значення x автомат може перейти в один з наступних регіонів: $(y = 0, 0 < x < 1)$, $(y = 0, x = 1)$, $(y = 0, x > 1)$. З позиції s_1 автомат за подією b може потрапити в позицію s_2 , при цьому, через обмеження на переході $y = 1$, автомат може перейти тільки в регіон $(y = 1, x > 1)$. За подією c автомат може перейти в позицію s_3 , але через обмеження на переході автомат може виконати даний перехід тільки з регіону $(y > 1, x < 1)$. При переході з s_2 в s_3 , таймер y буде скинутий, таким чином, автомат потрапить в регіон $(y = 0, 0 < x < 1)$. Перехід з позиції s_3 в s_1 аналогічний переходу з s_0 в s_1 .

Отриманий регіонний автомат наведено на рисунку 3.13.

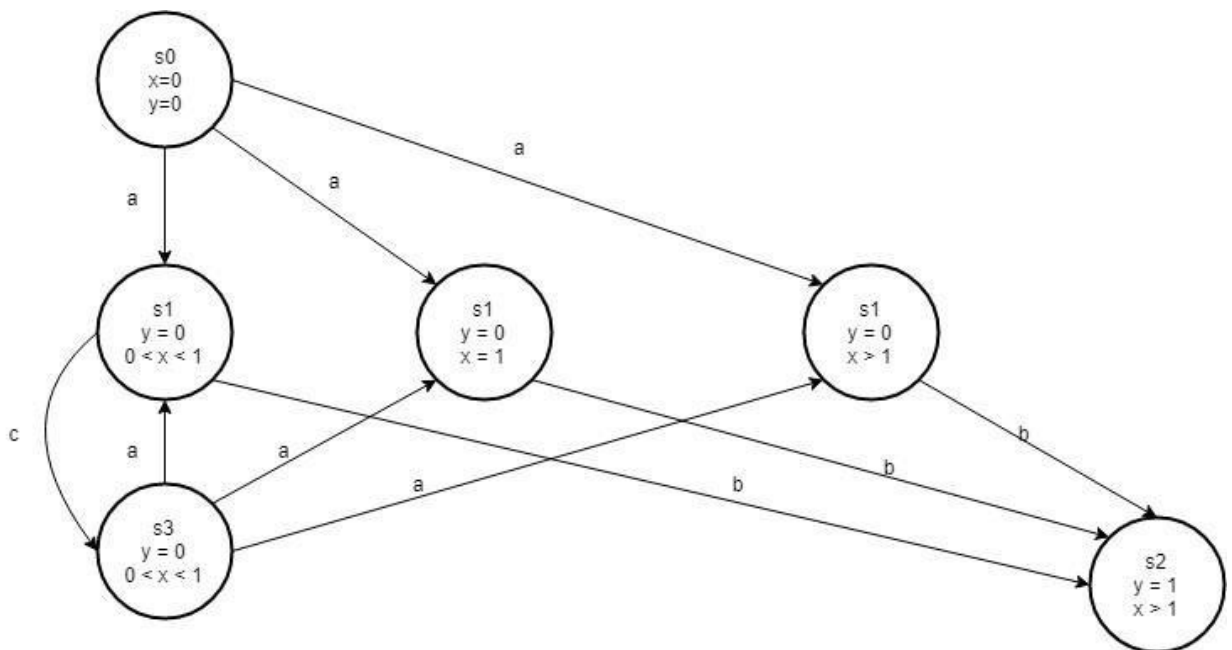


Рисунок 3.13 - Регіонний автомат

Аналіз побудованого регіонного автомата показує, що через часові обмеження перехід зі стану s2 ніколи не буде дозволений.

4 ЧАСОВИЙ АВТОМАТ З ЄДИНИМ ТАЙМЕРОМ

В [6] досліджуються методи формального тестування, які враховують часові характеристики системи. Розглядаються три множини допустимих значень, до яких можуть належати часові змінні:

- фіксовані значення, - випадкові значення, - часові інтервали.

Для опису поведінки системи використовується модель часового автомата. При цьому, підхід до моделювання часу, який використовується в [3] і [6] повністю різниться [7]. Замість множини таймерів пропонується використання

однієї часової змінної. Формально запропонована модель часового автомата (timed finite state machine) являє собою кортеж $M = (S, I, O, Tr, s_{in})$:

- S - кінцева множина станів,
- I - множина вхідних дій,
- O - множина вихідних дій,
- Tr - множина переходів, - s_{in} - початковий стан.

Нехай $Time$ буде областю визначення значень часу. Перехід, що належить Tr , є кортежем (s, s', i, o, d) :

- $s \in S$ - початковий стан переходу,
- $s' \in S$ - кінцевий стан переходу,
- $i \in I$ - вхідна дія,
- $o \in O$ - вихідна дія,
- $d \in Time$ - час, протягом якого перехід буде завершено.

Часовий автомат є повністю визначеним, якщо для всіх станів $s \in S$ і вхідної дії $i \in I$ існують $s' \in S$, $o \in O$, $d \in Time$ такі, що $(s, s', i, o, d) \in Tr$. Часовий автомат є спостережуваним, якщо не існує двох різних переходів (s, s_1, i, o, d_1) і (s, s_2, i, o, d_2) , що належать Tr .

Перехід (s, s', i, o, d) вказує на те, що якщо автомат знаходиться в стані s , і надходить вхідна дія i , то через час, який визначений d , автомат видає вихідний сигнал o і переходить в стан s' . Залежно від використаної множини допустимих значень d може бути фіксованим значенням часу, випадковою величиною або інтервалом часу. Якщо розглядаються фіксовані значення, то $d \in \mathbb{R}_+$ є невід'ємним дійсним числом, якщо йдеться про стохастичні значення, то $d \in V$ є випадковою величиною. Якщо йдеться про часові інтервали, то $d \in I_{\mathbb{R}_+}$.

$\hat{a} = [a_1, a_2]$ - це часовий інтервал, якщо $a_1 \in \mathbb{R}_+$, $a_2 \in \mathbb{R}_+ \cup \{\infty\}$ і $a_1 \leq a_2$. Нехай

$\hat{a} = [a_1, a_2]$ і $\hat{b} = [b_1, b_2]$ - інтервали часу. Вводяться такі позначення:

- $\hat{a} \subseteq \hat{b}$, якщо $b_1 \leq a_1$ і $a_2 \leq b_2$;

- $\hat{a} \preceq \hat{b}$, якщо $a_1 \leq b_1$ і $a_2 \leq b_2$;

- $\hat{a} \ll \hat{b}$, якщо $a_2 \leq b_1$;

- $\hat{a} + \hat{b}$ - інтервал $[a_1 + b_1, a_2 + b_2]$.

Нехай $\xi \in V$ - випадкова величина. Функція розподілу ймовірності випадкової величини ξ - це функція $F_\xi: \mathbb{R}_+ \rightarrow [0,1]$, така, що $F_\xi(X) = P(\xi \leq x)$, де $P(\xi \leq x)$ - ймовірність того, що ξ прийме значення менше або рівне x .

На рисунку 4.1 представлений часовий автомат з фіксованими часовими значеннями. Для даного автомата $I = \{a_1, a_2, a_3\}$, $O = \{b_1, b_2, b_3\}$, $Tr = \{t_{12}, t_{22}, t_{23}, t_{31}, t_{33}\}$. $t_{12} = (s_1, s_2, a_1, b_1, 3)$, $t_{22} = (s_2, s_2, a_2, b_2, 4)$, $t_{23} = (s_2, s_3, a_3, b_3, 5)$, $t_{31} = (s_3, s_1, a_3, b_3, 2)$, $t_{33} = (s_3, s_3, a_1, b_1, 2)$. Наприклад, t_{23} відповідає переходу зі стану s_2 в s_3 , який займе 5 одиниць часу.

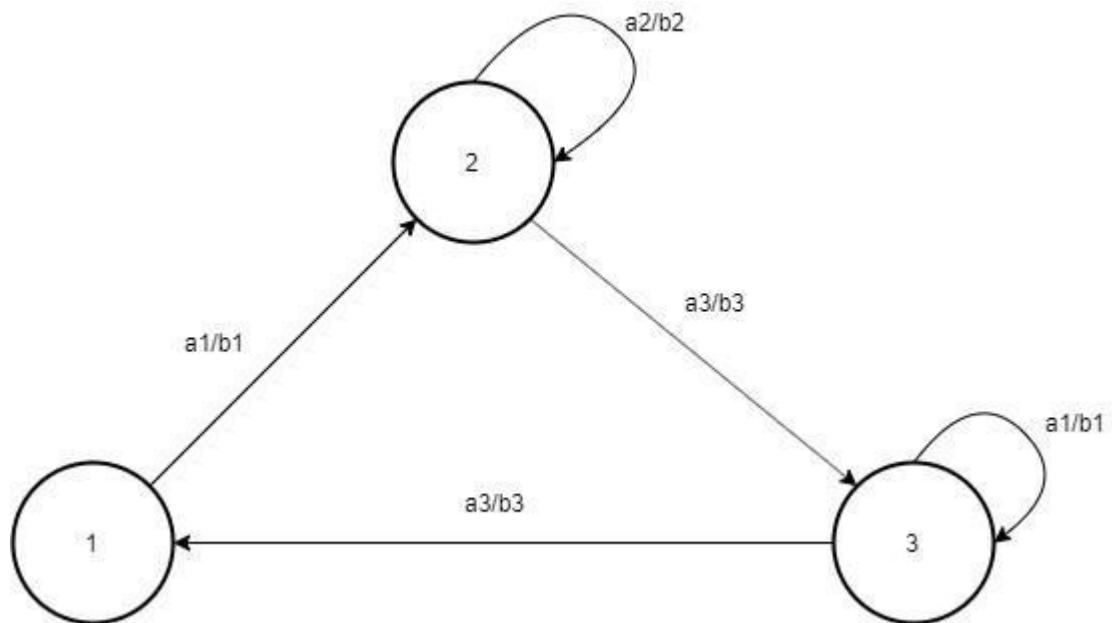
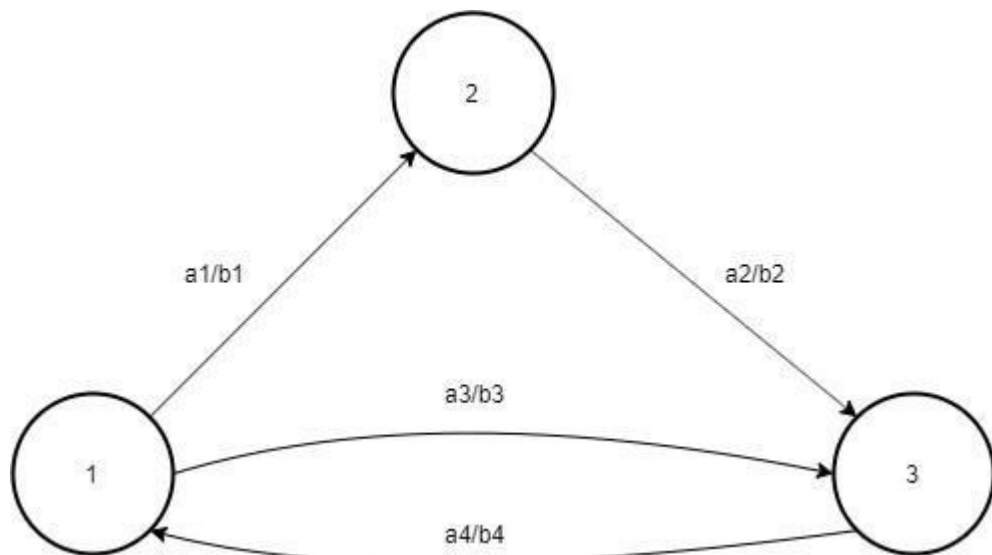


Рисунок 4.1 - Приклад автомата з фіксованими часовими затримками

На рисунку 4.2 представлений часовий автомат із затримками у вигляді інтервалів. Для даного часового автомата $I = \{a_1, a_2, a_3, a_4\}$, $O = \{b_1, b_2, b_3, b_4\}$, $Tr = \{t_{12}, t_{13}, t_{23}, t_{31}\}$. $t_{12} = (s_1, s_2, a_1, b_1, [2, 5])$, $t_{13} = (s_1, s_3, a_3, b_3, [3, 6])$, $t_{23} = (s_2, s_3, a_2, b_2, [4, 7])$, $t_{31} = (s_3, s_1, a_4, b_4, [2, 3])$. Наприклад, якщо автомат знаходиться в стані s_3 і надходить вхідна дія a_4 , то перехід t_{31} буде виконано за час, що дорівнює або перевищує 2 і менший або рівний 3.



Малюнок 4.2 - Приклад автомата з інтервальними затримками

На рисунку 4.3 представлений часовий автомат з випадковими затримками. Для даного часового автомата $I = \{a_1, a_2, a_3\}$, $O = \{b_1, b_2, b_3\}$, $Tr = \{t_{121}, t_{122}, t_{22}, t_{231}, t_{232}, t_{31}, t_{34}, t_{41}\}$. $t_{121} = (s_1, s_2, a_1, b_1, \xi_{11})$, $t_{122} = (s_1, s_2, a_2, b_2, \xi_{12})$, $t_{22} = (s_2, s_2, a_1, b_1, \xi_{21})$, $t_{231} = (s_2, s_3, a_2, b_2, \xi_{22})$, $t_{232} = (s_2, s_3, a_3, b_3, \xi_{23})$, $t_{31} = (s_3, s_1, a_1, b_1, \xi_{31})$, $t_{34} = (s_3, s_4, a_2, b_2, \xi_{32})$, $t_{41} = (s_4, s_1, a_1, b_1, \xi_{33})$, ξ_{ik} - незалежні випадкові величини, розподілені відповідно до F_i .

Таким чином, випадкова величина ξ_{11} рівномірно розподілена на інтервалі $[1; 6]$. Рівномірний розподіл дозволяє подати інтервальну затримку як випадкову. Функція розподілу випадкової величини ξ_{21} - це δ -функція, яка концентрує ймовірність в єдиній точці - в значенні 3. Така функція розподілу дозволяє подати

фіксовану затримку як стохастичну. Випадкові величини ξ_{31} , ξ_{32} , ξ_{33} розподілені експоненційно.

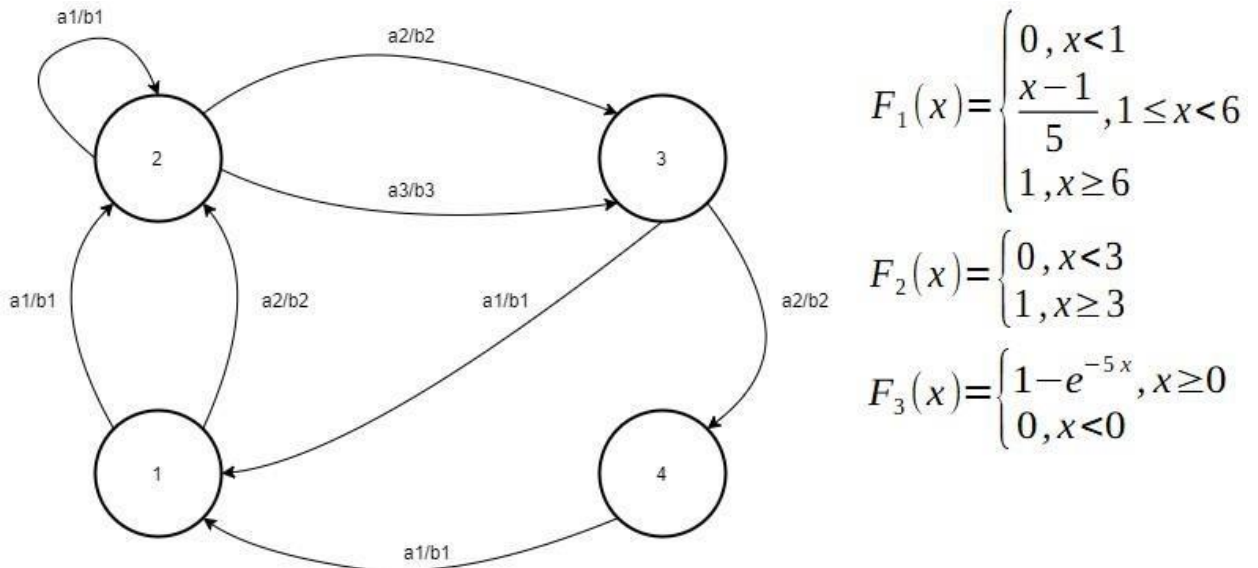


Рисунок 4.3 - Приклад автомата з випадковими затримками

Нехай $M = (S, I, O, Tr, s_{in})$ - часовий автомат. Часовою трасою (timed trace) часового автомата M є кортеж $(s, s', (i_1 / o_1, \dots, i_r / o_r), d)$, якщо існують переходи $(s, s_1, i_1, o_1, d_1), \dots, (s_{r-1}, s', i_r, o_r, d_r) \in Tr$, такі що $d = \sum d_i$.

Еволюцією (evolution) часового автомата M називають кортеж $(i_1 / o_1, \dots, i_r / o_r)$, якщо часова траса $(s_{in}, s', (i_1 / o_1, \dots, i_r / o_r), d)$ є часовою трасою автомата M . Множина еволюцій часового автомата M позначається $NTEvol(M)$. Часовою еволюцією (timed evolution) автомата M є пара $((i_1 / o_1, \dots, i_r / o_r), d)$, якщо часова траса $(s_{in}, s', (i_1 / o_1, \dots, i_r / o_r), d)$ є трасою часового автомата M . Множина часових еволюцій автомата M позначається $TEvol(M)$.

Таким чином, часова траса - це послідовність переходів. Крім того, для траси вказується час, за який були виконані відповідні переходи, тобто значення

часу, пов'язані з кожним із переходів, підсумовуються. Часова еволюція - часова траса, яка починається з ініціального стану часового автомата.

При генерації тестів передбачається, що опис моделі й імплементація представлені у вигляді часових автоматів. Більш того, імплементація представлена повністю визначеним автоматом. Таким чином, імплементація реагує на будь-яку вхідну дію.

Важливим теоретичним інструментом, який використовується при тестуванні, є відношення реалізації. По суті, відношення реалізації визначає умови, на підставі яких імплементація може вважатися відповідною специфікації. Для деяких систем імплементація є коректною, якщо для виконання завдань імплементації потрібно менше часу, ніж автомату-специфікації. Для інших систем необхідно, щоб час виконання завдань в точності відповідав специфікації.

Нехай S і I два часових автомати. I відповідає S без урахування часових характеристик (non-timely conforms), якщо для всіх $e = (i_1 / o_1, \dots, i_{r-1} / o_{r-1}, i_r / o_r) \in \text{NTEvol}(S)$ при $r \geq 1$ існує $e' = (i_1 / o_1, \dots, i_{r-1} / o_{r-1}, i_r / o'_r) \in \text{NTEvol}(I)$ і $e' \in \text{NTEvol}(S)$. Таке відношення позначається $I \text{ conf}_{\text{nt}} S$.

Ідея, що лежить в основі визначення відношення відповідності без урахування часових характеристик, полягає в тому, що реалізація не видає ніяких вихідних дій, відмінних від формальної моделі для тих вхідних даних, які вказані в моделі.

На рисунку 4.4 представлений приклад виконання відношення реалізації $I \text{ conf}_{\text{nt}} S$. Відношення виконується незважаючи на існування еволюції $e = (a_2 / b_1) \in \text{NTEvol}(I)$, оскільки $e \notin \text{NTEvol}(S)$.

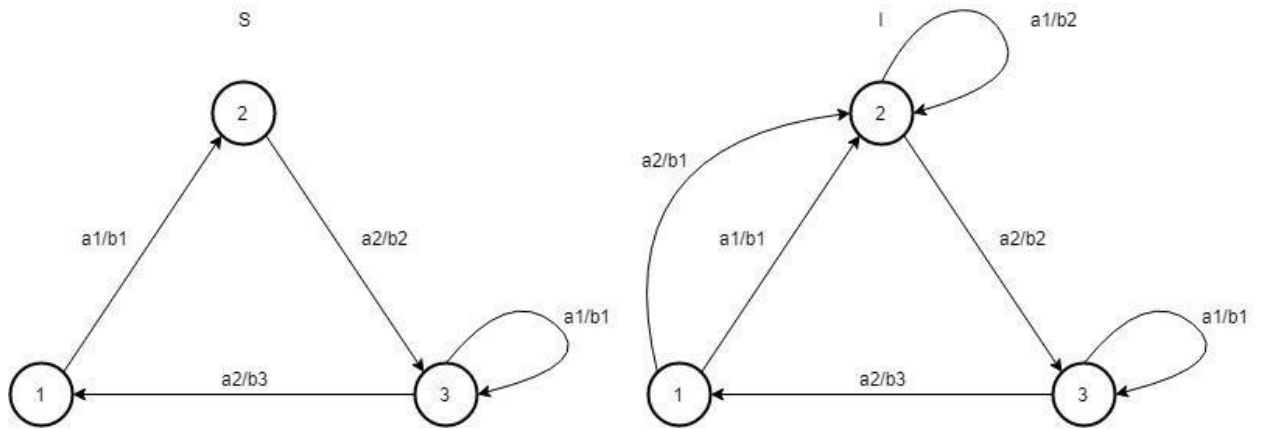


Рисунок 4.4 - Приклад виконання відношення реалізації $I \text{ conf}_{nt} S$

На рисунку 4.5 представлений приклад невиконання відношення реалізації $I \text{ conf}_{nt} S$. Відношення не виконується, оскільки $e = (a_1 / b_1, a_2 / b_1) \in \text{NTEvol}(S)$, але $e \notin \text{NTEvol}(I)$.

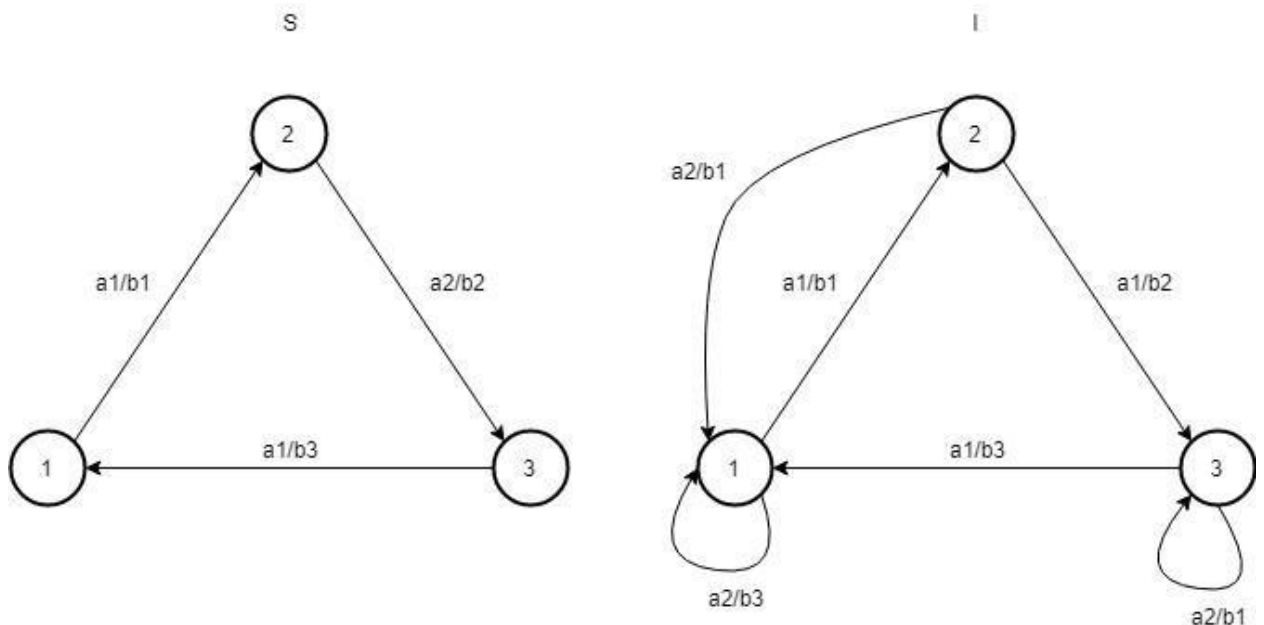


Рисунок 4.5 - Приклад невиконання відношення реалізації $I \text{ conf}_{nt} S$

Для ситуації, коли час виражається фіксованим значенням, може бути введено два відношення реалізації [6]. Відношення conf_a (завжди відповідає) має

на увазі, що для кожної часової еволюції (e, t) імплементації, якщо e є еволюцією моделі S , тоді (e, t) також є часовою еволюцією S . Це відношення означає, що реалізація імітує часову поведінку формальної моделі. Відношення conf_b (відповідає в кращому випадку) накладає додаткову умову: час кожної часової еволюції імплементації повинен бути менший, ніж час відповідної часової еволюції моделі. Це відношення реалізації висловлює, що реалізація швидше, ніж модель. Оскільки розглядається спостерігаємий автомат, існує не більше однієї еволюції, що задовольняє попереднім умовам.

Нехай S і I - два часових автомати.

$I \text{ conf}_a S$, якщо $I \text{ conf}_{nt} S$ і для всіх $e \in \text{NTEvol}(I) \cap \text{NTEvol}(S)$ для всіх значень часу $t \in \mathbb{R}_+$, таких що $(e, t) \in \text{TEvol}(I)$, виконується $(e, t) \in \text{TEvol}(S)$

$I \text{ conf}_b S$, якщо $I \text{ conf}_{nt} S$ і для всіх $e \in \text{NTEvol}(I) \cap \text{NTEvol}(S)$ для всіх значень часу $t \in \mathbb{R}_+$, таких що $(e, t) \in \text{TEvol}(I)$, виконується $(e, t') \in \text{TEvol}(S)$ і $t \leq t'$.

Розглянемо ситуацію, коли час виражається випадковою величиною. Може бути запропоновано відношення реалізації відповідності на основі еквівалентності функцій розподілу випадкових величин.

Нехай I й S - два часових автомати. Автомат I стохастично відповідає S , якщо $I \text{ conf}_{nt} S$ і для всіх $e \in \text{NTEvol}(I) \cap \text{NTEvol}(S)$ для всіх випадкових величин $\xi \in V$, таких що $(e, \xi) \in \text{TEvol}(I)$, виконується $(e, \xi') \in \text{TEvol}(S)$ і $\xi = \xi'$. Таке відношення позначається $I \text{ conf}_s S$.

Таким чином, відношення $I \text{ conf}_s S$ вимагає, щоб всі часові еволюції S , які можуть бути виконані імплементацією, мали однаково розподілені випадкові величини.

Проблема полягає в тому, що фактично знадобиться нескінченна кількість спостережень випадкової змінної з невідомою функцією розподілу, щоб гарантувати, що ця випадкова змінна розподілена як інша випадкова змінна з

відомим розподілом. Таким чином, необхідно ввести більш реалістичні відношення реалізації, засновані на кінцевому наборі спостережень.

Нехай I - це часовий автомат. Спостережуваний часовий запуск I - це пара $((i_1 / o_1, \dots, i_n / o_n), t)$, якщо спостереження I показує, що послідовність $(i_1 / o_1, \dots, i_n / o_n)$ виконується за час t .

У певному сенсі спостережуваний часовий запуск може розглядатися як екземпляр еволюції, яку може виконати реалізація.

Нехай Φ - це множина еволюцій, H - це мультимножина спостережуваних часових запусків, тоді функція $\text{sampling}_{(\Phi, H)}(e)$ - функція, яка для еволюції $e \in \Phi$ повертає вибірку-мультимножину значень часу, за яке еволюція була виконана.

Може бути запропонований наступний алгоритм застосування відношення відповідності.

1. Вибрати множину еволюцій Φ - підмножина $\text{NTEvol}(S)$.
2. Для кожної еволюції з Φ виконати деяке число раз запусків імплементації, відзначаючи час виконання. Занести дані в мультимножину H . Чим більше мультимножина H , тим точніше може бути зроблений висновок про відповідність імплементації моделі.
3. Вибрати значення параметра α , $0 \leq \alpha \leq 1$.
4. Вибрати γ - критерій для перевірки гіпотези про приналежність вибірки значень закону розподілу.
5. Для кожної еволюції $e \in \Phi$ застосувати критерій γ для перевірки гіпотези про приналежність вибірки $J = \text{sampling}_{(\Phi, H)}(e)$ закону розподілу ξ

відповідного даної еволюції в моделі S . I (α, H)-стохастично відповідає S , якщо для кожної такої еволюції $\gamma(\xi, \text{sampling}_{(H, \Phi)}(e)) > \alpha$.

Таке відношення позначається $I \text{conf}_S^{(\alpha, H)}$.

Розглянемо ситуацію, коли час виражається інтервалом. Може бути запропоновано відношення реалізації на основі еквівалентності часових інтервалів.

Нехай I й S - це два часові автомати. I відповідає S за часовими параметрами (conforms in time), якщо $I \text{conf}_{\text{int}} S$ і для всіх $e \in \text{NTEvol}(I) \cap \text{NTEvol}(S)$ для всіх часових інтервалів $\hat{t} \in \mathbb{I}_{R^+}$, таких що $(e, \hat{t}) \in \text{TEvol}(I)$, виконується $(e,) \hat{t} \in \text{TEvol}(S)$.

Таке відношення позначається $I \text{conf}_{\text{int}} S$.

Таке відношення є непрактичним, оскільки знадобиться нескінченна кількість спостережень, щоб гарантувати, що інтервал величини затримки імплементації відповідає інтервалу моделі. Щоб уникнути цієї проблеми застосовується метод, аналогічний ситуації з випадковими затримками, але замість використання критерію γ перевіряється приналежність спостережуваних величин вказаному інтервалу часу.

Нехай I й S - два часових автомати, Φ - це множина еволюцій, H - це мультимножина спостережуваних часових запусків. Для кожної еволюції $e \in \Phi$ інтервал значень вибірки e в H - це інтервал $\mathcal{S}_{(H, e)} = [\min(\text{sampling}_{(H, \Phi)}(e)), \max(\text{sampling}_{(H, \Phi)}(e))]$.

Вводиться наступне відношення реалізації.

I H -відповідає S за часовими параметрами (H -timely conforms), якщо $I \text{conf}_{\text{int}} S$ і для всіх $e \in \text{NTEvol}(I) \cap \text{NTEvol}(S)$ для всіх часових інтервалів $\hat{t} \in \mathbb{I}_{R^+}$, таких що $(e, \hat{t}) \in \text{TEvol}(S)$, виконується $\mathcal{S}_{(H, e)} \subseteq \hat{t}$. Таке відношення позначається $I \text{conf}_{\text{int}}^H S$.

Це відношення встановлює, що реалізація повинна відповідати

специфікації без урахування часових параметрів. Крім того, спостерігаємі значення часу виконання, відповідні еволюції, повинні належати інтервалу часу, зазначеному в специфікації для цієї еволюції.

В [6] пропонується алгоритм для генерації повного набору тестів для заданої формальної моделі, що дозволяє перевірити, чи відповідає імплементація введеним відношенням реалізації.

5 ЧАСОВИЙ РОЗШИРЕНИЙ АВТОМАТ

В [7] розвиваються ідеї, пов'язані з моделлю часового автомата, запропоновані в [6].

Пропонується розглядати два часових аспекти функціонування системи. По-перше, на виконання операцій системою йде час. Цей час може залежати від значень певних параметрів системи, таких як доступні ресурси. По-друге, якщо протягом деякого часу очікування не надходить зовнішніх впливів, то система може змінити свій внутрішній стан.

Нехай D_1, \dots, D_m - множини значень, $D = D_1 \times D_2 \times \dots \times D_m$. Часовий розширений автомат (timed extended finite state machine) - це кортеж $M = (S, I, O, TO, Tr, s_{in}, \bar{y})$, який включає наступні елементи:

- S - кінцева множина станів,
- I - множина вхідних дій,
- O - множина вихідних дій,

- $TO: S \rightarrow S \times (\text{Time} \cup \infty)$ - функція часу очікування,
- Tr - множина переходів за впливом,
- s_{in} - початковий стан,
- $\bar{y} \in D$ - множина початкових значень змінних.

Перехід за впливом - це кортеж (s, s', i, o, Q, Z, C) :

- $s \in S$ - початковий стан переходу,
- $s' \in S$ - кінцевий стан переходу,
- $i \in I$ - вхідна дія, пов'язана з переходом,
- $o \in O$ - вихідна дія, пов'язана з переходом,
- $Q: D \rightarrow \text{Bool}$ - це предикат на множині змінних,
- $Z: D \rightarrow D$ - перетворення поточних змінних,
- $C: D \rightarrow \text{Time}$ - час, необхідний для завершення переходу.

Конфігурація часового розширеного автомата - це пара (s, x) , де $s \in S$ - поточний стан, а $x \in D$ - кортеж, що містить поточні значення змінних.

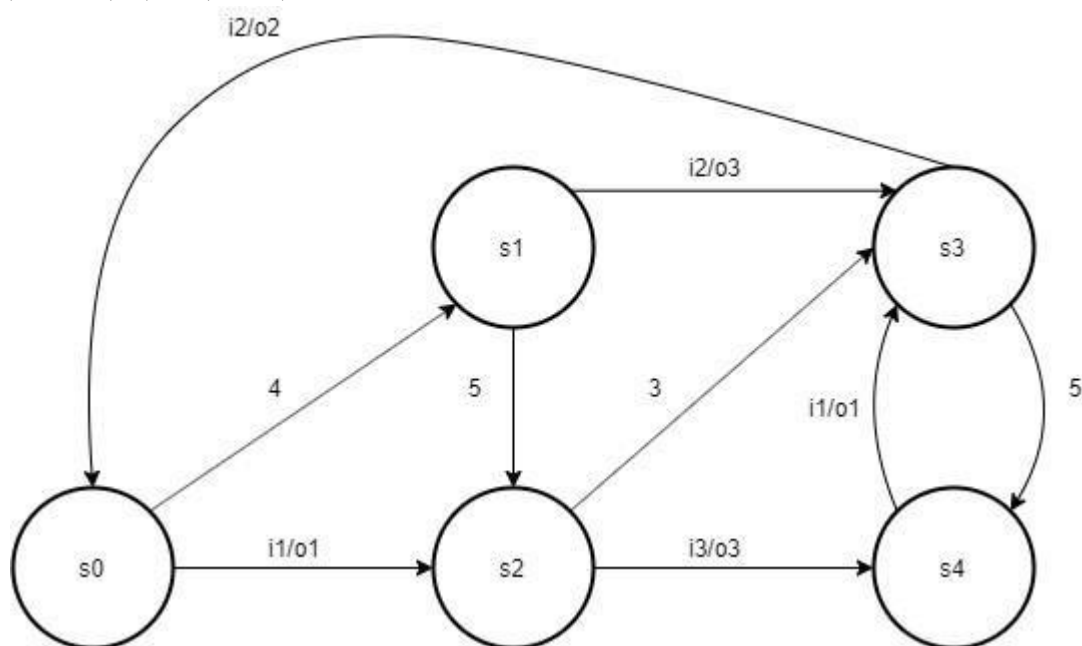
Часовий розширений автомат є повністю визначеним, якщо для всіх станів $s \in S$ і вхідних дій $i \in I$ існують s', o, Q, Z, C , такі що $(s, s', i, o, Q, Z, C) \in Tr$.

При заданій конфігурації (s, \bar{x}) перехід за дією (s, s', i, o, Q, Z, C) означає, що якщо надходить вхідна дія і виконується $Q(\bar{x})$, то після $C(\bar{x})$ одиниць часу буде згенеровано вихідна дія o і конфігурація стане $(s', Z(\bar{x}))$.

Для кожного стану $s \in S$ застосування функції таймаута $TO(s)$ повертає пару (s', t) , що вказує час, протягом якого автомат може залишатися в стані s , чекаючи вхідний вплив, і стан, в який автомат перейде по закінченню цього періоду. Мається на увазі, що $s \neq s'$, тобто закінчення таймаута завжди призводить до зміни

стану. Відсутність таймаута в даному стані вказується установкою t в ∞ . Приклад часового розширеного автомата представлений на рисунку 5.1.

Переходи за дією даного автомата: $t_{02} = (s_0, s_2, i_1, o_1, Q_0, Z_0, C_0)$, $t_{13} = (s_1, s_3, i_2, o_3, Q_1, Z_1, C_1)$, $t_{24} = (s_2, s_4, i_3, o_3, Q_2, Z_2, C_2)$, $t_{30} = (s_3, s_0, i_2, o_2, Q_3, Z_3, C_3)$, $t_{43} = (s_4, s_3, i_1, o_1, Q_4, Z_4, C_4)$. Таймаут для станів: $TO(s_0) = (s_1, 4)$, $TO(s_1) = (s_2, 5)$, $TO(s_2) = (s_3, 3)$, $TO(s_3) = (s_4, 5)$.



$$Q_i(x) = \begin{cases} (x_1 > 0) \wedge (x_2 > 0) \wedge (x_3 > 0), & i \in \{0, 2, 4\} \\ true, & i \in \{1, 3\} \end{cases}$$

$$Z_i(x) = x + \begin{cases} (5, 10, 12), & i = 3 \\ (-1, -2, -3), & i \in \{0, 2, 4\} \\ (1, 0, 3), & i = 1 \end{cases}$$

$$C_i(x) = \begin{cases} 2, & x_1 > 8 \\ 8, & x_1 < 2 \\ 10 - x_1, & x_1 \in [2, 8] \end{cases}$$

Рисунок 5.1 - Приклад часового розширеного автомата

Нехай $M = (S, I, O, TO, Tr, s_{in}, \bar{x})$ - часовий розширений автомат, а $c_0 = (s_0, \bar{x}_0)$ - конфігурація часового розширеного автомата. Кортеж $(s_0, s, i / o, \hat{t}, t_0, \bar{v})$ є кроком автомата M для конфігурації c_0 , якщо існує $k \geq 0$ станів $s_1, \dots, s_k \in S$, таких що для всіх $1 \leq j \leq k$ виконується $TO(s_{j-1}) = (s_j, t_j)$ і існує перехід $(s_k, s, i, o, Q, Z, C) \in Tr$

такий, що $Z(\bar{x}_0) = \bar{v}$, $T_0 = \sum_{j=1}^k t_j$, $\hat{t} = [T_0, T_0 + t_k]$, $t_0 = C(\bar{x}_0)$ і $Q(\bar{x}_0)$ виконується. $Steps(M, s, \bar{x})$ - це множина кроків часового розширеного автомата M для конфігурації (s, \bar{x}) .

Прикладом конфігурації для часового розширеного автомата на рисунку 5.1 може бути $(s_0, (15,15,15))$. Прикладом кроку автомата тоді може бути $(s_0, s_2, i_1 / o_1, [0, 4), 2, (14,13,12))$. Такий крок відповідає переходу t_{02} у відповідь на вхідну дію i_1 , що відбувся до закінчення часу очікування стану s_0 . На генерацію вихідної дії o_1 пішло $C_0((15,15,15)) = 2$ часові одиниці. Кортеж змінних після даного кроку $(14,13,12) = Z_0((15,15,15))$. Крок $(s_0, s_3, i_2 / o_3, [4, 9), 2, (16,15,18))$ відповідає ситуації, в якій після очікування вхідної дії протягом 4 одиниць часу автомат перейшов в стан s_1 . Після чого, був виконаний перехід t_{13} у відповідь на вхідну дію i_2 , що відбулася до закінчення часу очікування стану s_1 . На генерацію вихідної дії o_3 , відповідного переходу t_{13} , пішло $C_1((15,15,15)) = 2$ часові одиниці. Кортеж змінних після виконання кроку $(16,15,18) = Z_1((15,15,15))$.

Часова еволюція розширеного часового автомата M є кортеж $(\hat{t}_1 / i_1 / t_{o1} / o_1, \dots, \hat{t}_r / i_r / t_{or} / o_r)$, якщо існує r кроків $M(s_{in}, s_1, i_1 / o_1, \hat{t}_1, t_{o1}, \bar{y}_1), \dots, (s_{r-1}, s_r, i_r / o_r, \hat{t}_r, t_{or}, \bar{y}_r)$ для конфігурацій $(s_{in}, \bar{y}_1), \dots, (s_{r-1}, \bar{y}_{r-1})$ відповідно.

$TEvol(M)$ - множина часових еволюцій M . Кортеж $(\hat{t}_1 / i_1 / o_1, \dots, \hat{t}_r / i_r / o_r)$ - це функціональна еволюція M , $FEvol(M)$ - це множина функціональних еволюцій M .

Прикладом часової еволюції, що складається з трьох кроків, для часового розширеного автомата на рисунку 5.1 може бути $([9, 12), i_3 / 2 / o_3, [0, \infty), i_1 / 2 / o_1, [5, \infty), i_1 / 2 / o_1)$. Конфігурація автомата після першого кроку часової еволюції - $(s_4, (14, 13, 12))$, після другого кроку - $(s_3, (13, 11, 9))$, після третього кроку - $(s_3, (12, 9, 6))$.

Нехай $M = (S, I, O, TO, Tr, s_{in}, \bar{y})$ - часовий розширений автомат, розглянемо часову еволюцію $e = (\hat{t}_1 / i_1 / t_{o1} / o_1, \dots, \hat{t}_r / i_r / t_{or} / o_r)$. Кортеж $(t_1 / i_1 / t_{o1} / o_1, \dots, t_r / i_r / t_{or} / o_r)$ є екземпляром часової еволюції e (instanced timed evolution), якщо для всіх $1 \leq j \leq r$ $t_j \in \hat{t}_j$. Кортеж $(t_1 / i_1 / o_1, \dots, t_r / i_r / o_r)$ є екземпляром функціональної еволюції e . $InsTEvol(M)$ - це множина екземплярів часових еволюцій M , $InsFEvol(M)$ - множина екземплярів функціональних еволюцій.

Прикладами екземплярів часової еволюції $([9, 12), i_3 / 2 / o_3, [0, \infty), i_1 / 2 / o_1, [5, \infty), i_1 / 2 / o_1)$ можуть бути $(10, i_3 / 2 / o_3, 14, i_1 / 2 / o_1, 11, i_1 / 2 / o_1)$ і $(11, i_3 / 2 / o_3, 0, i_1 / 2 / o_1, 6, i_1 / 2 / o_1)$.

6 ЧАСОВИЙ АВТОМАТ З ТАЙМАУТАМИ І ЧАСОВИМИ ОБМЕЖЕННЯМИ

В [8] пропонується модель часового автомата з таймаутами і часовими обмеженнями (FSM with timed guards and timeouts), близька до моделі, представленої в [7]. Основна відмінність між моделями полягає в наступному. Часовий розширений автомат забезпечений вектором змінних, які використовуються для визначення часу обробки вхідної дії і можливості виконання переходу. Часовий автомат з таймаутами і часовими обмеженнями не містить вектора змінних, час обробки вхідного символу фіксований і залежить тільки від переходу, можливість виконання переходу залежить від значення часової змінної, а не від значень елементів вектора змінних.

Формально часовий автомат з таймаутами і часовими обмеженнями - це кортеж $S = (I, S, O, \lambda_S, \Delta_S)$:

- I - вхідний алфавіт,
- O - вихідний алфавіт,
- S - кінцева непорожня множина станів,
- $\lambda_S \subseteq (S \times I \times O \times S \times \Pi \times Z)$ - відношення переходів, - Δ_S - функція таймаута.

Π - це множина часових обмежень, Z - це множина вихідних затримок, які є невід'ємними цілими числами. Функція таймаута $\Delta_S: S \rightarrow S \times (N \cup \{\infty\})$, де N - це множина невід'ємних цілих чисел; для кожного стану функція таймаута вказує максимальний час очікування вхідного символу.

Часове обмеження $g \in \Pi$ описує область значень часової змінної, при яких перехід може бути виконаний, g задається у вигляді інтервалу з $[0, T)$, T - значення (вхідного) таймаута в поточному стані. Вихідна затримка визначає, скільки часу займає генерація вихідного символу після отримання вхідного символу. Перехід

(s, i, o, s', g, d) означає, що часовий автомат S , що знаходячись в стані s , приймає вхідний символ i в момент часу $t \in g$, яке обчислюється з моменту, коли часовий автомат S увійшов в стан s . Після отримання вхідного символу i таймер встановлюється в нуль, і S генерує вихідний символ o через d одиниць часу, підрахованих з моменту, коли вхідний символ i був отриманий. Потім таймер скидаються в нуль, щоб почати вимірювання затримки наступного часового вхідного символу.

Нехай стан s часового автомата такий, що $\Delta_s(s) = (s', T)$; якщо до закінчення часу T вхідні символи не надходять, то часовий автомат переходить в стан s' і таймер встановлюється в нуль.

Для кінцевого алфавіту A часовим символом є пара (a, t) , де $t \in \mathbb{R}$ - затримка символу $a \in A$. Часова послідовність - це кінцева послідовність $(a_1, t_1), (a_2, t_2), (a_3, t_3), \dots$ з часових символів, де затримки t_i є невід'ємними значеннями.

Часовий автомат є повністю визначеним, якщо для кожного стану s і часового вхідного символу (i, t) існує хоча б один перехід $(s, i, o, s', g, d) \in \lambda_s$ такий, що $t \in g$. Автомат є детермінованим, якщо існує не більше одного такого переходу. У повністю визначеному і детермінованому автоматі в заданий момент часу існує рівно один активний перехід для кожного вхідного символу.

Приклад часового автомата з таймаутами і часовими обмеженнями представлений на рисунку 6.1.

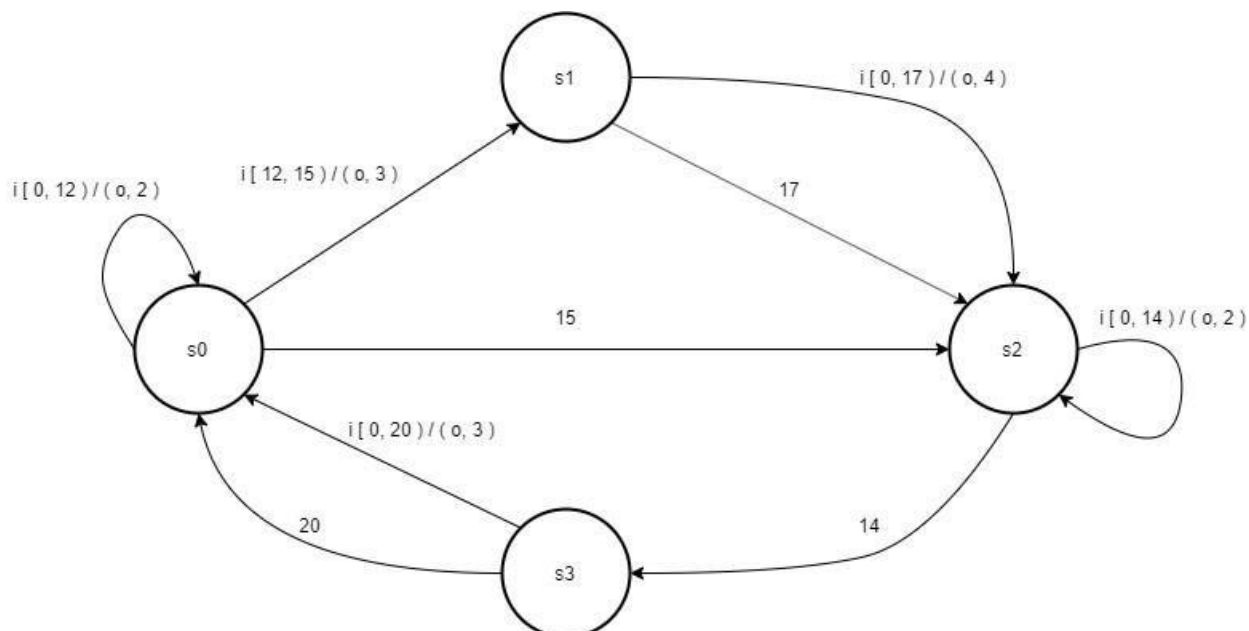


Рисунок 6.1 - Приклад часового автомата з таймаутами і часовими обмеженнями

У загальному випадку, запропонована модель містить три типи часових параметрів: таймаут, часове обмеження і час обробки вхідного символу. При цьому, можуть розглядатися часові автомати з меншим числом параметрів. Так, в [9] розглядаються автомати без затримки на формування вихідного символу, в [10] розглядаються часові автомати, що не мають часових обмежень для вхідних символів.

Поведінка часового автомата може бути описана з використанням стандартного кінцевого автомата, який називається кінцевою автоматною абстракцією [8].

В [8, 11, 12, 13] розглядаються завдання мінімізації часового автомата, перевірки еквівалентності часових автоматів, синтезу перевіряючих тестів. Кінцева автоматна абстракція - це допоміжний засіб, що використовується для вирішення цих завдань.

Розглянемо побудову кінцевої автоматної абстракції для часового автомата з таймаутом.

Часовий автомат з таймаутом є кортеж $M = (S, I, O, \lambda_S, s_0, \Delta_S)$:

- S - кінцева непорожня множина станів,
- I - кінцева непорожня множина вхідних символів,
- O - кінцева непорожня множина вихідних символів,
- s_0 - початковий стан,
- $\lambda_S \subseteq S \times I \times O \times S$ - відношення переходів,
- $\Delta_S: S \rightarrow S \times (\mathbb{N} \geq 1 \cup \{\infty\})$ - функція часу очікування [9].

Нехай $\Delta_S(s) = (s', x)$, тоді позначимо $\Delta_S(s)_1 = s'$, $\Delta_S(s)_2 = x$. Переходи часового автомата з таймаутом можуть бути ініційовані при отриманні вхідного символу і за таймаутом. Коли автомат переходить в стан s , таймер скидається в 0. Якщо вхідний символ і надходить до закінчення часу очікування $\Delta_S(s)_2$, і існує перехід $(s, i, o, s') \in S \times I \times O \times S$, то автомат видає вихідний символ o , переходить в стан s' , скидаючи таймер в 0. Якщо вхідний символ отримано до закінчення часу очікування $\Delta_S(s)_2$, то часовий автомат перейде в стан $\Delta_S(s)_1$, скидаючи таймер в 0. Якщо $\Delta_S(s)_2 = \infty$, то автомат може залишатися в стані s нескінченно довго, чекаючи введення. Станом часового автомата з таймаутом є пара $(s, x) \in S \times \mathbb{R}^+$ з обмеженням, що $x < \Delta_S(s)_2$.

Нехай M - часовий автомат з таймаутом, постійна $\max(M)$ визначає найбільше значення функції таймаута Δ_S , відмінне від ∞ . Стани кінцевої автоматної абстракції будуть парами (s, n) , де s - це стан M , а n - натуральне число в діапазоні від 0 до $\max(M) - 1$, що абстрагує значення таймера. Переходи можуть бути або стандартними переходами у відповідь на вхідний символ, поміченими

парами з $I \times O$, або переходами по закінченню часу, поміченими спеціальною парою $(1, 1)$, що представляє затримку в одиницю часу.

Нехай $M = (S, I, O, \lambda_S, s_0, \Delta_S)$ - часовий автомат з таймаутом, $N = \max(M)$ -
 1. Кінцева автоматна абстракція $A_M = (S \times \{0, \dots, N\}, I \cup \{1\}, O \cup \{1\}, \lambda_A, (s_0, 0))$ -
 це кінцевий автомат з наступними переходами:

- з (s, x) до $(s', 0)$, позначений i/o , якщо $(s, i, o, s') \in \lambda_S$,
- з $(s, 0)$ до $(s, 0)$, позначений $1/1$, якщо $\Delta_S(s)_2 = \infty$,
- з (s, x) до $(s, x + 1)$, позначений $1/1$, якщо $x + 1 < \Delta_S(s)_2 < \infty$,
- з (s, x) до $(s', 0)$, позначений $1/1$, якщо $\Delta_S(s) = (s', n)$ і $n = x + 1$ [9]. На
 рисунку 6.2 представлений часовий автомат з таймаутом.

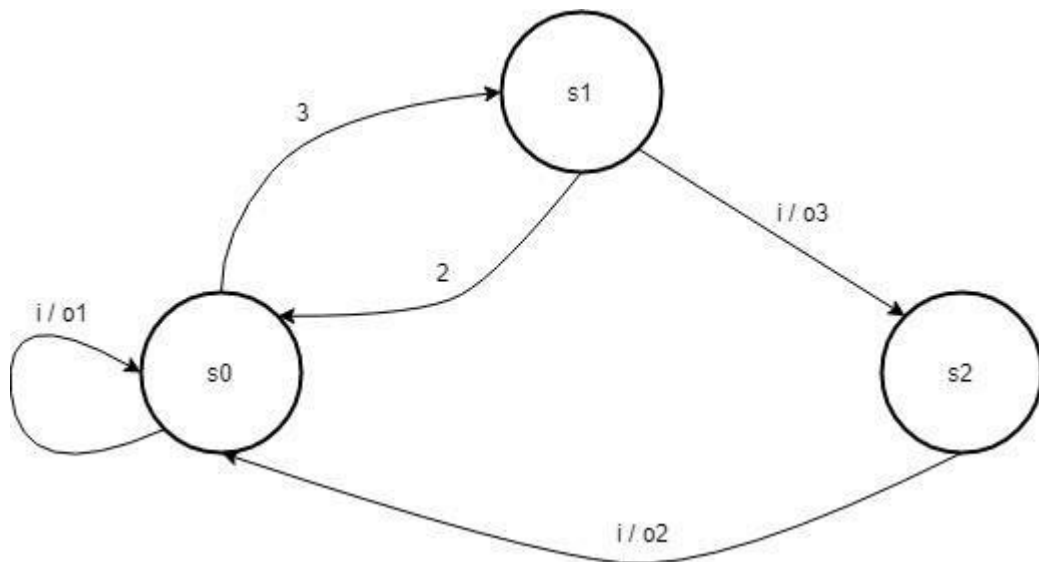


Рисунок 6.2 - Часовий автомат з таймаутом

В процесі побудови кінцевої автоматної абстракції A_M для автомата на рисунку 6.2 стан s_0 часового автомата перетворюється в стани $(s_0, 0)$, $(s_0, 1)$, $(s_0, 2)$ кінцевого автомата. Крім того, A_M має переходи зі стану $(s_0, 0)$ в $(s_0, 1)$ і зі стану $(s_0, 1)$ в $(s_0, 2)$, помічені парою $(1, 1)$. Оскільки часовий автомат має перехід зі стану s_1 в s_2 за вхідним символом i , то кінцева автоматна абстракція має переходи в стан $(s_2, 0)$ за вхідним символом i зі станів $(s_1, 0)$, $(s_1, 1)$. Оскільки в

часовому автоматі $\Delta_S(s2)_2 = \infty$, то A_M містить перехід-петлю зі стану $(s2, 0)$, позначену парою $(1, 1)$. Так в часовому автоматі $\Delta_S(s0) = (s1, 3)$, то A_M містить перехід зі стану $(s0, 2)$ в $(s1, 0)$, позначений $(1, 1)$.

На рисунку 6.3 представлена побудована кінцева автоматна абстракція.

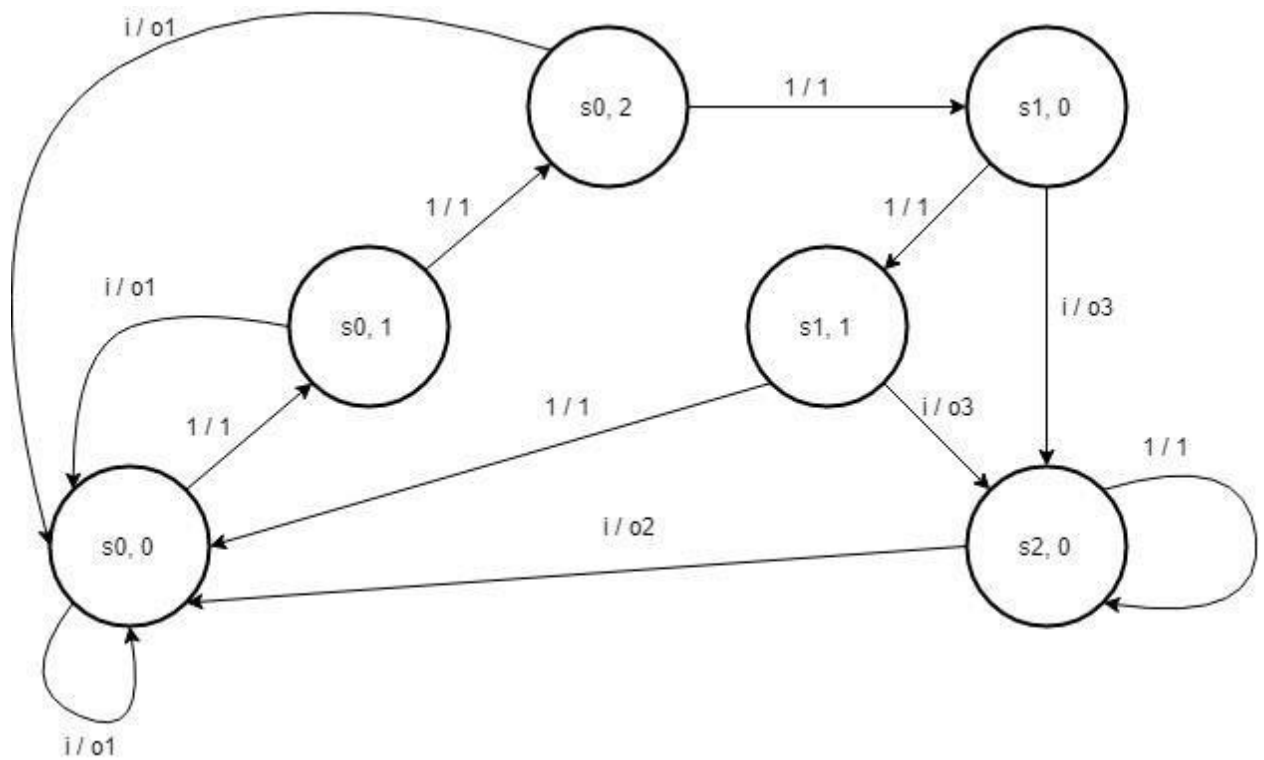


Рисунок 6.3 - Кінцева автоматна абстракція

7 СТРУКТУРНИЙ ЧАСОВИЙ АВТОМАТ МУРА

В [14] запропонована повна модель структурного часового автомата, яка використовується у даній роботі.

Для опису поведінки систем реального часу пропонується використовувати модель часового автомата з єдиною часовою змінною, як в моделях в [6, 8].

Запропонована модель структурного часового автомата явно розділяє вхідні сигнали на вхідні дії та події. В [8], по суті, використовуються тільки події. Запропонована модель розділяє вихідні сигнали на керуючі сигнали і діяльності. В [8] передбачається, що автомат генерує тільки керуючі сигнали.

Поділ вхідних сигналів на вхідні дії та події, а також поділ вихідних сигналів на керуючі сигнали й діяльності виконується подібно підходу, використовуваному в [1].

Для опису часових аспектів функціонування системи управління реального часу пропонується використовувати 3 параметри:

- вхідний таймаут,
- часове обмеження, - вихідну затримку.

Вхідний таймаут визначає максимальний час очікування зовнішньої події для кожного стану автомата. Якщо подія не відбулася до закінчення часу очікування, то автомат починає опитування вхідних змінних і може перейти в інший стан. Часові обмеження є інтервалами на переходах, які обмежують час, протягом якого перехід може бути виконаний. Вихідні затримки відображають час, який витрачається автоматом на виконання переходу, тобто вихідний сигнал з'являється на виході через інтервал часу, який визначається вихідною затримкою.

Розглянемо відмінності в розумінні вхідних таймаутів в запропонованій моделі та в [8].

В [8] вхідні таймаути асоційовані з кожним станом, у запропонованій моделі вхідні таймаути асоційовані з дугами. Таким чином, при переході в поточний стан

час перебування в ньому залежить від вхідної дуги, по якій був виконаний перехід.

В [8] по закінченню таймаута виконується безумовний перехід в певний стан. У запропонованій моделі по закінченню таймаута виконується опитування вхідних впливів, значення яких визначають стан, в який переходить автомат.

Якщо в якості моделі керуючого автомата розглядати модель Мура, то повна модель структурного часового автомата, яка складається з трьох параметрів $\langle t_c, t_{to}, t_d \rangle$ може бути спрощена.

В повній моделі вхідні timeout ставляться у відповідність кожній вхідній дузі. Але в автоматі Мура при рівності всіх вхідних timeout для певного стану автомату цю затримку можна внести до самого стану і вважати затримкою в стані.

При такій моделі вхідного timeout часові обмеження (t_c , input constraint) для кожної дуги для прийняття зовнішніх подій мають знаходитись в проміжку часу timeout. При цьому слід враховувати, що наявність події в проміжку часу t_c може привести до зміни стану, незважаючи на те, що t_{to} для даного стану ще не закінчився.

Вихідні затримки для всіх сигналів певного стану можуть бути різними, але при цьому має враховуватися максимальне t_c для даного стану. Інтервал прийому зовнішніх подій (input constraint) може починатися не раніше, ніж ініціюються всі вихідні сигнали з урахуванням t_{di} . Усі вихідні сигнали мають бути ініційовані до можливості зміни стану за зовнішньою подією.

Запропонована повна модель структурного часового автомата може бути представлена дев'яткою $W = (X, Y, Z, g, f, z_0, T_c, T_{to}, T_d)$:

– $X = \{X_C, X_E\}$ - множина вхідних змінних, X_C – множина

оповіщувальних сигналів від об'єкта керування, X_E – множина зовнішніх подій;

– $Y = \{Y_C, Y_F\}$ - множина вихідних змінних, Y_C – множина керуючих

сигналів, Y_F – множина діяльностей;

– Z – множина внутрішніх змінних, які визначають кодування станів автомата;

– f - функція переходів;

- g – функція виходів;

– z_0 - код початкового стану автомата;

– $T_c = \{t_{c1}, t_{c2}, \dots, t_{cp}\}$ - множина часових інтервалів для обмежень на кожній дузі графу переходів, де p – кількість дуг у графі переходів, $t_{ci} = [t_{cil}, t_{cih}]$, $t_{cil} = \{1, k\}$, $t_{cih} = \{1, k\}$, k – максимальна кількість тактів обмежень на переходах до i -ї

вершини графу переходів в режимі опитування, $k = \{1, \infty\}$, ∞ – відповідає виключно подієвій функції переходів;

– $T_{to} = \{t_{to1}, t_{to2}, \dots, t_{ton}\}$ - множина часових змінних для timeouts кожного стану автомата, $t_{toi} = \{1, n\}$ – timeout для кожного стану, n – кількість станів автомата;

– $T_d = \{t_{d1}, t_{d2}, \dots, t_{dm}\}$ - множина затримок для реалізації відповідного вихідного сигналу, де m – кількість вихідних змінних, $t_{di} = \{1, l\}$, де l – максимальна кількість тактів для реалізації функцій виходу в зазначеному стані автомата [14].

У запропонованій моделі часового автомата Мура логіка його роботи виглядає наступним чином.

При переході автомата у поточний стан a_i для нього визначається основний часовий параметр $t_{to}(a_i)$ (timeout), тобто час протягом якого автомат має знаходитися у поточному стані, якщо зовнішня подія достроково не переведе автомат в інший стан. t_{to} визначається в автоматних тактах. Після закінчення часу t_{to} автомат опитує вхідні сигнали та переходить у наступний стан. В класичній моделі Мура для усіх станів $t_{to} = 1$.

Вихідні сигнали автомата у поточному стані a_i з'являються на виходах автомата у час, який визначається $t_{dj}(a_i)$ (output delays, тобто вихідними затримками для сигналів u_j у стані a_i). Для кожного з вихідних сигналів u_j вихідна затримка визначається в автоматних тактах. Вихідна затримка задає початок та кінець вихідного сигналу, $t_d = t_{d0} + t_{\Delta}$, де t_{d0} – початок вихідного сигналу, а t_{Δ} – довжина вихідного сигналу. При $t_{\Delta} = \infty$ вихідний сигнал тримається до його зміни у наступному стані. В класичній моделі Мура для всіх вихідних сигналів $t_d = t_{d0} + t_{\Delta}$, де $t_{d0} = 0$, $t_{\Delta} = \infty$.

Обробка зовнішніх подій здійснюється наступним чином. Для кожного стану a_i задаються часові обмеження $t_c(a_i)$ (input constraints), тобто проміжок часу, протягом якого автомат, знаходячись у стані a_i , може обробляти зовнішні події. Часове обмеження визначається в автоматних тактах та обчислюється як $t_c = [c_1, c_h]$, де c_1 – початок інтервалу часових обмежень, c_h – кінець інтервалу часових обмежень. Автомат, знаходячись у стані a_i , обробляє подію z_p у початку $c_1, c_{1+1}, c_{1+2}, \dots, c_h$ тактів. Якщо $t_c = \emptyset$, то обробка події z_p заборонена у стані a_i . Якщо $t_c = [1, \infty]$, то автомат, знаходячись у стані a_i , обробляє подію z_p протягом усього таймауту стану a_i .

Таким чином, обробка зовнішніх подій відбувається в момент активного фронту синхросигналу. На рисунку 7.1 представлені моменти обробки події Btn.

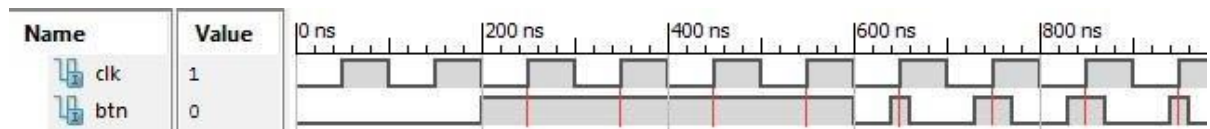


Рисунок 7.1 - Обробка події Btn

8 АПАРАТНИЙ ПІДХІД ДО РЕАЛІЗАЦІЇ СИСТЕМ УПРАВЛІННЯ РЕАЛЬНОГО ЧАСУ

Можна виділити два способи реалізації локального цифрового пристрою, призначеного для обробки інформації або управління:

- апаратний,
- програмно-апаратний [15].

При апаратному підході алгоритм функціонування пристрою описується мовою опису апаратури. Після чого, HDL-опис може бути синтезовано в ПЛІС (програмована логічна інтегральна схема) або ASIC (інтегральна схема спеціального призначення). При синтезі зазвичай використовуються засоби систем автоматизованого проектування.

Переваги апаратного підходу:

- досить велика швидкодія, - апаратна гнучкість.

Недоліками даного способу реалізації є відносна складність в реалізації часових характеристик системи і необхідність проектування інтерфейсу вводу/виводу для даної системи.

При програмно-апаратному підході пристрій, як правило, реалізується на мікроконтролері. Алгоритм функціонування реалізованого пристрою описується на апаратно-орієнтованій мові програмування.

Переваги програмно-апаратного підходу:

- наявність апаратно реалізованих інтерфейсів вводу-виводу,
- наявність спеціальних апаратно-орієнтованих засобів, що значно спрощують реалізацію часових характеристик.

До недоліків даного підходу відносяться менша гнучкість, пов'язана з архітектурою апаратури, і менша швидкодія.

Розглянемо особливості апаратного способу реалізації систем логічного управління на основі запропонованої моделі часового автомата Мура.

Пристрої логічного управління, побудовані на основі кінцевих автоматів, функціонують в автоматному часу, що вимірюється в автоматних тактах, тобто дискретних відрізках часу, за який автомат переходить з одного стану в інший. Тривалість автоматного такту в реальних пристроях, як правило, визначається частотою синхросигналу. Але, з іншого боку, пристрої логічного управління є пристроями реального часу і перехід з одного керуючого стану в інший визначається часовими параметрами алгоритму функціонування пристрою [14].

Граф переходів автомата, який враховує часові параметри називають темпоральний граф переходів (temporal state diagram). У цьому графі для кожного стану відображається таймаут T_i , який відповідає цьому стану. Вихідні сигнали, відповідні даному стану автомата, протягом цього часу не змінюються.

Для реалізації автомата з єдиною часовою змінною на програмованій логічній інтегральній схемі пропонується використовувати додатковий лічильник.

Зазначений лічильник використовується для підрахунку числа автоматних тактів, протягом яких автомат перебуває в деякому стані. Коли часовий автомат виконує перехід, значення даного лічильника скидається в 0.

Розглянемо підхід до реалізації 3 часових параметрів:

- вхідних таймаутів, - часових обмежень,
- вихідних затримок.

Таймаут T_i здійснюється багаторазовим переходом зі стану в цей самий стан, при цьому число переходів визначається числом тактів таймаута. Значення лічильника порівнюється з $T_i - 1$, оскільки при переході в стан a_i автомат один такт знаходиться в ньому до перевірки count, і щоб таймаут був точно T_i тактів, необхідно ще $T_i - 1$ тактів повторення. Тобто в темпоральному графі таймаут станів реалізується за допомогою петель, умовами для яких є перевірки значення лічильника. Така петля представлена на рисунку 8.1.

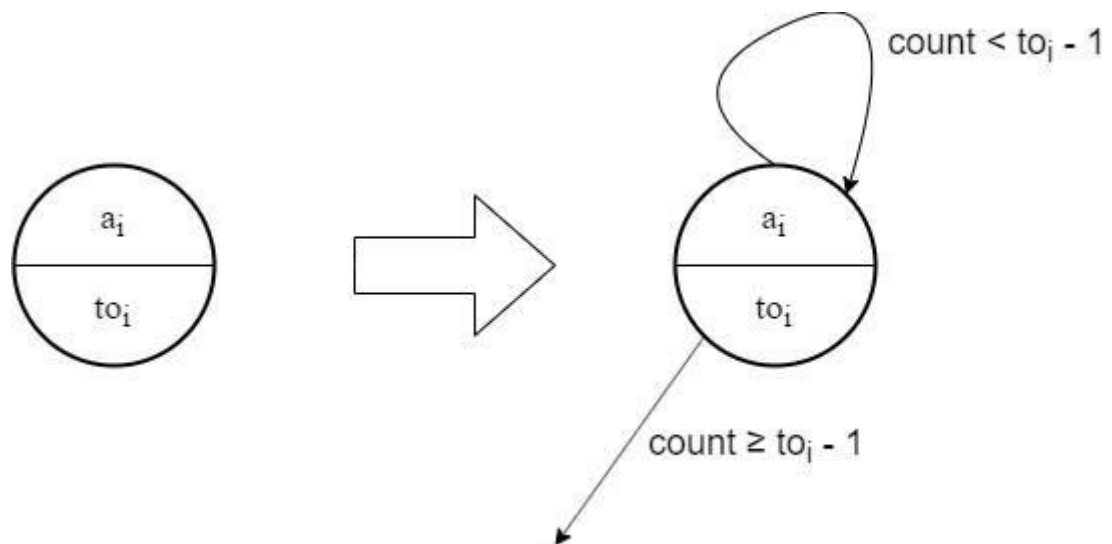


Рисунок 8.1 - Реалізація вхідних таймаутів

Часове обмеження виражається як $t_c = [c_l, c_h]$. Для реалізації часового обмеження необхідно порівняння значення, що зберігається в лічильнику, з нижньою і верхньою межею відповідного часового обмеження. Розглянемо нижню межу часового обмеження c_l . Значення лічильника порівнюється з c_l-1 , оскільки при переході в стан a_i автомат один такт знаходиться в ньому до перевірки count, і щоб нижня межа була точно c_l тактів, необхідно ще c_l-1 тактів перебування автомата в даному стані. Аналогічно, в останній такт, в якому дозволена обробка події, значення лічильника порівнюється з c_h-1 .

На рисунку 8.2 представлені умови на дузі, необхідні для реалізації часового обмеження.

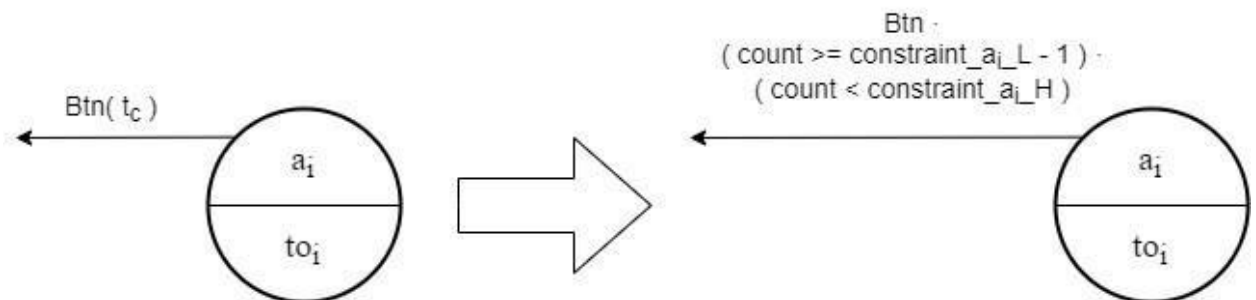


Рисунок 8.2 - Реалізація часових обмежень

Для апаратної реалізації вихідних затримок необхідно, щоб значення вихідних сигналів автомата залежали не тільки від поточного стану, а й від поточного значення лічильника.

9 ПОСТАНОВКА ТЕХНІЧНОГО ЗАВДАННЯ

Технічне завдання полягає в розробці VHDL-моделей системи управління дорожнім світлофором з використанням САПР на основі моделі часового автомата, що можуть бути синтезовані в програмовану логічну інтегральну схему.

Розглянемо алгоритм функціонування дорожнього світлофора, який працює у двох режимах: денному і нічному.

Множина вхідних сигналів $X = \{Onn, St, Btn\}$:

- $On = \{0, 1\}$ - сигнал включення дорожнього світлофора,
- $St = \{0, 1\}$ - сигнал запуску денного циклу роботи світлофора,
- $Btn = \{0, 1\}$ - зовнішня подія для включення зеленого світла на пішохідному переході.

Таким чином, On і St є оповіщувальними сигналами, а Btn - подією.

Множина вихідних сигналів $Y = \{R1, YRG, YGR, G1, R2, G2\}$:

- $R1$ - сигнал включення червоного світла світлофора на основній дорозі;
- YRG, YGR - сигнали включення жовтого світла світлофора на основній дорозі;
- $G1$ - сигнал включення зеленого світла світлофора на основній дорозі;
- $R2$ - сигнал включення червоного світла світлофора на пішохідному переході;

- G2 - сигнал включення зеленого світла світлофора на пішохідному переході.

Інтерфейс системи управління дорожнім світлофором представлений на рисунку 9.1.

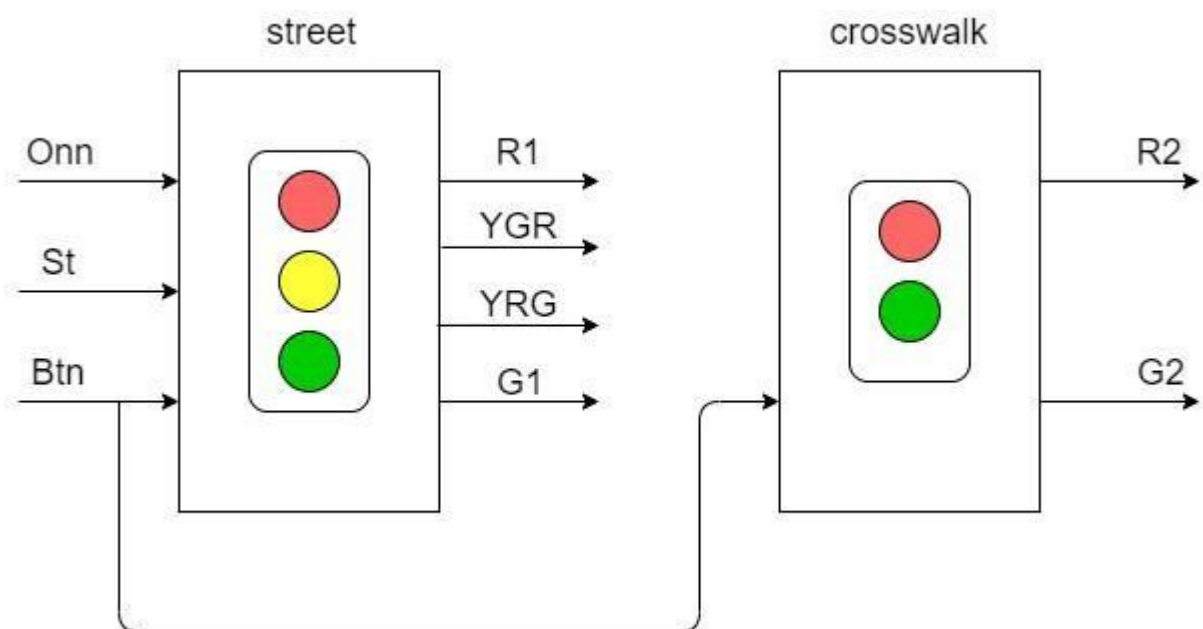


Рисунок 9.1 - Інтерфейс системи управління дорожнім світлофором

Необхідно розробити пристрій управління дорожнім світлофором, який функціонує у двох режимах.

1. У нічному режимі роботи світлофора відбувається миготіння жовтого світла на основній дорозі з інтервалом t_{01} , світлофор на пішохідному переході не працює.

2. У денному циклі роботи світлофора пристрій управління видає такі набори вихідних сигналів:

- { YGR, R1, R2 } - на пішохідному переході горить червоний, на

основній дорозі - червоний і жовтий;

- {R1, G2} - на пішохідному переході горить зелений, на основній дорозі - червоний;

- {YRG, R1, R2} - на пішохідному переході горить червоний, на основній дорозі - червоний і жовтий;

- {R2, G1} - на пішохідному переході горить червоний, на основній дорозі - зелений;

- {YGR, R1, R2} - на пішохідному переході горить червоний, на основній дорозі - червоний і жовтий.

При цьому час видачі {YRG, R1, R2} - t_{02} , час видачі {R1, G2} або {R2, G1}

- t_{03} .

Для стану a_5 , коли на основній дорозі горить зелений, визначено вікно прийому t_c зовнішньої події V_{tn} . При обробці даної події система управління переходить в стан a_6 . На основній дорозі спалахує червоний. На пішохідному переході протягом t_d горить червоний сигнал, після чого загоряється зелений. Затримка, що відповідає стану a_6 , дорівнює t_{06} .

Таким чином, для системи управління дорожнім світлофором може бути побудований темпоральний граф переходів часового автомата Мура. Даний граф представлений на рисунку 9.2.

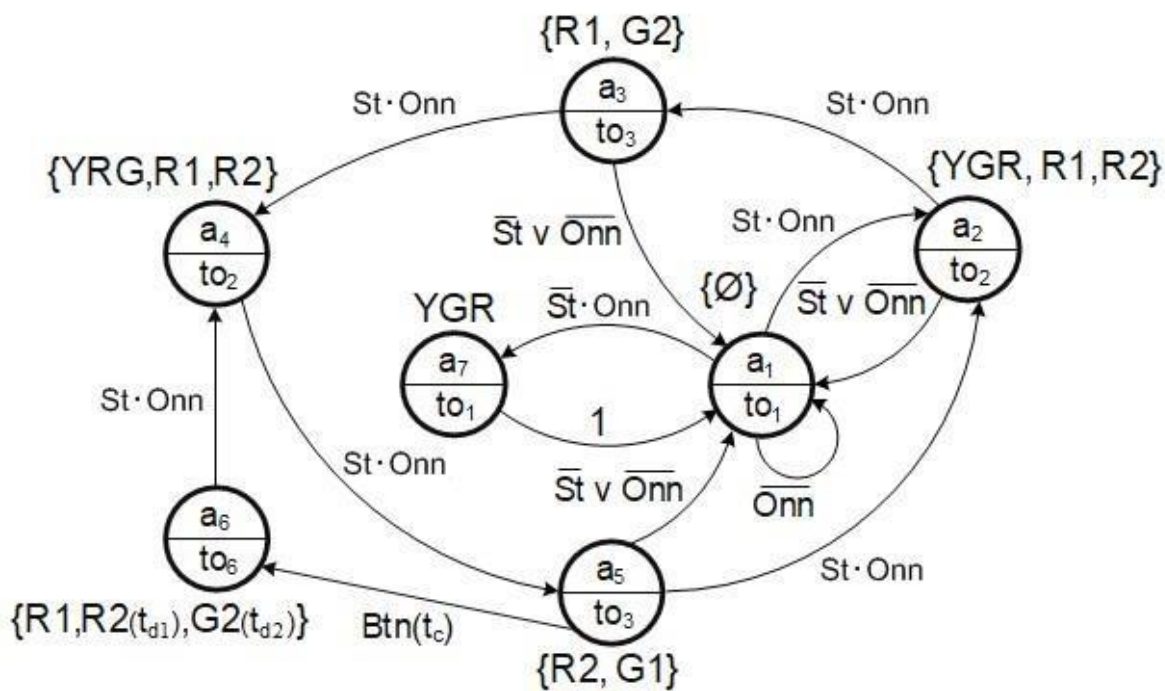


Рисунок 9.2 - Темпоральний граф часового автомата Мура для системи управління дорожнім світлофором

У побудованому темпоральному графі стани a_1 і a_7 відповідають нічному режиму функціонування світлофора, а стани a_2 , a_3 , a_4 , a_5 - денному режиму.

На рисунку 9.3 представлена часова діаграма функціонування часового автомата Мура для системи управління дорожнім світлофором.

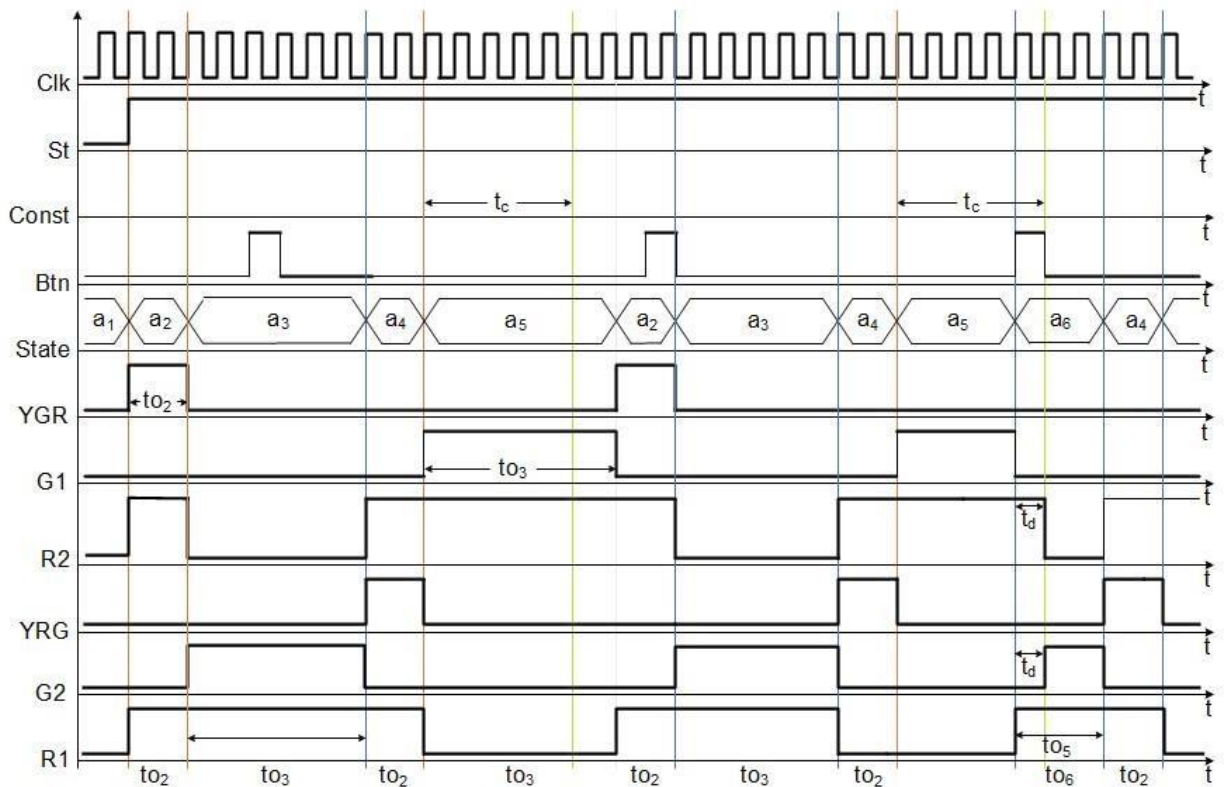


Рисунок 9.3 - Часова діаграма автомата Мура для системи управління дорожнім світлофором

На рисунку 9.3 розглянуто денний цикл роботи світлофора, а також натискання кнопки Btn для включення зеленого світла на пішохідному переході.

10 VHDL-МОДЕЛІ ЧАСОВОГО АВТОМАТА МУРА

Запропоновано три варіанти VHDL-опису часового автомата Мура для системи управління дорожнім світлофором:

- однопроцесний,
- двопроцесний,
- трьохпроцесний.

Були визначені конкретні значення часових характеристик автомата. Дані характеристики задані в автоматних тактах. Значення вхідних таймаутів:

- $t_{o1} = 1$ - інтервал миготіння жовтого світла в нічному режимі,
- $t_{o2} = 5$ - проміжок часу, протягом якого горять жовтий і червоний на основній дорозі в денному режимі,
- $t_{o3} = 45$ - проміжок часу, протягом якого горить зелений на основній дорозі або на пішохідному переході, - $t_{o6} = 20$ - таймаут стану a_6 .

Часове обмеження для події B_{tn} $t_c = [10, 40]$. Значення вихідних затримок в стані a_6 :

- для G2 $t_{d0} = 2$, $t_{\Delta} = \infty$; - для R2 $t_{d0} = 0$, $t_{\Delta} = 2$.

Усі варіанти VHDL-опису використовують той самий набір констант, що описують часові параметри автомата.

Декларація констант виконується в пакеті `t fsm_pack`. Таким чином, пакет містить декларацію наступних констант:

- `count_length` - розмір таймера,
- `T1`, `T2`, `T3`, `T6` - значення вхідних таймаутів,
- `output_delay_G2` - значення вихідної затримки,
- `constraint_a5_L`, `constraint_a5_H` - нижня і верхня межі

часового обмеження для стану a_5 .

На рисунку 10.1 представлений фрагмент зазначеного пакета.

```
package t fsm_pack is
  constant count_length : integer := 6;
  constant T1 : std_logic_vector ( count_length - 1 downto 0 ) := "000001";
  ...
  constant output_delay_G2 : std_logic_vector ( count_length - 1 downto 0 ) := "000010";           -- 2
  constant constraint_a5_L : std_logic_vector ( count_length - 1 downto 0 ) := "001010";           -- [10, 40]
  constant constraint_a5_H : std_logic_vector ( count_length - 1 downto 0 ) := "101000";
end t fsm_pack;
```

Рисунок 10.1 - Фрагмент пакета tfsm pack

10.1 Трьохпроцесна модель

На рисунку 10.2 представлено опис заголовка об'єкта проектування. Для всіх портів використовується тип даних `std_logic`.

```
entity tfsm_three_proc is
  port (
    clk, reset : in std_logic;
    Onn, St : in std_logic;
    Btn : in std_logic;
    G1, YGR, YRG, R1 : out std_logic;
    G2, R2 : out std_logic
  );
end;
```

Рисунок 10.2 - Опис заголовка об'єкта проектування

Трьохпроцесна модель використовує такі процеси:

- процес для формування послідовної частини, що відповідає стану автомата,
- процес для реалізації таймера,
- процес для реалізації функції переходів.

На рисунку 10.3 представлений фрагмент VHDL опису автомата з декларацією сигналів `state` і `count`.

```
type state_type is ( a1, a2, a3, a4, a5, a6, a7 );
signal state, next_state : state_type;
signal count : std_logic_vector ( count_length - 1 downto 0 );
```

Рисунок 10.3 - Декларація сигналів `state` і `count`

Процес для реалізації зворотного зв'язку з затримкою на один такт і синхронізацією в трьохпроцесній моделі для часового автомата аналогічний

процесу в шаблоні для класичного автомата Мура. Даний процес представлений на рисунку 10.4.

```
-- state synchronization
process (Clk, Reset)
begin
    if Reset = '1' then
        state <= a1;
    elsif rising_edge(Clk) then
        state <= next_state;
    end if;
end process;
```

Рисунок 10.4 - Процес для реалізації зворотного зв'язку з затримкою на один такт і синхронізацією для трьохпроцесної моделі

Процес для реалізації таймера представлений на рисунку 10.5. Таймер повинен скидатися в 0, коли активний сигнал Reset. Крім того, скидання відбувається при виконанні переходу часового автомата. Перехід виконується, якщо значення сигналу state не дорівнює значенню сигналу next_state. Крім того, може бути виконаний перехід-петля зі стану a₁.

```
-- timer synchronization
process (Clk, Reset)
begin
    if Reset = '1' then
        count <= (others => '0');
    elsif rising_edge(Clk) then
        if State /= next_state then
            count <= (others => '0');
        elsif state = a1 and count >= T1 - 1 and Onn = '0' then
            count <= (others => '0');
        else
            count <= count + 1;
        end if;
    end if;
end process;
```

Рисунок 10.5 - Процес для реалізації таймера для трьохпроцесної моделі

На рисунку 10.6 представлений фрагмент опису вихідних сигналів, для яких не визначена вихідна затримка. Можна помітити, що даний опис є аналогічним шаблону для класичного автомата Мура.

```
YRG <= '1' when
    ( state = a4 )
    else '0';

R1 <= '1' when
    ( ( state = a2 ) or ( state = a3 ) or ( state = a4 ) or ( state = a6 ) )
    else '0';
```

Рисунок 10.6 - Фрагмент опису вихідних сигналів, для яких не визначена вихідна затримка

На рисунку 10.7 представлений фрагмент опису вихідних сигналів, для яких визначена вихідна затримка: G2 і R2. Для стану a6 використовуються додаткові перевірки значення лічильника.

```
G2 <= '1' when
    ( state = a3 )
    or ( state = a6 and count >= output_delay_G2 )
    else '0';

R2 <= '1' when
    ( ( state = a2 ) or ( state = a4 ) or ( state = a5 ) )
    or ( state = a6 and count < output_delay_G2 )
    else '0';
```

Рисунок 10.7 - Опис вихідних сигналів, для яких визначена вихідна затримка

На рисунку 10.8 представлений фрагмент процесу для реалізації функції переходів. Основні відмінності від шаблону для класичного автомата Мура полягають у наступному:

- в список чутливості процесу додається сигнал count,
- виконується додаткове порівняння значення таймера з таймаутом поточного стану,

- виконуються додаткові перевірки значення сигналу Btn і перевірка, чи задовольняє значення таймера часовому обмеженню для поточного стану.

```

-- transition function
process (state, Onn, St, Btn, count )
begin
  case state is
    ...
    when a5 =>
      if      Btn = '1'
        and   count >= constraint_a5_L - 1
        and   count < constraint_a5_H then
        next_state <= a6;
      elsif count < T3 - 1 then
        next_state <= state;
      elsif Onn = '0' then
        next_state <= a1;
      elsif St = '0' then
        next_state <= a1;
      else
        next_state <= a2;
      end if;
    ...
  end case;
end process;

```

Рисунок 10.8 - Фрагмент процесу для реалізації функції переходів для трьохпроцесної моделі

10.2 Однопроцесна модель

На рисунку 10.9 представлені фрагменти процесу, що використовується в однопроцесній моделі. Єдиний процес в даному шаблоні використовується для наступних цілей:

- для реалізації зворотного зв'язку з затримкою на один такт і синхронізацією,
- для реалізації таймера,
- для реалізації функції переходів.

Опис вихідних сигналів виконується аналогічно трьохпроцесній моделі.

```

process (Clk, Reset)
begin
    if Reset = '1' then
        state <= a1;
        count <= (others => '0');
    elsif rising_edge(Clk) then
        count <= ( others => '0' );

        case state is
            ...
            when a5 =>
                if
                    Btn = '1'
                    and count >= constraint_a5_L - 1
                    and count < constraint_a5_H then
                        state <= a6;
                    elsif count < T3 - 1 then
                        state <= state;
                        count <= count + 1;
                    elsif Onn = '0' then
                        state <= a1;
                    elsif St = '0' then
                        state <= a1;
                    else
                        state <= a2;
                    end if;
                ...
            end case;
        end if;
    end process;

```

Рисунок 10.9 - Фрагмент процесу для однопроцесної моделі

10.3 Двопроцесна модель

На рисунку 10.10 представлений фрагмент VHDL-опису автомата з декларацією сигналів state і count для двопроцесної моделі. Відмінність від трьохпроцесної моделі полягає в додатковій декларації сигналу next_count, який використовується аналогічно сигналу next_state.

```

type state_type is ( a1, a2, a3, a4, a5, a6, a7 );
signal state, next_state : state_type;
signal count, next_count : std_logic_vector ( count_length - 1 downto 0 );

```

Рисунок 10.10 - Декларація сигналів state і count для двопроцесної моделі

На рисунку 10.11 представлений процес для реалізації послідовної частини часового автомата. По суті, цей процес об'єднує такі процеси трьохпроцесної моделі:

- для реалізації зворотного зв'язку з затримкою на один такт і синхронізацією,
- для реалізації таймера.

Використання сигналу next_count дозволяє значно спростити даний фрагмент VHDL-опису.

```
-- state synchronization
process (Clk, Reset)
begin
    if Reset = '1' then
        state <= a1;
        count <= ( others => '0' );
    elsif rising_edge(Clk) then
        state <= next_state;
        count <= next_count;
    end if;
end process;
```

Рисунок 10.11 - Процес для реалізації зворотного зв'язку з затримкою на один такт і синхронізацією для двопроцесної моделі

На рисунку 10.12 представлений фрагмент процесу для реалізації функції переходів. Основна відмінність від трьохпроцесного шаблону часового автомата полягає в додаванні операторів призначення сигналу next_count.

```

process (state, Onn, St, Btn, count )
begin
    next_count <= (others => '0');
    case state is
        ...
        when a5 =>
            if
                Btn = '1'
                and count >= constraint_a5_L - 1
                and count < constraint_a5_H then
                    next_state <= a6;
            elsif count < T3 - 1 then
                next_state <= state;
                next_count <= count + 1;
            elsif Onn = '0' then
                next_state <= a1;
            elsif St = '0' then
                next_state <= a1;
            else
                next_state <= a2;
            end if;
        ...
    end case;
end process;

```

Рисунок 10.12 - Фрагмент процесу для реалізації функції переходів для двопроесної моделі

11 РЕЗУЛЬТАТИ СИНТЕЗУ

Було виконано синтез пристрою управління дорожнім світлофором в системі Xilinx ISE для мікросхеми FPGA XC3S500E сімейства Spartan 3E в корпусі FG 320. На рисунку 11.1 представлені коди станів, отримані в результаті синтезу. При використанні трьохпроєсної, однопроєсної й двопроесної моделей кодування станів однакове.

State	Encoding
a_1	000
a_2	001
a_3	011
a_4	100
a_5	101
a_6	110
a_7	101

Рисунок 11.1 - Кодування станів

На рисунку 11.2 представлені результати синтезу пристрою на мікросхемі XC3S500E.

За кількістю послідовних елементів результати використання всіх 3 моделей ідентичні. Було синтезовано 9 D-тригерів з асинхронним скиданням. Для зберігання поточного стану використовується 3 D-тригери, оскільки число станів дорівнює 7. Для зберігання поточного значення лічильника використовується 6 D-тригерів, оскільки максимальне константне значення, з яким відбувається порівняння лічильника, дорівнює 45.

	Однопроцесна модель	Двопроцесна модель	Трьохпроцесна модель
BELS	50	50	57
INV	1	1	0
LUT2	4	4	7
LUT2_L	1	1	0
LUT3	12	12	15
LUT3_D	0	0	1
LUT3_L	2	2	4
LUT4	23	23	18
LUT4_D	3	3	5
LUT4_L	1	1	6
MUXF5	3	3	1
FlipFlops/Latches	9	9	9
FDC	9	9	9

Рисунок 11.2 - Результати синтезу

Для реалізації функцій переходів і виходів були синтезовані комбінаційні схеми. За кількістю і типом синтезованих комбінаційних елементів результати використання одного і двох процесів ідентичні. При використанні трьох процесів було синтезовано дещо більшу кількість елементів.

Отримані результати синтезу дозволяють стверджувати наступне.

1. Всі 3 запропоновані VHDL-моделі розроблені коректно.
2. З точки зору використовуваних апаратних ресурсів усі 3

VHDL-моделі дають приблизно однакові результати.

12 РЕЗУЛЬТАТИ ВЕРИФІКАЦІЇ МОДЕЛЕЙ

Для виконання верифікації використовується TestBench. Результати моделювання представляються на часових діаграмах. Моделювання було виконано за допомогою системи ISim, яка інтегрована в середу Xilinx ISE.

При верифікації перевіряються наступні сценарії:

- світлофор вимкнений,
- нічний цикл роботи світлофора,
- стандартний цикл роботи світлофора,
- вимикання світлофора,
- перехід в нічний цикл роботи світлофора,
- перехід в стандартний цикл роботи світлофора,
- надходження події Vtn поза допустимого інтервалу прийому подій,
- обробка події Vtn в стані a_6 .

12.1 Моделювання до імплементації

Розглянемо результати моделювання до імплементації для трьохпроцесної моделі. Період синхросигналу $\tau = 100$ нс.

На рисунку 12.1 наведено фрагмент часової діаграми, який відповідає виключеному світлофору, тобто $Onn = "0"$. У 50 і 150 нс виконуються переходипетлі зі стану a_1 .

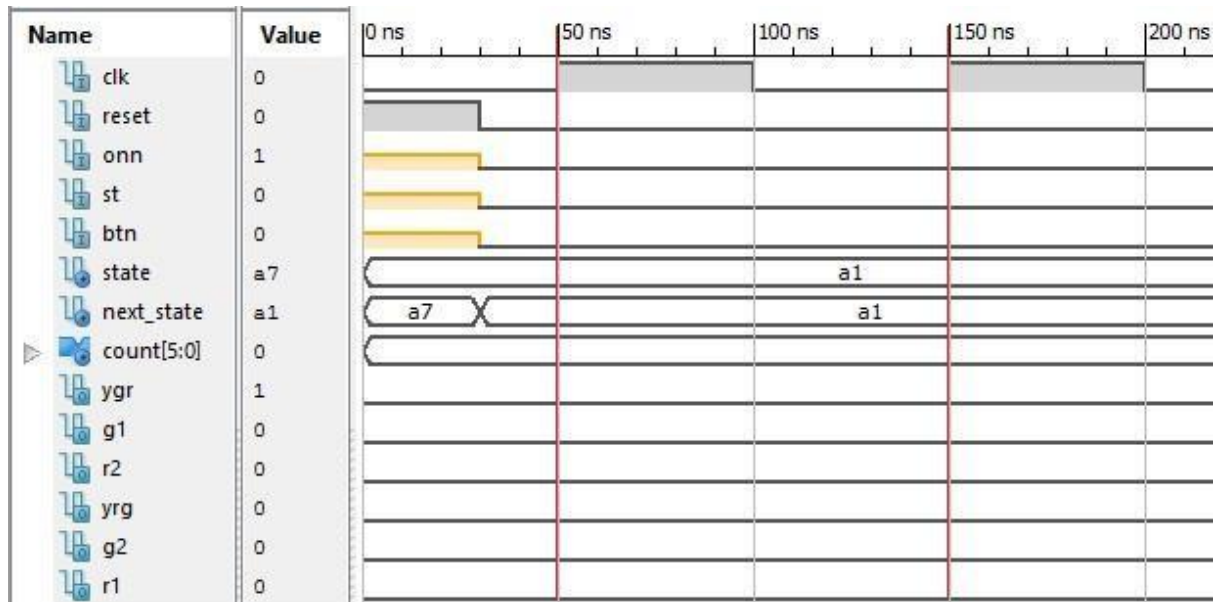


Рисунок 12.1 - Фрагмент часової діаграми, що відповідає вимкненому світлофору

На рисунку 12.2 наведено фрагмент часової діаграми, що відповідає нічному режиму роботи світлофора, тобто $Onn = "1"$, $St = "0"$.

У 250 нс виконується перехід з a_1 в a_7 . З 250 нс до 350 нс (1 такт) вихідний сигнал YGR активний. У 350 нс виконується перехід з a_7 в a_1 . З 350 нс до 450 нс (1 такт) усі вихідні сигнали обнулені. Таким чином, виконується миготіння жовтого.

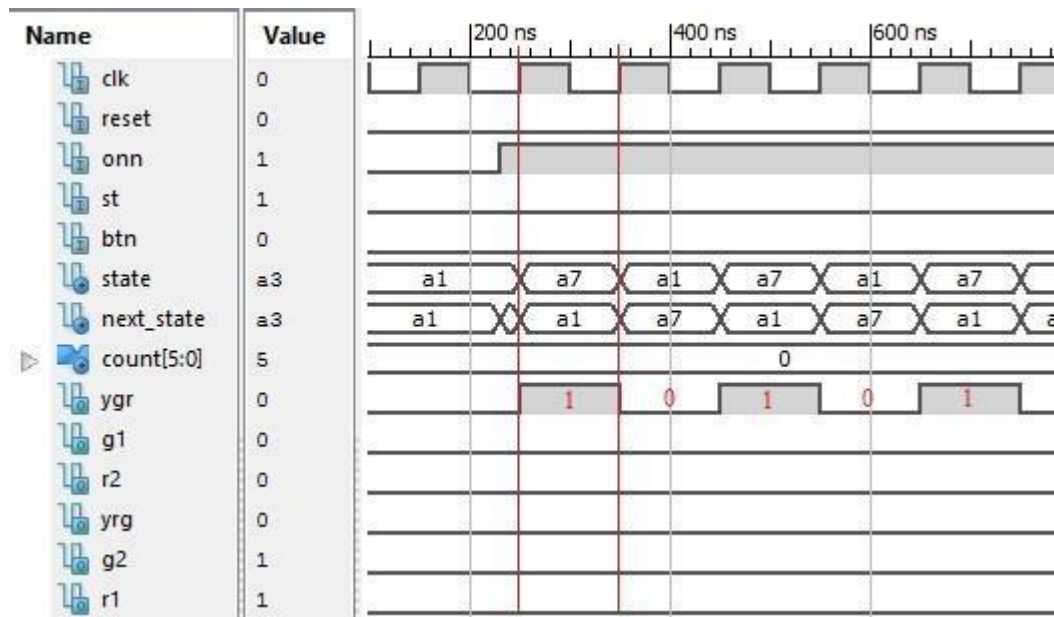


Рисунок 12.2 - Фрагмент часової діаграми, що відповідає нічному режиму роботи світлофора

На рисунку 12.3 наведено фрагмент часової діаграми, відповідний переходу в денний цикл світлофора, тобто $Onn = "1"$, $St = "1"$.

До 850 нс автомат знаходиться в стані a1, в даному стані всі вихідні сигнали обнулені. По закінченню часу очікування для стану a1 (1 такт) в 850 нс автомат переходить в стан a2. На пішохідному переході загоряється червоний, на основній дорозі горить червоний і жовтий.

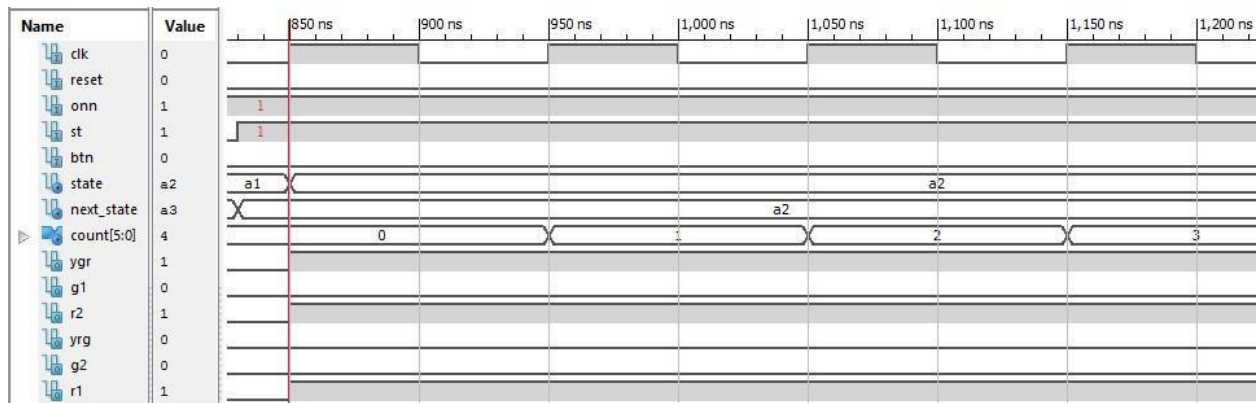


Рисунок 12.3 - Фрагмент часової діаграми, що відповідає переходу в денний ЦИКЛ

На рисунку 12.4 наведено фрагмент часової діаграми, що відповідає денному режиму роботи світлофора, тобто $Onn = "1"$, $St = "1"$.

До 5850 нс автомат знаходиться в стані аз, даний стан відповідає ситуації, коли на основній дорозі горить червоний, на пішохідному переході - зелений. По закінченню часу очікування для стану аз (45 тактів) в 5850 нс автомат переходить в стан а4 і знаходиться в ньому з 5850 нс до 6350 нс (5 тактів). На пішохідному переході загоряється червоний, на основній дорозі горить червоний і жовтий.

По закінченню часу очікування для стану а4 (5 тактів) в 6350 нс автомат переходить в стан а5. На основній дорозі загоряється зелений.

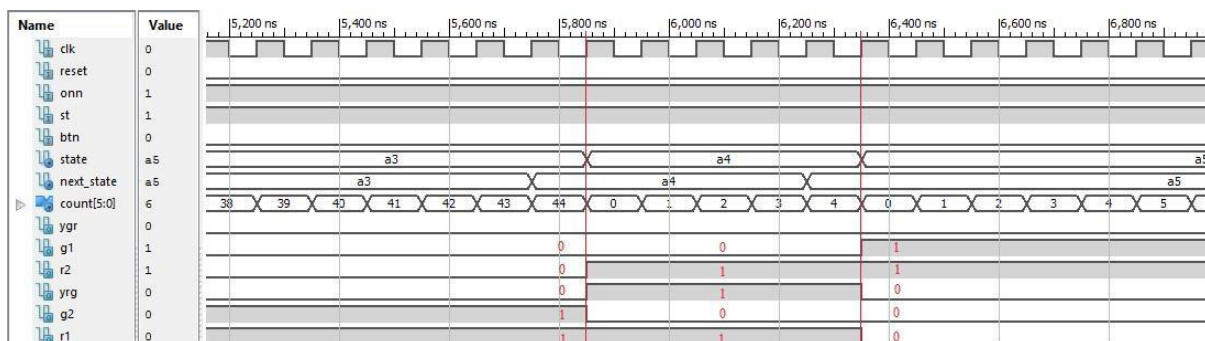


Рисунок 12.4 - Фрагмент часової діаграми, що відповідає денному циклу світлофора

На рисунку 12.5 наведено фрагмент часової діаграми, відповідний виключенню світлофора.

З 10850 нс до 11350 нс (5 тактів) автомат знаходиться в стані a2, на пішохідному переході горить червоний, на основній дорозі - червоний і жовтий.

В 11350 нс, оскільки $Onn = "0"$, автомат переходить в стан a1, в якому всі вихідні сигнали обнулені.

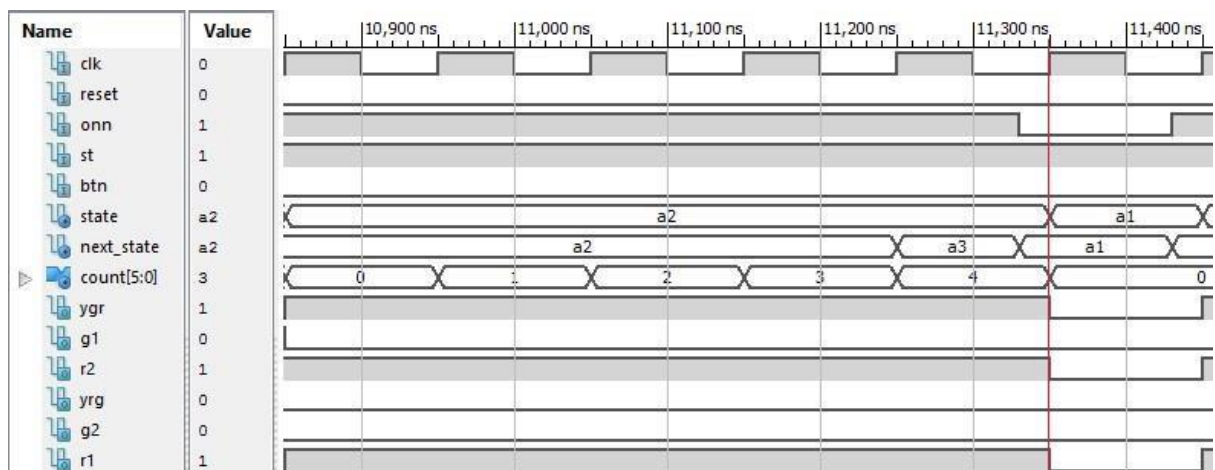


Рисунок 12.5 - Фрагмент часової діаграми, що відповідає виключенню світлофора

На рисунку 12.6 наведено фрагмент часової діаграми, що відповідає переходу в нічний цикл світлофора зі стану a2.

До 34050 нс автомат знаходиться в стані a2, на пішохідному переході горить червоний, на основній дорозі - червоний і жовтий.

В 34050 нс, оскільки $St = "0"$, автомат переходить в стан a1, в якому всі вихідні сигнали обнулені.

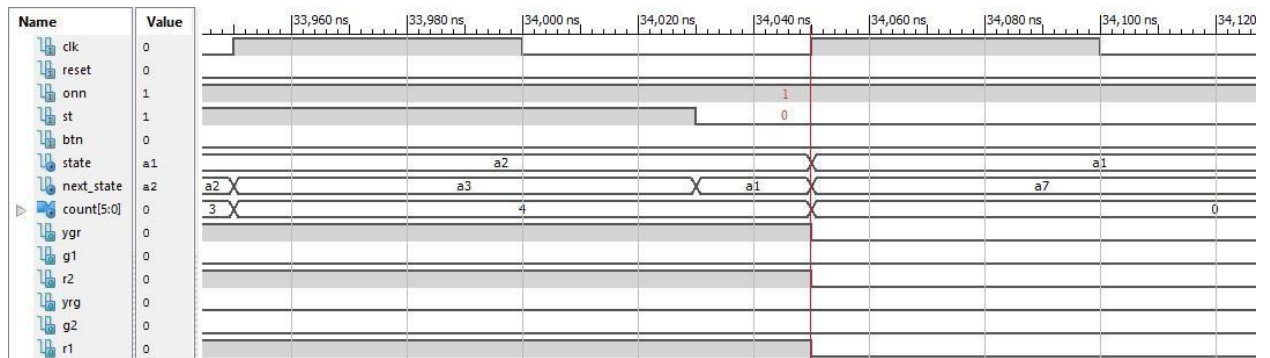


Рисунок 12.6 - Фрагмент часової діаграми, що відповідає переходу в нічний цикл зі стану a2

На рисунку 12.7 наведено фрагмент часової діаграми, що відповідає переходу в нічний цикл світлофора зі стану a3.

До 39150 нс автомат знаходиться в стані a3, на пішохідному переході горить зелений, на основній дорозі - червоний. В 39150 нс, оскільки $St = "0"$, автомат переходить в стан a1, в якому всі вихідні сигнали обнулені.

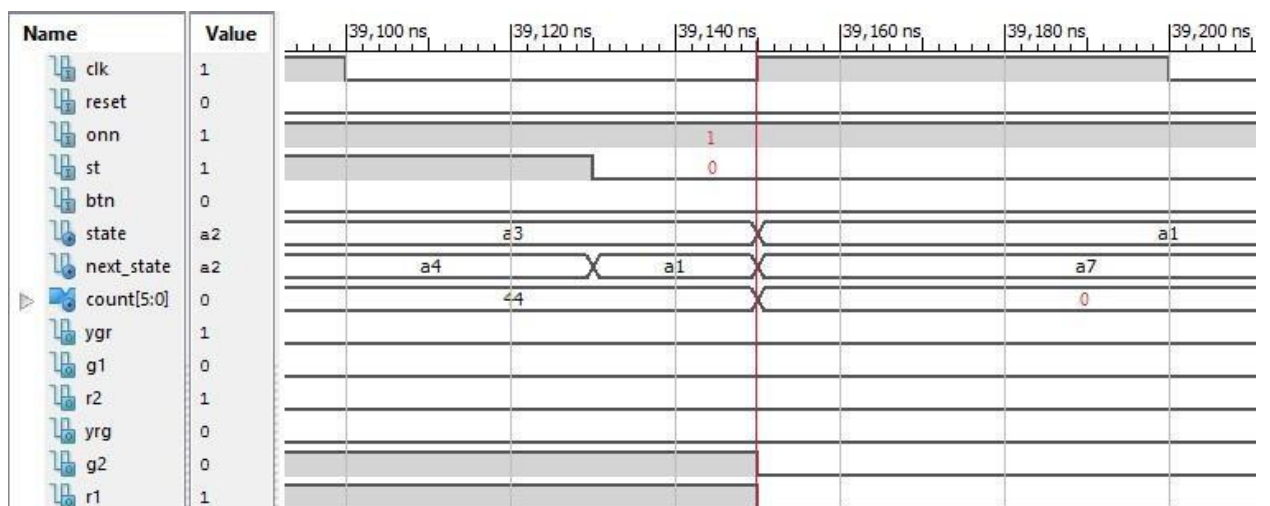


Рисунок 12.7 - Фрагмент часової діаграми, що відповідає переходу в нічний цикл зі стану a3

На рисунку 12.9 наведено фрагмент часової діаграми, що відповідає натисканню кнопки Btn в стані a_5 в момент, що задовольняє часовому обмеженню.

До 31450 нс автомат знаходиться в стані a_5 , на основній дорозі горить зелений, на пішохідному переході - червоний. В 31450 нс виконується перехід до a_6 у відповідь на натискання кнопки Btn , оскільки Btn натиснута в момент, що задовольняє часовому обмеженню. На основній дорозі загоряється червоний. Оскільки вихідна затримка $R2$ $t_{d0} = 0$, $t_{\Delta} = 2$, з 31450 нс до 31650 нс (2 такти) на пішохідному переході горить червоний, а вихідна затримка $G2$ $t_{d0} = 2$, $t_{\Delta} = \infty$, в 31650 нс загоряється зелений.

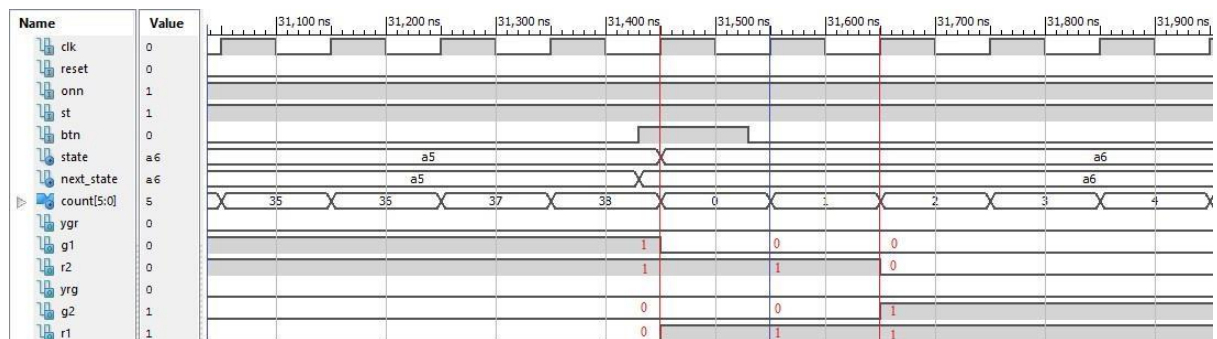


Рисунок 12.9 - Фрагмент часової діаграми, що відповідає натисканню кнопки Btn в момент всередині допустимого інтервалу

При моделюванні до імплементації відмінностей у функціонуванні однопроцесної, двопроцесної й трьохпроцесної моделей не виявлено.

12.2 Моделювання після імплементації на FPGA

12.2.1 Трьохпроцесна модель

На рисунку 12.10 наведено фрагмент часової діаграми для трьохпроцесної моделі, що відповідає нічному режиму роботи світлофора, тобто $Onn = "1"$, $St =$

"0".

До 250 нс автомат знаходиться в стані a_1 . У 250 нс автомат переходить в стан a_7 . З 258,5 нс до 358.5 нс (1 такт) вихідний сигнал YRG, відповідний стану a_7 , активний. У 350 нс автомат переходить в стан a_1 . З 358.5 нс до 458.5 нс (1 такт) усі вихідні сигнали обнулені, що відповідає стану a_1 .

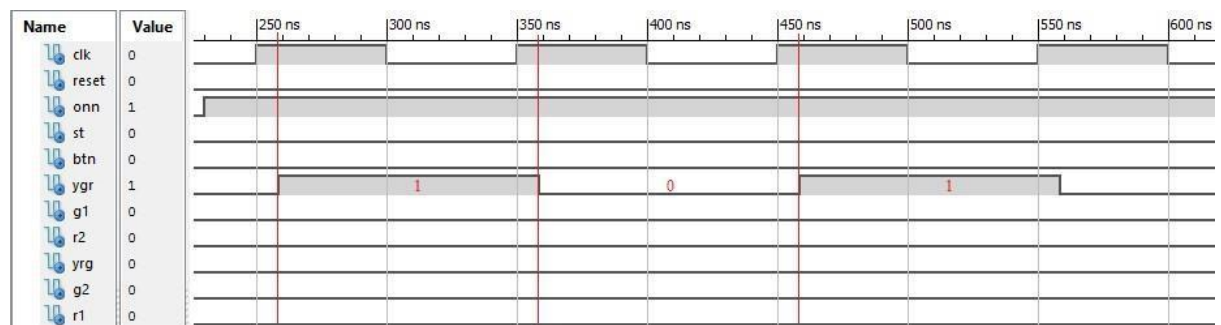


Рисунок 12.10 - Фрагмент часової діаграми для трьохпроцесної моделі, що відповідає нічному режиму роботи світлофора

На рисунку 12.11 наведено фрагмент часової діаграми для трьохпроцесної моделі, відповідний переходу в денний режим роботи світлофора.

До 850 нс автомат знаходиться в стані a_1 . В даному стані всі вихідні сигнали обнулені. У 850 нс автомат переходить в стан a_2 . С 858.5 нс автомат видає YRG, R2, R1, що відповідає стану a_2 .

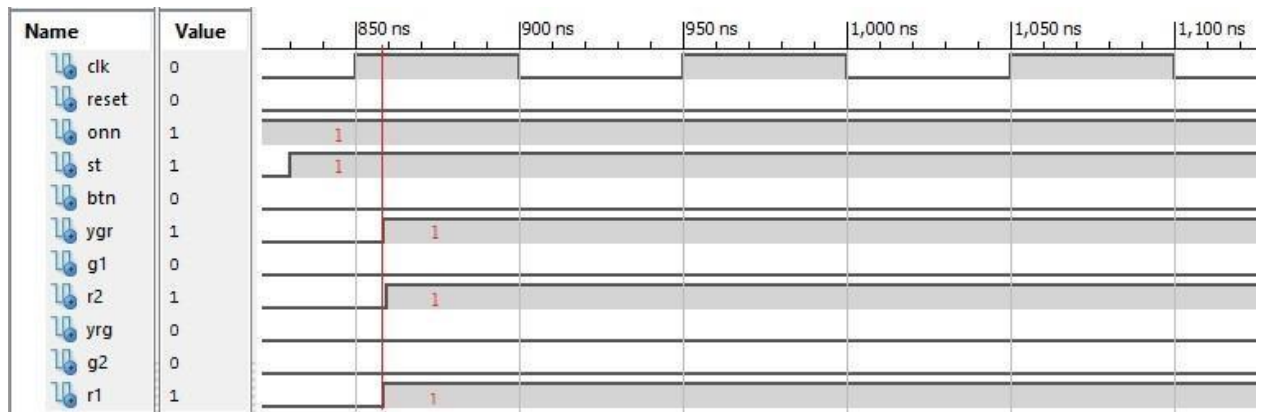


Рисунок 12.11 - Фрагмент часової діаграми для трьохпроцесної моделі, що відповідає переходу в денний цикл

На рисунку 12.12 наведено фрагмент часової діаграми для трьохпроцесної моделі, що відповідає денному режиму роботи світлофора, тобто $Onn = "1"$, $St = "1"$.

До 5850 нс автомат знаходиться в стані а3. Активними вихідними сигналами є G2 і R1. У 5850 автомат переходить в стан а4. С 5858.9 нс до 6358 нс (приблизно 5 тактів) автомат видає R1, R2 і YRG, що відповідає стану а4. У 6350 нс за переднім фронтом елементи пам'яті автомата змінюють свій стан. З 6358 нс автомат видає вихідні сигнали: G1, R2, що відповідають стану а5.

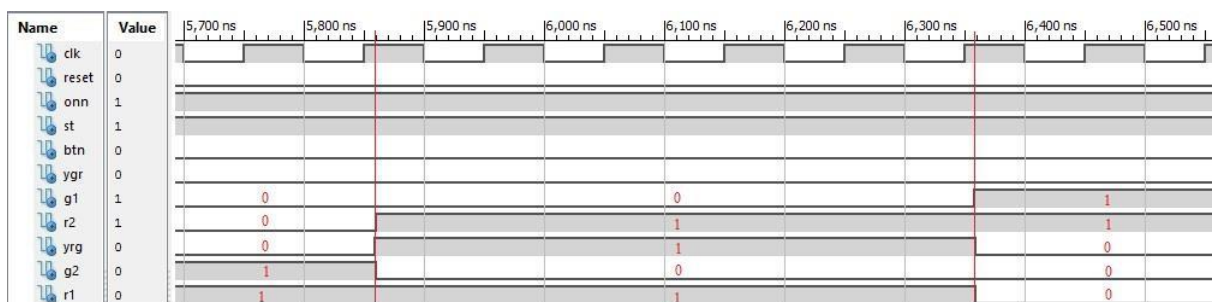


Рисунок 12.12 - Фрагмент часової діаграми для трьохпроцесної моделі, що відповідає денному циклу світлофора

На рисунку 12.13 наведено фрагмент часової діаграми для трьохпроцесної моделі, що відповідає натисканню кнопки V_{tn} в стані a_5 в момент, що задовольняє часовому обмеженню.

В 31430 нс відбувається натискання кнопки Btn. В 31450 нс автомат переходить в стан a_6 . До 31458 нс автомат видає вихідні сигнали: G1, R2, що відповідає стану a_5 . З 31458 нс активний сигнал R1, що відповідає стану a_6 . З 31461 нс до 31659.8 нс (приблизно 2 такти) автомат видає сигнал R2, оскільки його вихідна затримка $t_{d0} = 0$, $t_{\Delta} = 2$. З 31659.8 нс автомат видає сигнал G2, оскільки вихідна затримка сигналу $t_{d0} = 2$, $t_{\Delta} = \infty$. З 31459.6 нс до 31461 нс $G1 = "0"$, $R2 = "1"$.

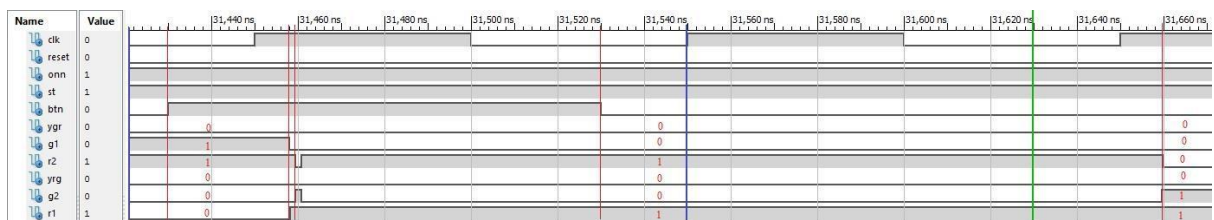


Рисунок 12.13 - Фрагмент часової діаграми для трьохпроцесної моделі, що відповідає натисканню Btn в момент всередині допустимого інтервалу

12.2.2 Двопроцесна модель

На рисунку 12.14 наведено фрагмент часової діаграми для двопроцесної моделі, що відповідає переходу в денний режим роботи світлофора.

До 850 нс автомат знаходиться в стані a_1 . В даному стані всі вихідні сигнали обнулені. У 859 нс автомат переходить в стан a_2 . С 859 нс автомат видає YRG, R2, R1, що відповідає стану a_2 .

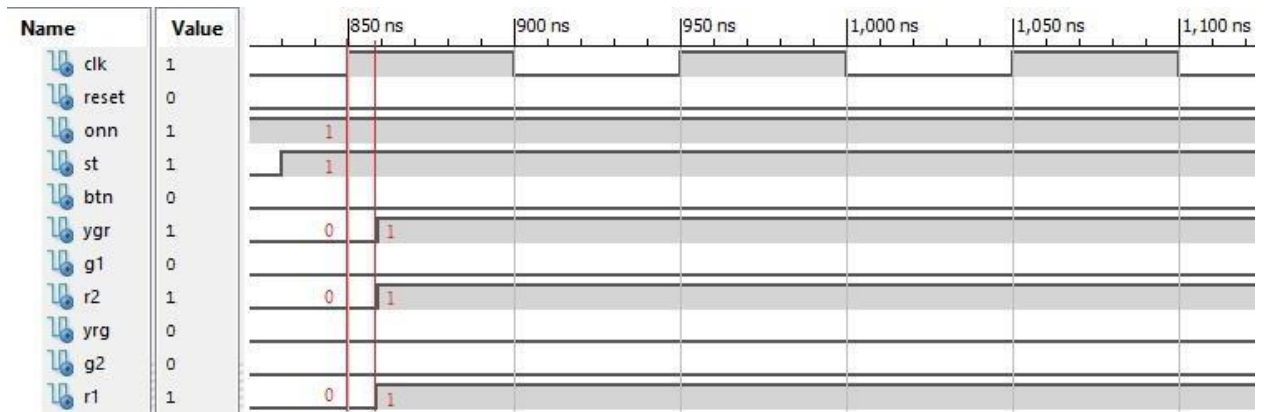


Рисунок 12.14 - Фрагмент часової діаграми для двопроцесної моделі, що відповідає переходу в денний режим

На рисунку 12.15 наведено фрагмент часової діаграми для двопроцесної моделі, що відповідає денному режиму роботи світлофора, тобто при $Onn = "1"$, $St = "1"$.

До 5850 нс автомат знаходиться в стані аз. Активними вихідними сигналами є G2 і R1. У 5850 автомат переходить в стан а4. С 5857.39 нс до 6358.26 нс (приблизно 5 тактів) автомат видає R1, R2 і YRG, що відповідає стану а4. У 6350 нс по передньому фронту елементи пам'яті автомата змінюють свій стан. З 6358.26 нс автомат видає вихідні сигнали: G1, R2, що відповідають стану а5.

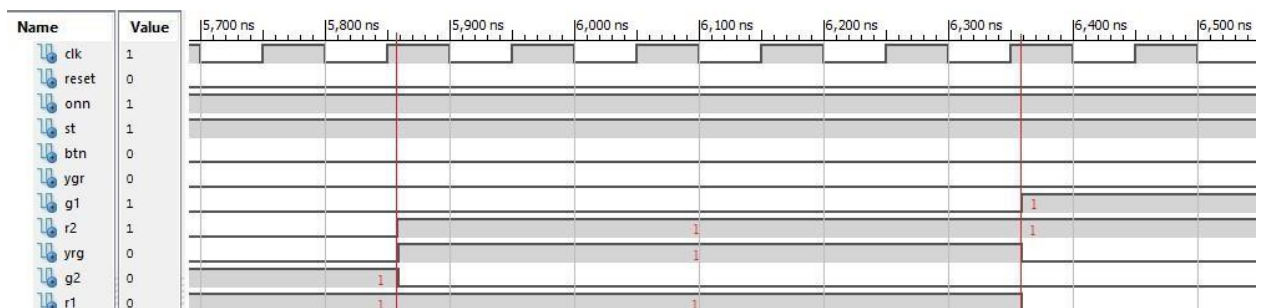


Рисунок 12.15 - Фрагмент часової діаграми для двопроцесної моделі, що відповідає денному циклу

В 31430 нс відбувається натискання кнопки Btn. В 31450 нс автомат

6

На рисунку 12.16 наведено фрагмент часової діаграми для двопроесної моделі, що відповідає натисканню кнопки Btn в стані a5 в момент, що задовольняє часовому обмеженню. переходить в стан a. До 31458.3 нс автомат видає вихідні сигнали: G1, R2, що відповідає стану a5. З 31461.7 нс активний сигнал R1, що відповідає стану a6. З 31461.7 нс до 31660 нс (приблизно 2 такти) автомат видає сигнал R2, оскільки його вихідна затримка $t_{d0} = 0$, $t_{\Delta} = 2$. З 31660 нс автомат видає сигнал G2, оскільки вихідна затримка сигналу $t_{d0} = 2$, $t_{\Delta} = \infty$. З 31458.3 нс до 31461.7 нс $G1 = "0"$, $R2 = "1"$.

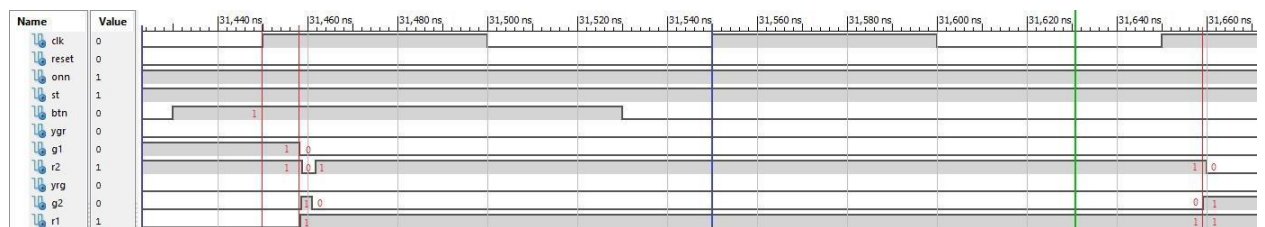


Рисунок 12.16 - Фрагмент часової діаграми для двопроесної моделі, що відповідає натисканню Btn в момент всередині допустимого інтервалу

12.2.3 Однопроесна модель

На рисунку 12.17 наведено фрагмент часової діаграми для однопроесної моделі, що відповідає переходу в денний цикл світлофора, тобто при $Onn = "1"$, $St = "1"$.

До 850 нс автомат знаходиться в стані a1. У 850 нс автомат переходить в стан a2. До 858.7 нс усі вихідні сигнали обнулені. З 858.7 нс на пішохідному переході загоряється червоний, на основній дорозі горить червоний і жовтий.

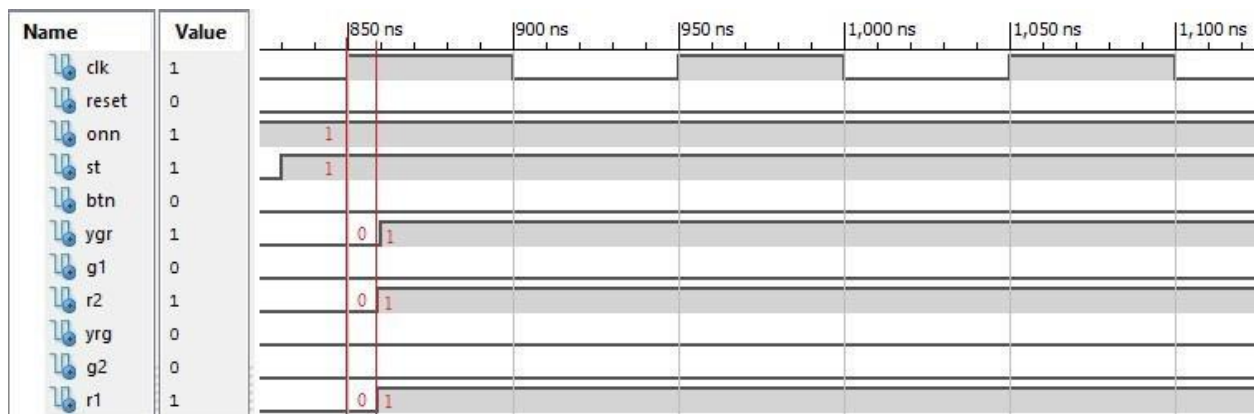


Рисунок 12.17 - Фрагмент часової діаграми для однопроцесної моделі, що відповідає переходу в денний цикл

На рисунку 12.18 наведено фрагмент часової діаграми для однопроцесної моделі, що відповідає денному режиму роботи світлофора, тобто при $Onn = "1"$, $St = "1"$.

До 5850 нс автомат знаходиться в стані аз. Активними вихідними сигналами є G2 і R1. У 5850 автомат переходить в стан а4. С 5858.2 нс до 6359.05 нс (приблизно 5 тактів) автомат видає R1, R2 і YRG, що відповідає стану а4. У 6350 нс в момент переднього фронту елементи пам'яті автомата змінюють свій стан. З 6359.05 нс автомат видає вихідні сигнали: G1, R2, що відповідають стану а5.

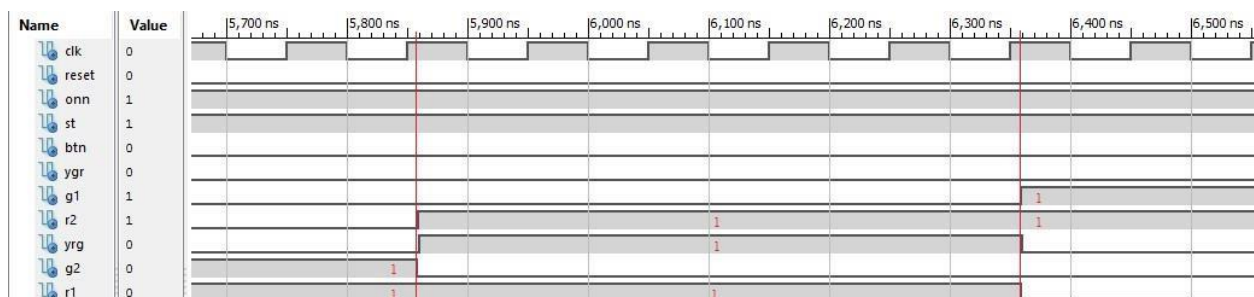


Рисунок 12.18 - Фрагмент часової діаграми для однопроцесної моделі, що відповідає денному циклу

В 31430 нс відбувається натискання кнопки Btn. В 31450 нс автомат

6

На рисунку 12.19 наведено фрагмент часової діаграми для однопроцесної моделі, що відповідає натисканню кнопки Btn в стані a5 в момент, що задовольняє часовому обмеженню. автомат переходить в стан a. До 31458.4 нс автомат видає вихідні сигнали: G1, R2, що відповідають стану a5. З 31459.05 нс активний сигнал R1, що відповідає стану a6. З 31460.35 нс до 31659.63 нс (приблизно 2 такти) автомат видає сигнал R2, оскільки його вихідна затримка $t_{d0} = 0$, $t_{\Delta} = 2$. З 31659.63 нс автомат видає сигнал

G2, оскільки вихідна затримка $t_{d0} = 2$, $t_{\Delta} = \infty$. З 31458.4 нс до 31460.35 нс G2 = "1", R2 = "0".

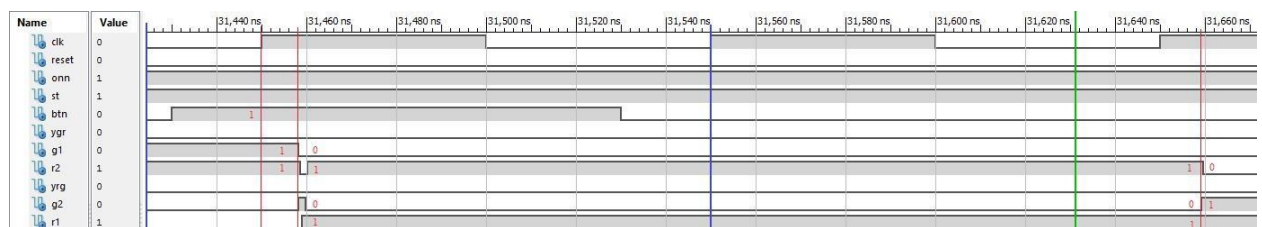


Рисунок 12.19 - Фрагмент часової діаграми для однопроцесної моделі, що відповідає натисканню Btn в момент всередині допустимого інтервалу

Таким чином, результати верифікації підтверджують коректність всіх трьох розроблених VHDL-моделей.

ВИСНОВКИ

В результаті атестаційної роботи розроблений метод побудови опису на HDL для систем реального часу, що може бути синтезовано в програмовану логічну інтегральну схему.

У даній роботі були розглянуті моделі систем реального часу, такі як діаграма станів, часовий автомат з множиною таймерів, часовий автомат з єдиним таймером, часовий розширений автомат, часовий автомат з таймаутами та часовими обмеженнями. На основі даних моделей була запропонована повна структурна модель часового автомата Мура.

Були проаналізовані особливості апаратного та програмно-апаратного підходів до реалізації систем реального часу. Були запропоновані способи використання додаткового лічильника для зберігання значення часової змінної та реалізації часових параметрів при використанні апаратного підходу на основі структурної моделі часового автомата Мура.

Був проаналізований алгоритм функціонування дорожнього світлофора. На основі даного алгоритму був розроблений темпоральний граф Мура для системи керування дорожнім світлофором.

Для отриманого темпорального графа були розроблені VHDL-моделі часового автомата Мура: трьохпроцесна, двопроцесна, однопроцесна. Були виділені відмінності даних моделей від VHDL-шаблонів класичного автомата Мура.

За допомогою середовища Xilinx ISE були виконані синтез і верифікація розроблених моделей VHDL-моделей. Синтез і моделювання до та після імплементації виконувалось для мікросхеми XC3S500E.

Результати верифікації та синтезу підтверджують коректність всіх трьох розроблених VHDL-моделей. Також показано, що з точки зору

В 31430 нс відбувається натискання кнопки Btn. В 31450 нс автомат
6
використовуваних апаратних ресурсів усі три VHDL-моделі дають приблизно
однакові результати.

Основні положення і результати дослідження були представлені в [14].

Наукова новизна роботи полягає у визначенні методу побудови HDL-опису системи реального часу, характерною особливістю якого є синтез опису в програмовану логічну інтегральну схему, що дозволяє збільшити швидкодію розроблюваних пристроїв за рахунок збільшення витрат часу на розробку інтерфейсу вводу-виводу.

Практична цінність роботи полягає в можливості апаратної реалізації систем реального часу на основі моделі часового автомата, що дає можливість збільшити гнучкість і швидкодію розроблюваних систем.

Розроблений метод побудови HDL-опису систем реального часу може бути використаний в навчальному процесі в курсі “Комп’ютерна логіка”.

Напрямок подальших досліджень може бути використання моделі Мілі для реалізації часових автоматів. Можуть бути проаналізовані можливості розширення часових параметрів автомата Мура. Так, можуть бути розглянуті способи реалізації часового обмеження, представленого множиною часових інтервалів, а також реалізації таймаута, асоційованого з дугами темпорального графа, а не вершинами.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Harel D. Statechars: a Visual Formalism for complex systems // Science of Computer Programming. – 1987. – Vol. 8. – P. 231-274.
2. Alur R. A theory of timed automata / R. Alur, D. L. Dill. A // Theoretical Computer Science. – 1994. – V.126. – N 2. – P. 183-235.
3. Alur R., Timed Automata. In: Halbwachs N., Peled D. (eds) Computer Aided Verification. CAV 1999. Lecture Notes in Computer Science, vol 1633. Springer, Berlin, Heidelberg
4. E.M.Clarke, O. Grumberg, D. Peled. Model checking // MIT Press, 1999.

P.

314.

5. Верификация автоматных программ. / С. Э. Вельдер, М. А. Лукин, А. А. Шалыто, Б. Р. Яминов – СПб: Наука, 2011. – 244 с.
6. Merayo M. G., Núñez M., Rodríguez I. Formal testing from timed finite state machines. // *Computer Networks*. – 2008. – Vol.52(2). – P. 432–460.
7. Merayo M. G., Núñez M., Rodríguez I. Extending efsms to specify and test timed systems with action durations and timeouts. // *IEEE Transactions on Computers*. - 2008. – Vol.57(6). P. 835-844.
8. Bresolin D. Minimizing Deterministic Timed Finite State Machines / D. Bresolin, A. Tvardovskii, N. Yevtushenko, T. Villa, M. Gromov // In 14th IFAC Workshop on Discrete Event Systems WODES 2018. – IFAC-PapersOnLine, 2018. – Vol. 51, issue 7. – P. 486-492.
9. Bresolin D. Deterministic Timed Finite State Machines: Equivalence Checking and Expressive Power / D. Bresolin, K. El-Fakih, T. Villa, N. Yevtushenko // Intern Conf. GANDALF. – 2014. – P. 203-216.
10. Gromov M., Popov D., Yevtushenko N. Deriving test suites for timed Finite State Machines // *Proceedings of IEEE East-West Design & Test Symposium 08, Kharkov: KhNURE, 2008. – P.339-343.*
11. El-Fakih K. Distinguishing Experiments for Timed Nondeterministic Finite State Machines / K. El-Fakih, M. Gromov, N. Shabaldin, N. Yevtushenko // *Acta Cybernetica*. – 2013. – № 2. – P. 205-222.
12. El-Fakih K. A practical approach for testing timed deterministic finite state machines with single clock / K. El-Fakih, N. Yevtushenko, A. Simão // *Science of Computer Programming. Elsevier*. – 2014. – Vol. 80. – P. 343-355.
13. Zhigulin M. FSM-Based Test Derivation Strategies for Systems with TimeOuts / M. Zhigulin, N. Yevtushenko, S. Maag, A.R. Cavalli // *QSIC 2011*. – P. 141149.

14. Shkil A. Design of real-time logic control system on FPGA / M. Miroschnyk, A. Shkil, E. Kulak, D. Rakhlis, I. Filippenko, M. Hoha, M. Malakhov, V. Serhiienko // Proceedings of 2019 IEEE East-West Design & Test Symposium (EWDTS'19), September 13-16, Batumi, Georgia, 2019. – P.489-491.

15. Автоматизированное проектирование систем логического управления с использованием шаблонов автоматного программирования / Шкиль А.С., Кулак Э.Н., Филипенко И.В., Кучеренко Д.Е., Гога М.В, // Радиоэлектроника и информатика : научно-технический журнал / М-во образования и науки Украины
ХНУРЕ. – Харьков, 2018. – № 3. С.75-82

