

УДК 681.326

**АНАЛІЗ АПАРАТУРНИХ ВИТРАТ ПРИ ТЕСТОПРИДАТНОМУ  
ПРОЕКТУВАННІ КЕРУЮЧИХ ЦИФРОВИХ АВТОМАТІВ****Мірошник М.А., Пахомов Ю. В., Шкіль О.С., Кулак Е.М.,  
Кучеренко Д.Ю., Філіппенко І.В.****ANALYSIS OF HARDWARE COSTS DURING THE DESIGN  
FOR TESTABILITY OF CONTROL FINITE STATE MACHINES****Miroschnyk M.A., Pakhomov Y.V., Shkil A.S., Kulak E.N.,  
Kucherenko D.Y., Filippenko I.V.**

*Метою роботи є аналіз апаратурних витрат на забезпечення тестопридатності кінцевих автоматів при різних варіантах організації додаткового переходу між станами автомату в залежності від наявності безумовного переходу, умовного переходу та відсутності переходів між станами автомату, які аналізуються. Висновок про додаткові апаратурні витрати робиться на основі порівняння результатів синтезу тестопридатних HDL-моделей засобами САПР ПЛІС. У роботі вирішено завдання автоматизованого проектування тестопридатних автоматів, що управляють, на основі застосування методів установки автоматів в заданий стан. Оптимальним з точки зору апаратурних витрат способом організації додаткових переходів при установці автоматів, що управляють, у довільний стан, є той перехід, для якого сумарна оцінка апаратурних витрат для функцій збудження мінімальна з урахуванням кодування станів автомата.*

**Ключові слова:** керуючий автомат, таблиця переходів-виходів, мова опису апаратури, автоматизований синтез.

**Вступ.** У сучасному циклі автоматизованого проектування цифрових пристроїв (ЦП) специфікація на пристрій, що проектується, задається, як правило, у вигляді алгоритму функціонування, представленого на одній з мов опису апаратури (Hardware Description Language, HDL). Для такого способу подання ЦП застосування структурних методів діагностування, орієнтованих на виявлення константних несправностей, стає мало ефективним. З іншого боку, застосовувати функціональні методи досить складно з огляду на розмірності сучасних цифрових пристроїв.

Одним з поширених способів вихідного опису спеціалізованих ЦП обробки даних та управління є кінцевий автомат, а формою його подання – таблиця переходів-виходів (ТПВ) і побудований на її основі граф переходів автомата. Зважаючи на складність

діагностичних експериментів (ДЕ) з автоматами, були запропоновані різні методи забезпечення тестопридатності, тобто модифікації автоматних моделей ЦП, які передбачають введення апаратурної надлишковості, як на структурному рівні шляхом введення додаткових входів і виходів для забезпечення простоти проведення ДЕ, так і на функціональному рівні за рахунок внесення доповнень та змін у функціональний опис автомата, а саме, в його ТПВ. Такий підхід дозволяє зробити тестопридатний автомат легкотестованим, тобто таким, для якого завдання забезпечення тестового діагностування вирішуються максимально просто в межах встановлених витрат на проектування. Зробити об'єкт легкотестованим можна шляхом скорочення вартості одного або декількох головних чинників, що визначають трудомісткість тестового діагностування або вартість додаткових витрат апаратури.

Методи підвищення тестопридатності ЦП шляхом внесення в схему реалізацію апаратурної надлишковості досить розвинені та широко використовуються при проектуванні. У класичній праці по розробці і класифікації структурних методів тестопридатного проектування [1] введено поняття «керованості» та «спостережуваності», як основи оцінки структурної тестопридатності. Викладено принципи організації зсувних регістрів в запам'ятовуючій частині ЦП і побудова на їх основі сканованого шляху. Крім того запропоновані модифікації відомих структурних методів побудови тестів при використанні методів сканування. Практичні рекомендації щодо використання структурних методів підвищення тестопридатності ЦП та побудови алгоритмів діагностування на їх основі детально викладені в [2].

У фундаментальній праці [3] розглянуті методи вбудованого самотестування, як основи забезпечення тестопридатності пристроїв, що проектуються. Показано застосування технології BIST (built-in self-test) при проектуванні цифрових систем на ПЛІС. У роботі [4] отримали подальший розвиток ідеї тестопридатного проектування на основі технології граничного сканування (Boundary Scan) з використанням порту JTAG. Розглянуто стандарти граничного сканування IEEE 1149.x, IEEE 1500 і IEEE, а також стандарт тестопридатного проектування IEEE 1687.

Функціональні методи тестопридатного проектування ЦП детально розглянуті в [5]. Для цифрових автоматів, представлених у формі ТПВ, введені поняття діагностованих і визначно-діагностованих класів автоматів та запропоновані способи приведення ТПВ автоматів до зазначених класів. Розглянуто процедури проведення ДЕ над автоматами з використанням установчих, синхронізуючих, діагностичних та характеристичних послідовностей. Крім того, обґрунтована ідея підвищення тестопридатності автомата за рахунок внесення апаратною надмірності шляхом розширення вхідного алфавіту, вихідного алфавіту та алфавіту станів.

В [6] розглянуто структурні моделі кінцевих автоматів, які дозволяють використовувати значення вихідних змінних в якості кодів внутрішніх станів. Даний підхід застосований при синтезі кінцевих автоматів Мілі на ПЛІС, що дозволило знизити вартість реалізації в 1,5-2 рази. Також запропонований спосіб кодування станів дозволяє знизити споживану потужність схемних реалізацій кінцевих автоматів [7].

У роботі [8] з метою підвищення тестопридатності керуючого кінцевого автомату авторами викладена концепція введення апаратної надлишковості в модель абстрактного автомату, яка полягає у додаванні стовпця Sh у таблицю переходів-виходів (ТПВ) (додаткова дуга Sh у графі переходів автомата), що дає змогу встановити автомат в будь-який стан не більше, ніж за  $n-1$  тактів. При  $Sh = 1$  автомат працює в режимі установки в будь-який заданий стан, а при  $Sh = 0$  автомат реалізує заданий алгоритм. При автоматизованому синтезі вказаних автоматів з використанням моделей на мовах опису апаратури (HDL-моделей) введення додаткового стовпця реалізується шляхом додавання сигналу Sh в опис функції переходів у двохпроцесному автоматному шаблоні HDL-моделі. При синтезі це реалізувалося у додаткові мультиплексори, які дозволяли організувати сканування станів автомату в процесі проведення діагностичного експерименту.

Зазначено, що додаткові апаратні витрати при цьому не перевищують 25-30% в залежності від типу автомата та способу кодування його станів, але при цьому використовувався природний порядок обходу станів автомата, що безумовно впливає на додаткові апаратні витрати.

Таким чином актуальною є задача мінімізації додаткових апаратних витрат за рахунок організації оптимального обходу станів автомату.

Мета статті – аналіз апаратних витрат на забезпечення тестопридатності кінцевих автоматів при різних варіантах організації додаткового переходу між станами автомату в залежності від наявності безумовного переходу, умовного переходу та відсутності переходів між станами автомату, які аналізуються. Апаратні витрати аналізуються на основі порівняння результатів синтезу тестопридатності HDL-моделей засобами САПР ПЛІС.

**Аналіз апаратних витрат при автоматизованому проектуванні цифрових автоматів.** У якості абстрактної моделі ЦУ з пам'яттю будемо використовувати кінцевий автомат, який визначається п'ятіркою  $W = \langle X, A, Y, \delta, \lambda \rangle$ , де  $X = \{x_1, x_2, \dots, x_m\}$  – безліч букв вхідного алфавіту;  $A = \{a_1, a_2, \dots, a_n\}$  – множина станів автомата;  $Y = \{y_1, y_2, \dots, y_r\}$  – множина букв вихідного алфавіту;  $\delta(a_i, x_k) = a_j$  – функція переходів автомата,  $\lambda(a_i, x_k) = y_\alpha$  – функція виходів автомата.

При високорівневому проектуванні цифрових пристроїв управління на основі кінцевих автоматів формою подання специфікації проєктованого пристрою є таблиця переходів-виходів (state table) або граф переходів автомата (state diagram). Одним із способів опису моделей ЦП в формі кінцевих автоматів на мові VHDL є автоматний шаблон, тобто спосіб опису моделей керуючих кінцевих автоматів, специфікація на які задана у вигляді ТПВ або графа переходів. Це спеціальна структура VHDL-моделі, в якій функції переходів і виходів виділені в окремі процеси (процес), а призначення нового стану здійснюється у спеціальному процесі, пов'язаному з синхронізацією. При проектуванні тестопридатних керуючих автоматів апаратну надмірність, що забезпечує легкотестованість, доцільно вносити ще на початковому етапі проектування, тобто при побудові HDL-моделей пристроїв, що проектуються. Легкотестованим будемо називати кінцевий автомат, для якого можна побудувати діагностичний експеримент мінімальної довжини шляхом забезпечення встановлення автомата у будь-який стан за мінімальне число тактів.

Таким чином, дана функціональна модель абстрактного автомата у вигляді ТПВ або графа переходів і на його основі будується VHDL-модель у формі автоматного шаблону. Необхідно розглянути різні способи внесення апаратною надлишковості в VHDL-модель для забезпечення тестопридатності, та вибрати оптимальний спосіб з точки зору додаткових апаратних витрат. Апаратна надлишковість в VHDL-моделях забезпечується шляхом внесення в HDL-код додаткових умовних операторів, що забезпечують

побудову сканованого шляху в запам'ятовуючій частині автомата, що підтверджується результатами автоматизованого синтезу. Оптимальним з точки зору внесення додаткових апаратурних витрат будемо вважати той спосіб, який забезпечує мінімальні додаткові апаратурні витрати при оцінці по Квайну вентильного еквівалента схеми, що синтезується в кристали програмованих логічних інтегральних схем (ПЛІС) в автоматичному режимі інструментальними засобами систем автоматизованого проектування (САПР) ЦП.

Розглянемо можливі варіанти додавання переходу з сигналом Sh між вершинами графу переходів (станів автомату)  $a_i$  та  $a_j$ .

Перший варіант полягає в додаванні Sh до безумовного переходу ( $a_0 \rightarrow a_1$ ) у автоматі Мура. Відмітимо, що для використання наведених фрагментів графу переходів в процесі автоматизованого синтезу у графі мають враховуватися усі варіанти переходів та зворотна дуга (що забезпечує відсутність вершин, які «висять», рис. 1(а)). ТПВ фрагментів автомату для початкового графу та графу з Sh наведені на рис.1 (б). В результаті між вершинами виявляються дві дуги з сигналами Sh та  $\overline{Sh}$  (рис. 1 (в)).

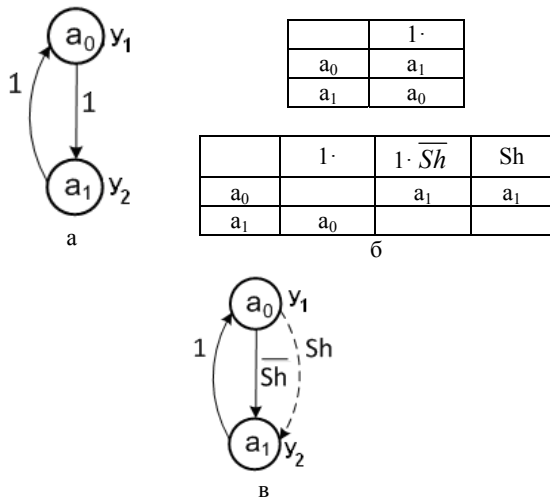


Рис. 1. Додавання Sh до безумовного переходу

В даному випадку функція переходу в стан  $a_1$  матиме вигляд:  $a_1 = a_0Sh \vee a_0\overline{Sh} = a_0$ .

Застосовуючи закон склеювання, отримуємо  $a_0$ , що означає відсутність додаткових витрат апаратури при реалізації мінімізованого виразу. Для кодування двох станів автомату використовується один D-тригер,  $a_0$  кодується 0,  $a_1$  кодується 1. Таким чином, вираз  $a_1 = a_0$  фактично реалізується інвертором.

Розглянемо VHDL-модель наведеного фрагменту автомату (рис. 2) та виконаємо її автоматизований синтез (рис. 3).

Нижче наведені результати синтезу даної VHDL-моделі з використанням пакету САПР: XILINX ISE 10.1, плата Spartan 3E, мікросхема FPGA XC3S500E, Package FG 320. На рис. 3

наведена схемна реалізація автомату, яка підтверджує попередні викладки.

```
library IEEE;
use IEEE.std_logic_1164.all;
entity FSM is port (Sh, Reset, Clk: in STD_LOGIC;
y1, y2: out STD_LOGIC); end;

architecture Moore of FSM is
    type State_type is (a0, a1);
    signal State, NextState: State_type;
begin
    Sreg0_CurrentState: process (Clk, reset) begin
        if Reset='1' then State <= a0;
        elsif Clk'event and Clk = '0' then
            State <= NextState;    end if;
    end process;
    Sreg0_NextState: process (State, Sh) begin
        case State is
            when a0=> if sh='1' then NextState <= a1;
                       else NextState <= a1;    end if;
            when a1=> NextState <= a0;
            when others => NextState <= a0;
        end case;
    end process;
    y1 <= '1' when State=a0 else '0';
    y2 <= '1' when State=a1 else '0';
end;
```

Рис. 2. VHDL-модель фрагменту автомату з безумовним переходом

*Synthesizing Unit <FSM>. Related source file is "C:/Temp/shkil2018/shkil\_2states\_2018/fsm.vhd".*  
**WARNING:Xst:647 - Input <Sh> is never used.**  
*This port will be preserved and left unconnected if it belongs to a top-level block or it belongs to a sub-block and the hierarchy of this sub-block is preserved.*  
*Found 1-bit register for signal <State<0>>.*

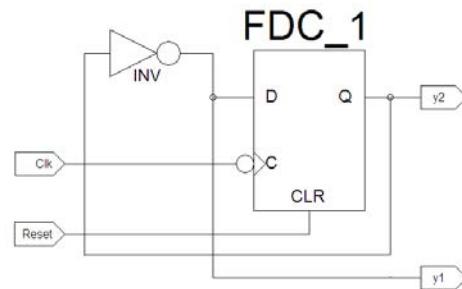


Рис. 3. Результати синтезу VHDL-моделі автомату з безумовним переходом

Другий варіант полягає в додаванні Sh до переходів між  $a_i$  і  $a_j$  при відсутності переходів в інші стани, відмінні від  $a_j$ . Ця ситуація характерна для автомата Мілі, але для аналізу цієї ситуації будемо використовувати змішаний автомат на три стани, у якого стани  $a_1$  та  $a_2$  відображаються моделлю Мура, а переходи ( $a_0 \rightarrow a_1$ ) – моделлю Мілі. Граф переходів вказаного автомата зображений на рис. 4 (а), а його ТПВ – на рис. 4 (б). На рис. 4 (б) в ТПВ для спрощення сигнал  $\overline{Sh}$  не вказується. Граф

переходів автомату з сигналами Sh та  $\overline{Sh}$  наведений на рис. 4 (в).

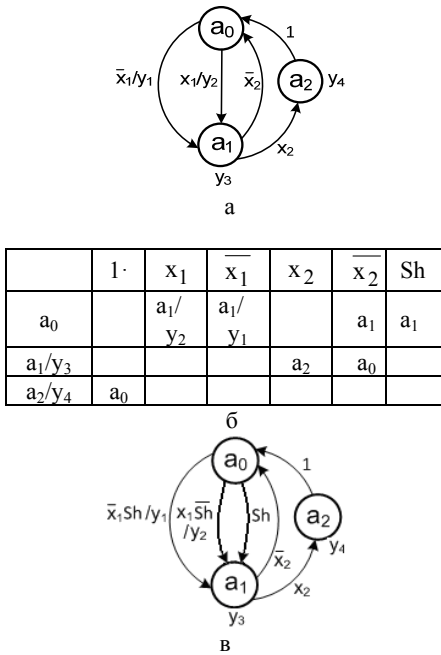


Рис. 4. Додавання Sh при наявності переходів тільки між a<sub>0</sub> та a<sub>1</sub>

Функція переходу в стан a<sub>1</sub> для двох вихідних переходів і доданого переходу Sh буде мати вигляд:  $a_1 = a_0Sh \vee a_0x_1\overline{Sh} \vee a_0\overline{x_1}Sh = a_0Sh \vee a_0\overline{Sh} = a_0$ .

Застосовуючи поетапно закон склеювання, отримуємо a<sub>0</sub>, що означає відсутність додаткових витрат апаратури при реалізації мінімізованого виразу. Для будь-якого числа дуг, укладених між a<sub>i</sub> і a<sub>j</sub>, результат буде таким же. При реалізації вихідного виразу додаткові апаратурні витрати – додатковий вхід  $\overline{Sh}$  в усі вентиля, що реалізують терми вихідної функції переходів (рис. 5 (а)) і вентиль, який реалізує терм a<sub>i</sub>Sh (рис. 5 (б)).

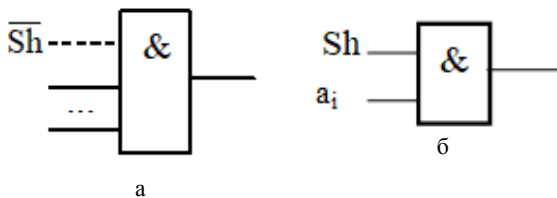


Рис. 5. Додаткові апаратурні витрати при наявності переходів тільки між a<sub>0</sub> та a<sub>1</sub>

Розглянемо VHDL-модель наведеного фрагменту автомату (рис. 6) та виконаємо її автоматизований синтез (рис. 7). Для зменшення обсягу VHDL-коду в подальшому доцільно наводити тільки фрагменти архітектури VHDL-моделей, а саме, тільки функції переходів.

```

Sreg0_NextState: process (State, Sh, x1, x2)
begin
  case State is
    when a0=>if sh='1' then nextState <= a1;
              elsif x1='1' then nextState <= a1; y2 <= '1';
              else nextState <= a1; y1 <= '1';
              end if;
    when a1=>if x2='1' then nextState <= a2;
              else nextState <= a0;
              end if;
    when a2=> nextState <= a0;
    when others => nextState <= a0;
  end case;
end process;

y3 <= '1' when State=a0 else '0';
y4 <= '1' when State=a1 else '0';
end;
    
```

Рис. 6. Фрагмент VHDL-моделі автомату з переходами тільки між a<sub>0</sub> та a<sub>1</sub>

Found finite state machine <FSM\_0> for signal <State>.  
 Analyzing FSM <FSM\_0> for best encoding.  
 Optimizing FSM <State/FSM> on signal <State[1:2]> with gray encoding.  
 State | Encoding a0 | 00 a1 | 01 a2 | 11.

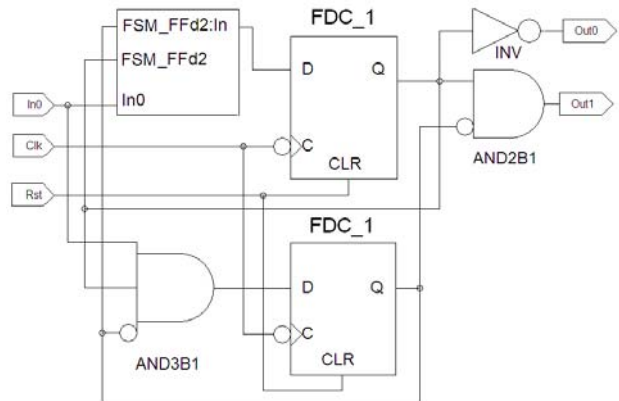


Рис. 7. Результати синтезу VHDL-моделі автомату з переходами тільки між a<sub>0</sub> та a<sub>1</sub>

Отримана схема підтверджує, що сигнал Sh (In0) через вентиль AND додається до функції збудження кожного з тригерів.

Третій варіант полягає в додаванні Sh до переходів між a<sub>i</sub> і a<sub>j</sub> при наявності переходів і в інші стани, відмінні від a<sub>j</sub> (рис. 8 (а)).

При цьому функція переходу в стан a<sub>i</sub> (a<sub>1</sub>) для одного вихідного переходу, доданого переходу Sh та переходу в стан a<sub>k</sub> ≠ a<sub>j</sub> (рис. 8 (а)) матиме вигляд:

$$a_j = a_iSh \vee a_ix_1\overline{Sh} = a_i(Sh \vee x_1\overline{Sh}) = a_iSh \vee a_ix_1$$

Вираз a<sub>i</sub>Sh ∨ a<sub>i</sub>x<sub>1</sub> отримаємо застосовуючи закон Блейка-Порецького. Це означає, що додаткові апаратурні витрати при реалізації мінімізованого виразу для функції переходу в стан a<sub>j</sub> – вентиль, який реалізує терм a<sub>i</sub>Sh (рис. 5 (б)). При реалізації вихідного виразу для функції переходу в a<sub>j</sub> додаткові апаратурні витрати – вентиль, який

реалізує терм  $a_i Sh$  плюс додатковий вхід в вентиль, який реалізує терм вихідної функції переходів (рис. 5 (а) та 8 (в)). Якщо переходів  $a_i \rightarrow a_j$  в вихідному графі більше одного, то для кожного з цих переходів додається додатковий вхід в вентиль.

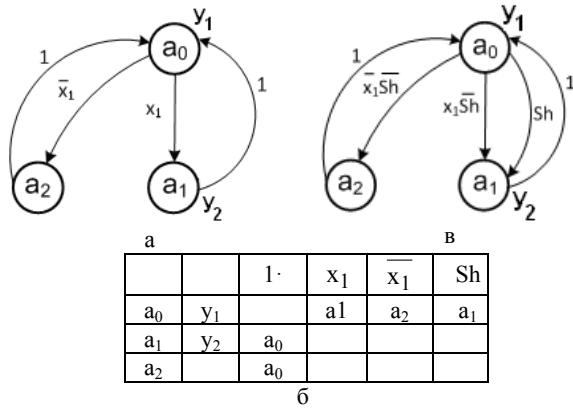


Рис. 8. Додавання Sh при наявності переходів між  $a_0$  і  $a_1$  та інших вихідних переходів

Крім того, в даному варіанті,  $\forall(a_i \rightarrow a_k)$ ,  $a_k \neq a_j$  переходи  $a_i \rightarrow a_k$  також несуть додаткові апаратні витрати у вигляді додаткового входу  $\overline{Sh}$  у вентиль, який реалізує терм вихідної функції переходів. Для прикладу на рис. 8 (в)  $a_k = a_i x_1 \overline{Sh}$ .

Розглянемо VHDL-модель наведеного фрагменту автомату (рис. 9) та виконаємо її автоматизований синтез (рис. 10).

```

Sreg0_NextState: process (State, Sh, x)
begin
  case State is
    when a0=> if sh='1' then NextState <= a1;
               elsif x='1' then NextState <= a1;
               else NextState <= a2;
               end if;
    when a1=> NextState <= a0;
    when a2=> NextState <= a0;
    when others => NextState <= a0;
  end case;
end process;
y1 <= '1' when State=a0 else '0';
y2 <= '1' when State=a1 else '0';
end;
    
```

Рис. 9. Фрагмент VHDL-моделі автомату при наявності переходів  $a_0 \rightarrow a_1$  та інших переходів

Четвертий варіант полягає в додаванні переходу з Sh між  $a_i$  і  $a_j$  при відсутності переходу  $a_i \rightarrow a_j$  взагалі що показано на рис. 11 (а) та рис. 11 (в).

Тут додаткові апаратні витрати – вентиль, який реалізує терм  $a_i Sh$  плюс додатковий вхід  $\overline{Sh}$  у вентилі, що реалізують  $\forall(a_i \rightarrow a_k)$ ,  $a_k \neq a_j$  терм вихідної функції переходів. Слід також зазначити, що інші дуги  $\forall(a_s \rightarrow a_j)$ ,  $a_s \neq a_i$  на переходи  $a_s \rightarrow a_j$  Sh не впливає.

Analyzing FSM <FSM\_0> for best encoding.  
 Optimizing FSM <State/FSM> on signal <State[1:2]> with speed1 encoding.  
 State | Encoding  $a_0$  | 10  $a_1$  | 01  $a_2$  | 00.

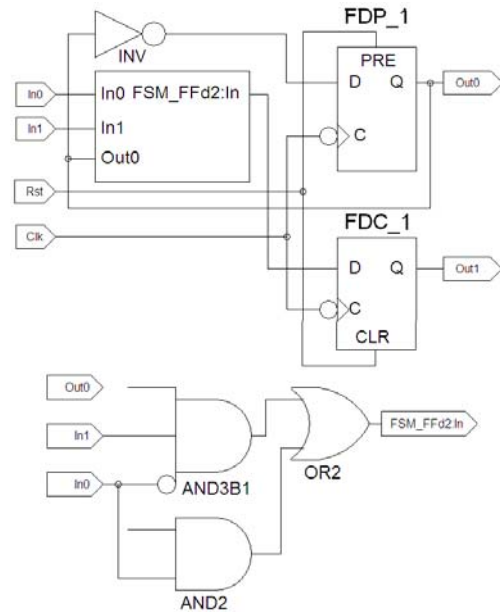


Рис. 10. Результати синтезу VHDL-моделі автомату при наявності переходів між  $a_0$  і  $a_1$  та інших вихідних переходів

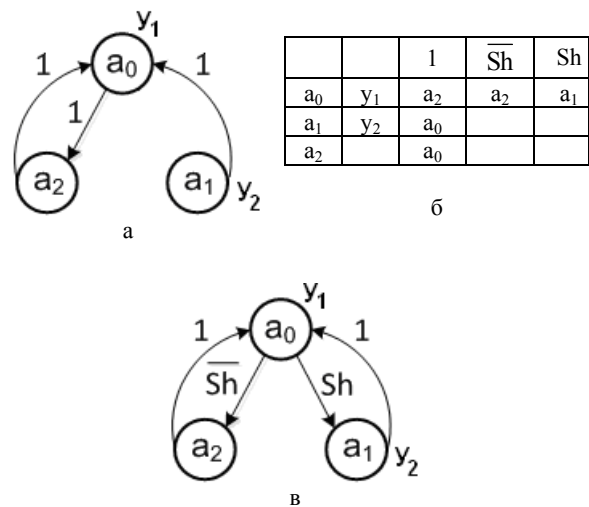


Рис. 11. Додавання Sh при відсутності переходів між  $a_0$  і  $a_1$

Розглянемо VHDL-модель наведеного фрагменту автомату (рис. 12) та виконаємо її автоматизований синтез (рис. 13).

Підсумовуючи вищевикладене можна визначити наступне.

1) Для вибору шляху призначення додаткових переходів з Sh однозначно з точки зору мінімізації витрат апаратури по спадаючому пріоритету йде 1 варіант, за ним другий, а потім 3 або 4.

2) При обранні 1-го або 2-го варіанту при реалізації не мінімізованого виразу функції переходів, краще обирати 1-й, так він менше за кількістю вихідних дуг, а значить, буде менше

додаткових входів  $\overline{Sh}$  в вентилі. При реалізації мінімізованого виразу функції переходів, немає різниці 1-й або 2-й, в обох випадках немає додаткових витрат апаратури.

```

Sreg0_NextState: process (State, Sh)
begin
  case State is
    when a0=> if sh='1' then NextState <= a1;
               else NextState <= a2;
               end if;
    when a1=> NextState <= a0;
    when a2=> NextState <= a0;
    when others => NextState <= a0;
  end case;
end process;
y1 <= '1' when State=a0 else '0';
y2 <= '1' when State=a1 else '0';
end;
    
```

Рис. 12. Фрагмент VHDL-моделі автомату при відсутності переходів між  $a_0$  і  $a_1$

Optimizing FSM <State/FSM> on signal <State[1:2]> with speed1 encoding.  
 State | Encoding : a0 | 10 a1 | 01 a2 | 00.

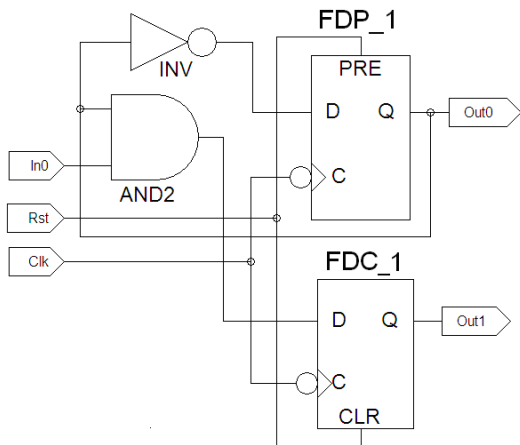


Рис. 13. Результати синтезу VHDL-моделі автомату при відсутності переходів між  $a_0$  і  $a_1$

3) Для обрання між 3 і 4 варіантами визначальним швидше буде число вихідних дуг, що задовольняють умові:  $\forall(a_i \rightarrow a_k), a_k \neq a_j$ . В якому варіанті менше, такий і вибирати, оскільки додатковий вентиль на перехід  $a_i \rightarrow a_j$  з  $Sh$  йде в обох випадках. Це для способу реалізації мінімізованого виразу функції переходів.

Для способу реалізації не мінімізованого виразу функції переходів потрібно в варіанті 3 для підрахунку числа дуг для порівняння з варіантом 4 до умови:  $\forall(a_i \rightarrow a_k), a_k \neq a_j$ , додати  $\forall(a_i \rightarrow a_j)$

крім  $a_i \rightarrow a_j$  з  $Sh$ .

**Тестопридатне проектування керуючого автомату.** В якості прикладу побудови тестопридатного керуючого автомату (КА) розглядатимемо мікропрограмний автомат (МПА)

складання чотирирозрядних двійкових знакових чисел в додатковому модифікованому коді. Для цієї мікропрограми в якості КА розглядається автомат Мура. На рис. 14 (а) показаний фрагмент змістовної граф-схеми алгоритму (ГСА) вказаної мікропрограми з відмітками станів КА для автомата Мура. Кожній операторній вершині ГСА (стану автомата Мура) відповідає набір сигналів  $y_i$ , що керують, кожен з яких ініціює виконання певної мікрооперації операційного автомату. Для спрощення подальшого викладу в цьому фрагменті мікропрограми не аналізується результат складання на переповнювання розрядної сітки, що в цілому не міняє структури мікропрограми. На рис. 14 (б) наведений змістовний граф переходів КА Мура для цієї мікропрограми.

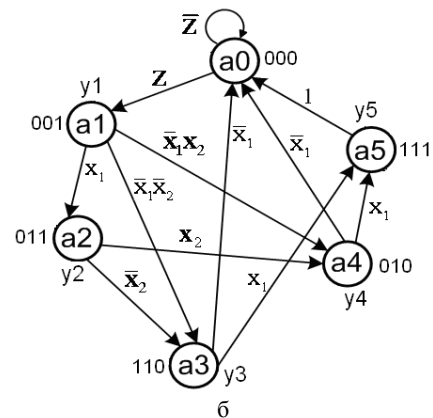
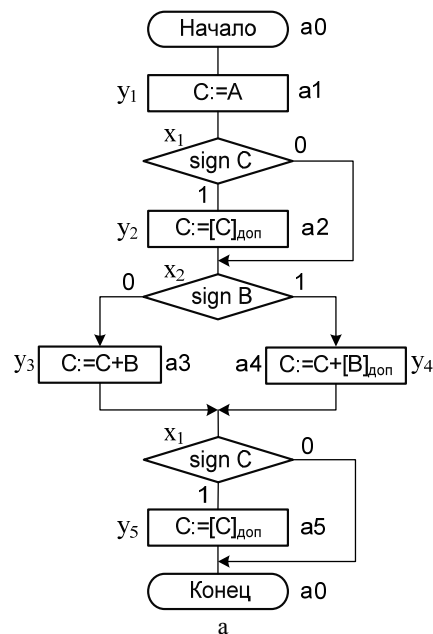


Рис. 14. Фрагмент ГСА МПА складання (а) та граф переходів його КА (б)

Таким чином при обранні пари станів автомата, між якими встановлюється додатковий перехід (дуга  $Sh$  у графі переходів автомату), обирається той стан-наступник, для якого сумарна оцінка апаратних витрат для функцій збудження мінімальна з

урахуванням кодування станів автомата. При цьому відсутність переходів між станами прирівнюється до переходу з однією умовою.

На рис. 15 (а) наведені таблиця переходів-виходів зі стовбцем Sh та граф переходів з додатковими дугами Sh (рис. 15 (б)) при організації обходу вершин графу за природною нумерацією (перший варіант), а на рис. 16 – другий варіант з урахуванням складності функцій збудження.

a/y	l	Z	x <sub>1</sub>	$\bar{x}_1$	x <sub>2</sub>	$\bar{x}_2$	$\bar{x}_1x_2$	$\bar{x}_1\bar{x}_2$	Sh
a0		a1							a1
a1	y1		a2				a3	a4	a2
a2	y2				a4	a3			a3
a3	y3		a5	a0					a4
a4	y4		a5	a0					a5
a5	y5	a0							a0

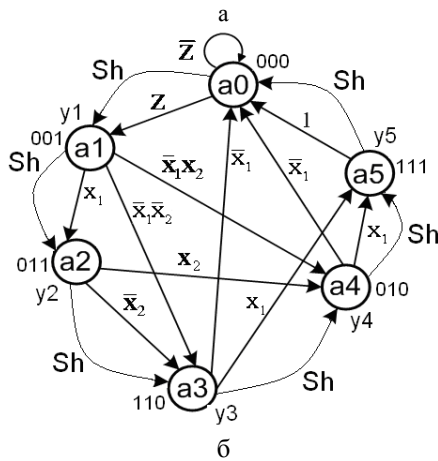


Рис. 15. Перший варіант ТПВ тестопридатного керуючого МПА (а) та його граф переходів (б)

a/y	l	Z	x <sub>1</sub>	$\bar{x}_1$	x <sub>2</sub>	$\bar{x}_2$	$\bar{x}_1x_2$	$\bar{x}_1\bar{x}_2$	Sh
a0		a1							a1
a1	y1		a2				a3	a4	a2
a2	y2				a4	a3			a4
a3	y3		a5	a0					a5
a4	y4		a5	a0					a3
a5	y5	a0							a0

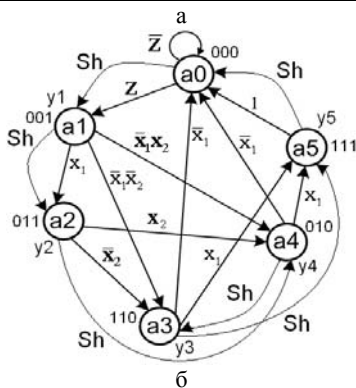


Рис. 16. Другий варіант ТПВ тестопридатного керуючого МПА (а) та його граф переходів (б)

Розглянемо VHDL-моделі тестопридатних МПА з різним способом розміщення Sh (рис. 15 та 16)) та виконаємо її автоматизований синтез. Для скорочення місця наведемо VHDL-модель для першого варіанту та на ній вкажемо зміни, характерні для другого варіанту (рис. 17).

```

architecture Moore of FSM is
    type State_type is (a0, a1, a2, a3, a4, a5);
    signal State, NextState: State_type;
begin
    Sreg0_CurrentState: process (Clk, reset)
    begin
        if Reset='1' then State <= a0;
        elsif Clk'event and Clk = '0' then State <=
        NextState;
        end if;
    end process;
    Sreg0_NextState: process (State, x1, x2, Z, Sh)
    begin
        case State is
            when a0=> if sh='1' then NextState <= a1;
            elsif Z='1' then NextState <= a1;
            else NextState <= a0;
            end if;
            when a1=> if sh='1' then NextState <= a2;
            elsif x1='1' then NextState <= a2;
            elsif x2='1' then NextState <= a4;
            else NextState <= a3;
            end if;
            when a2=> if sh='1' then NextState <= a3; -- (a4)
            elsif x2='1' then NextState <= a4;
            else NextState <= a3;
            end if;
            when a3=> if sh='1' then NextState <= a4; -- (a5)
            elsif x1='1' then NextState <= a5;
            else NextState <= a0;
            end if;
            when a4=> if sh='1' then NextState <= a5; -- (a3)
            elsif x1='1' then NextState <= a5;
            else NextState <= a0;
            end if;
            when a5=> if sh='1' then NextState <= a0;
            else NextState <= a0;
            end if;
            when others => NextState <= a0;
        end case;
    end process;
    y1 <= '1' when State=a1 else '0';
    y2 <= '1' when State=a2 else '0';
    y3 <= '1' when State=a3 else '0';
    y4 <= '1' when State=a4 else '0';
    y5 <= '1' when State=a5 else '0';
end;
    
```

Рис. 17. Архітектура VHDL-моделі тестопридатного керуючого МПА складання

На рис. 18 наведена часова діаграма моделювання діагностичного експерименту (ДЕ) по обходу станів керуючого автомата за першим варіантом розміщення додаткових дуг Sh.

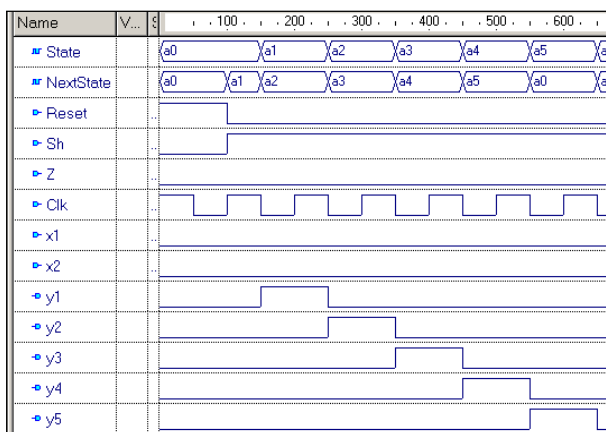


Рис. 18. Часова діаграма моделювання ДЕ по обходу станів керуючого автомату

На рис. 19 наведений протокол автоматизованого синтезу зазначених VHDL-моделей. Відмітимо, що кодування станів для всіх варіантів є незмінним.

*Analyzing FSM <FSM\_0> for best encoding.  
Optimizing FSM <State/FSM> on signal  
<State[1:3]> with gray encoding.  
State | Encoding : a0| 000 a1| 001 a2| 011  
a3| 110 a4| 010 a5| 111.*

Рис. 19. Протокол автоматизованого синтезу VHDL-моделей

В аналітичному вигляді функція збудження триггеру першого розряду (при першому варіанті розміщення дуг Sh) буде:

$$D_1 = (Q_1 \bar{Q}_3 \vee \bar{Q}_1 Q_2 \bar{Q}_3) x_1 \bar{Sh} \vee \bar{Q}_1 Q_2 \bar{Q}_3 x_2 \bar{Sh} \vee \vee (\bar{Q}_1 Q_2 \bar{Q}_3 \vee \bar{Q}_1 Q_2 Q_3) \bar{Sh} \vee \bar{Q}_2 Q_3 x_1 x_2 \bar{Sh}$$

а для другого варіанту:

$$D_1 = (Q_1 \bar{Q}_3 \vee \bar{Q}_1 Q_2 \bar{Q}_3) x_1 \bar{Sh} \vee \bar{Q}_1 Q_2 \bar{Q}_3 x_2 \bar{Sh} \vee \vee (\bar{Q}_1 Q_2 \bar{Q}_3 \vee \bar{Q}_1 Q_2 Q_3) \bar{Sh} \vee \bar{Q}_2 Q_3 x_1 x_2 \bar{Sh}$$

Таким чином, зменшення буде на 1 вхід вентиля та 1 інвертор, тобто на 4 % при оцінці по Квайну.

Загальні апаратні витрати по Квайну при синтезі для схеми без Sh – 133, при першому варіанті схеми з Sh за умови кодування станів КА, яке формує САПР, – 184, а при другому варіанті – 187, що обумовлює збільшення апаратних витрат відповідно на 40% та 38%. Проведений аналіз показує, що при обранні переходу розміщення Sh слід враховувати не тільки складність функції збудження, а й кодування станів автомату.

**Висновки.** У роботі представлений метод автоматизованого проектування тестопридатних керуючих автоматів шляхом внесення апаратної надлишковості. Модель автомата представлена на мові VHDL в формі автоматного шаблону.

Способом вирішення є внесення додаткових фрагментів VHDL-коду, що забезпечують примусову установку автомата в довільний стан. Запропоновано метод розширення таблиці переходів-виходів автомата, який забезпечує режим обходу всіх вершин графа переходів автомата (станів) в режимі діагностування. Проаналізовані апаратні витрати при різних варіантах організації додаткового переходу між станами автомату в залежності від наявності безумовного переходу, умовного переходу та відсутності переходів між станами автомата, які аналізуються. При обранні додаткового переходу (дуга Sh у графі переходів автомата), обирається той стан-наступник, для якого сумарна оцінка апаратних витрат для функцій збудження мінімальна з урахуванням кодування станів автомата. Зазначений підхід підвищує керуваність станів автомата, що значно покращує його тестопридатність. Моделювання розширених VHDL-моделей автомата засобами Active-HDL підтвердило працездатність даного підходу. Синтез даних моделей засобами САПР XILINX ISE підтвердив отримання тестопридатних структур та підтвердив отримання мінімальних апаратних витрат запропонованого методу.

Наукова новизна роботи полягає в подальшому розвитку методу оптимізації апаратних витрат при підвищенні тестопридатності кінцевих автоматів за рахунок розширення вхідного алфавіту в HDL-моделях в формі автоматного шаблону, що дало можливість автоматизувати процес проектування тестопридатних автоматів з мінімальними апаратними витратами.

Практична цінність отриманих результатів полягає в розробці процедур внесення надлишковості та розширення вхідного алфавіту в HDL-моделях кінцевих автоматів в формі автоматного шаблону шляхом внесення додаткових умовних операторів в HDL-код, які забезпечують установку автомата в довільний стан при мінімальних апаратних витратах. Розроблені процедури можуть бути застосовані при розробці додаткового програмного модулю САПР цифрових пристроїв, який буде в автоматизованому режимі формувати HDL-код тестопридатного кінцевого автомата.

#### Література

1. Беннеттс Р.Дж. Проектирование тестопригодных логических схем: пер. с англ. / Р. Дж. Беннеттс. – М.: Радио и связь, 1990. – 176 с.
2. Горяшко А.П. Проектирование легко тестируемых дискретных устройств: идеи, методы, реализация / А.П. Горяшко // Автоматика и телемеханика. – 1984. – № 7. – С. 5-35.
3. Stroud С.Е. A designer's guide to built-in self-test / Charles E. Stroud. – Kluwer Academic Publishing, 2002. – 319 p.
4. Городецкий А. Введение в технологии JTAG и DFT. Тестирование в технологиях граничного сканирования и тестопригодное проектирование / Ами Городецкий //



- Palmarium Academic Publishing, Germany, 2012. – 308 с.
- Тоценко В.Г. Алгоритмы технического диагностирования цифровых устройств / В.Г. Тоценко. – М.: Радио и связь, 1985. – 240 с.
  - Solov'ev V.V. Minimization of mealy finite-state machines by using the values of the output variables for state assignment / V.V. Solov'ev // Journal of Computer and Systems Sciences International. – January 2017. – Volume 56, Issue 1. – P. 96-104.
  - Solov'ev V.V. Minimization of Power Consumption of Finite State Machines by Splitting Their Internal States / T.N.Grzes, V.V. Solov'ev // Journal of Computer and Systems Sciences International. – 2015. – Vol. 54, No. 3. – P. 367-374.
  - Shkil A. Design Automation of Testable Finite State Machines / M.Miroschnyk, Y. Pakhomov, E. German, A. Shkil, E. Kulak, D. Kucherenko // Proceedings of the International Sympos. EWDTs'2017, September 29-October 2, 2017 – Novi Sad, Serbia, 2017. – P.203-208.

### References

- Bennetts R.G. Proektirovanie testoprigradnyh logicheskikh shem: per. s angl. / R. G. Bennetts. – M.: Radio i svjaz', 1990. – 176 p.
- Gorjashko A.P. Proektirovanie legko testiruemykh diskretnykh ustrojstv: idei, metody, realizacija / A.P. Gorjashko // Avtomatika i telemekhanika. – 1984. – № 7. – P. 5-35.
- Tocenko V.G. Algoritmy tehničeskogo diagnostirovanija cifrovyyh ustrojstv / V.G. Tocenko. – M.: Radio i svjaz', 1985. – 240 p.
- Stroud C.E. A designer's guide to built-in self-test / Charles E. Stroud. – Kluwer Academic Publishing, 2002. – 319 p.
- Gorodetsky A. Introduction to JTAG and DFT technology. Testing in edge scanning technologies and testable design / A. Gorodetsky // Palmarium Academic Publishing. – Germany, 2012. – 308 p.
- Solov'ev V.V. Minimization of mealy finite-state machines by using the values of the output variables for state assignment / V.V. Solov'ev // Journal of Computer and Systems Sciences International. – January 2017. – Volume 56, Issue 1. – P. 96-104.
- Solov'ev V.V. Minimization of Power Consumption of Finite State Machines by Splitting Their Internal States / T.N.Grzes, V.V. Solov'ev // Journal of Computer and Systems Sciences International. – 2015. – Vol. 54, No. 3. – P. 367-374.
- Shkil A. Design Automation of Testable Finite State Machines / M.Miroschnyk, Y. Pakhomov, E. German, A. Shkil, E. Kulak, D. Kucherenko // Proceedings of the International Sympos. EWDTs'2017, September 29-October 2, 2017 – Novi Sad, Serbia, 2017. – P.203-208.

**Мірошник М.А., Пахомов Ю. В., Шкіль А.С., Кулак Э.Н., Кучеренко Д.Е., Филиппенко И.В.**

*Целью работы является анализ аппаратурных затрат на обеспечение тестопригодности конечных автоматов при различных вариантах организации дополнительного перехода между состояниями автомата в зависимости от наличия безусловного перехода, условного перехода и отсутствия переходов между состояниями автомата, которые анализируются. Вывод по дополнительным аппаратурным затратам делается на основе сравнения результатов синтеза*

*тестопригодных HDL-моделей средствами САПР ПЛИС. В работе решена задача автоматизированного проектирования тестопригодных управляющих автоматов на основе применения методов установки автоматов в заданное состояние. Оптимальным с точки зрения аппаратурных затрат способом организации дополнительных переходов при установке управляющих автоматов в произвольное состояние, является тот переход, для которого суммарная оценка аппаратурных затрат для функций возбуждения минимальная с учетом кодирования состояний автомата.*

**Ключевые слова:** управляющий автомат, таблица переходов-выходов, язык описания аппаратуры, автоматизированный синтез.

**Miroschnyk M.A., Pakhomov Y.V., Shkil A.S., Kulak E.N., Kucherenko D.Y., Filippenko I.V. Analysis of hardware costs during the design for testability of control finite state machines.**

*The aim of the work is to analyze hardware costs of ensuring the testability of finite state machines with various options for organizing an additional transition between FSM's states depending on the presence of an unconditional transition, a conditional transition, and the absence of transitions between states of analyzed FSM. The conclusion on additional hardware costs is made on the basis of a comparison of synthesis results of testable HDL-models by means of CAD FPGA. The paper solved the problem of computer-aided design of testable control FSM based on the application of methods for setting FSM in a given state. The best way to organize additional transitions during setting of control FSM in an arbitrary state is the transition for which the total hardware cost estimate for the excitation functions is minimal, taking into account the coding of FSM's states.*

**Key words:** control FSM, state table, hardware description language, automated synthesis.

**Мірошник Марина Анатоліївна**, д-р техн. наук, зав. кафедрой інформаційних технологій Українського державного університету залізничного транспорту. E-mail: marinagmiro@gmail.com.

**Пахомов Юрій Васильович**, асистент кафедри експлуатації газових і теплових систем Харківського національного університету міського господарства. E-mail: abc050073@gmail.com.

**Шкіль Олександр Сергійович**, канд. техн. наук, доцент кафедри АПОТ ХНУРЕ. Наукові інтереси: діагностика цифрових систем, дистанційна освіта. E-mail: oleksandr.shkil@nure.ua.

**Кулак Ельвіра Миколаївна**, канд. техн. наук, доцент кафедри АПОТ ХНУРЕ. Наукові інтереси: автоматизоване проектування цифрових автоматів, мови опису апаратури. E-mail: elvira.kulak@nure.ua.

**Філіппенко Інна Вікторівна**, канд. техн. наук, доцент кафедри АПОТ ХНУРЕ. Наукові інтереси: проектування цифрових пристроїв на базі мікроконтролерів, цифрові фільтри. E-mail: inna.filippenko@nure.ua.

**Кучеренко Дарія Юхимівна**, канд. техн. наук, доцент кафедри АПОТ ХНУРЕ. Наукові інтереси: мови опису апаратури, експертні системи, технічна діагностика, нечітка логіка. E-mail: dariia.kucherenko@nure.ua.

*Рецензент:* д.т.н., проф. **Рязанцев О.І.**

Стаття подана 29.11.2018