

С. С. Штаненко¹, Ю. Я. Самохвалов², О. Ю. Іохов³, В. Г. Малюк⁴

¹ Військовий інститут телекомунікацій та інформатизації імені Героїв Крут, Київ, Україна

² Київський національний університет імені Тараса Шевченка, Київ, Україна

³ Національна академія Національної гвардії України, Харків, Україна

МІКРОПРОЦЕСОРНІ СИСТЕМИ НА ПРОГРАМОВАНИХ ЛОГІЧНИХ ІНТЕГРАЛЬНИХ СХЕМАХ ЯК ОБ'ЄКТ ДІАГНОСТИКИ

Анотація. Предметом дослідження в статті є методи тестування цифрових пристроїв, які реалізовані на програмованих логічних інтегральних схемах (ПЛІС). **Метою роботи** є обґрунтування підходу до діагностування мікропроцесорних систем у базисі ПЛІС, який являє собою подальший розвиток методу самодіагностики. **У статті вирішуються наступні завдання:** проведено аналіз існуючих методів тестування цифрових пристроїв, розкрито їх переваги та недоліки; проаналізовано існуючі підходи до діагностування мікропроцесорних систем, основу яких складають інтегральні схеми із програмованою структурою; запропоновано підхід до діагностування мікропроцесорних систем в базисі ПЛІС, в основу якого покладено принцип взаємодії процесорів між собою шляхом введення в багатопроцесорну систему сервісного процесора. **Отримані наступні результати:** доведено, що реалізація запропонованих моделей взаємодії процесорів між собою в багатопроцесорній системі надасть сервісному процесору діагностичну інформацію про технічний стан системи. Зазначено, що наявність діагностичної інформації про технічний стан мікропроцесорної системи є основою для прийняття рішення на відновлення системи шляхом перепрограмування ПЛІС. **Висновки:** розглянуто проблему діагностування мікропроцесорних систем у базисі ПЛІС. Проведено аналіз існуючих методів тестування інтегральних схем із програмованою структурою, відзначено їх переваги та недоліки. Запропоновано в якості діагностичного пристрою використовувати вбудований сервісний процесор, основною функцією якого є збір діагностичної інформації та прийняття рішення про реконфігурацію мікропроцесорної системи з метою оперативного автоматичного відновлення її функціонування.

Ключові слова: мікропроцесорна система; програмована логічна інтегральна схема; діагностика цифрових пристроїв.

Вступ

Стрімкий розвиток мікроелектронної технології в даний час дозволило серійно випускати великі і надвеликі інтегральні схеми (ВІС/НВІС), що містять тисячі елементів на кристалі і мають широкі функціональні можливості.

Одним з найбільш перспективних ВІС/НВІС, з точки зору досягнення найбільшої щільності розміщення елементів на кристалі, є інтегральні схеми (ІС), що мають регулярну структуру. Прагнення створити універсальну ВІС/НВІС, не вводячи значної структурної надмірності, призвело до появи спеціалізованих ІС (*ASIC – Application-Specific Integrated Circuit*), які мають індивідуальний характер функціонування та розробляються (проектуються) за конкретним замовленням. У цьому напрямі найважливішим досягненням стала поява ІС з програмованою структурою – програмованих логічних інтегральних схем (ПЛІС). ПЛІС представляє собою матрицю програмованих логічних елементів з *SPLD (Simple Programmable Logic Devices)*, *CPLD (Complex Programmable Logic Device)*, *FPGA (Field-Programmable Gate Array)*, *FLEX (Flexible Logic Element Matrix)* структурами. За допомогою даних структур, застосовуючи на нижньому рівні (рівень регістрових передач) мови опису апаратури *AHDL*, *VHDL*, *Verilog*, на середньому (блочному) – технологію *System-on-Chip* та верхньому – високорівневі мови програмування *C/C++*, *System C*, *Python*, *Java*, ми отримуємо можливість проектувати не лише цифрові пристрої, а й мікропроцесорні системи.

На думку [1–4] саме застосування ПЛІС в якості елементної бази побудови сучасних мікропроце-

сорних систем на сьогоднішній день є перспективним напрямком з точки зору проектування високонадійних, відмовостійких (живучих) складних технічних систем, що функціонують в умовах несприятливого впливу, у тому числі і кібератак.

При проектуванні, виготовленні та подальшій експлуатації таких складних технічних пристроїв, якими є ІС, неминучі різного роду відхилення, наприклад, помилки проектування, несправності (дефекти) виготовлення, які в процесі функціонування призводять до відмови елементів та вузлів системи. Більшість цих відхилень, такі як, помилки проектування, незначно впливають на виконання заданих функцій, але деякі відхилення, наприклад, несправності (дефекти), а також відмови елементів і вузлів системи в процесі експлуатації, призводять до неправильного функціонування ІС, внаслідок чого вона не може використовуватися за призначенням. При цьому одним із найбільш поширених способів виявлення таких відхилень є тестове та функціональне діагностування. Так, згідно з [5–7] тестове діагностування полягає у подачі на входи ІС серії наборів тестових вхідних сигналів та спостереження за відповідною послідовністю вихідних сигналів. Такі дослідження, на рівні логічних елементів, дозволяють виявити помилки проектування та виготовлення, а також визначити несправності елементів ІС. При цьому під тестовим діагностуванням розуміється логічне тестування, при якому всі сигнали набувають два значення – 0 або 1.

Проте слід зазначити, що одним з недоліків логічного тестування є те, що воно не враховує фізичної природи вхідних та вихідних сигналів, перехідних процесів та затримок в елементах пристрою, їх

навантажувальних можливостях, а також впливу параметрів зовнішнього середовища на роботу ІС.

Функціональне тестування, в свою чергу, застосовується для перевірки правильності реалізації функцій, якими наділяється цифровий пристрій, не враховуючи при цьому інформацію про логічну структуру схеми. Так, основними видами функціонального тестування є випадкове та вичерпне тестування. При випадковому тестуванні тестові набори являють собою n -мірні двійкові вектори (псевдовипадкові послідовності), що генеруються датчиком випадкових чисел. При цьому дані послідовності мають усі необхідні для тестування властивості випадкових чисел, а також перевагу, що псевдовипадкова послідовність завжди може бути повторена, починаючи з будь-якого набору. Однак поряд з перевагами випадкового тестування існують також недоліки, які пов'язані зі складністю оцінки результатів тестування, іншими словами, із встановленням зв'язку довжини випадкового тесту з ймовірністю виявлення помилок.

У свою чергу, вичерпне тестування є межею випадкового тестування і являє собою таке тестування, при якому в тесті вичерпним чином перебираються всі 2^n комбінації тестових сигналів. При цьому успішне проходження вичерпного тесту рівнозначне доказу відповідності схеми вихідній булевої функції. Крім цього, вичерпний тест дуже легко генерувати, однак він має, принаймні, два недоліки.

Перший недолік у тому, що існують дефекти, які виводять схему з класу комбінаційних схем. Типовим дефектом такого роду є перемичка, яка не виявляється вичерпним тестуванням.

Другий недолік вичерпного тестування – велика довжина тесту. Так, для комбінаційної схеми розмірністю 10×2 необхідно подати на її входи 2^{10} тестових сигналів. Тому для більшості схем вичерпне тестування є складним.

Крім цього слід зазначити, що методи тестового та функціонального діагностування не дають повної діагностичної інформації про технічний стан ІС. А враховуючи той факт, що ІС з програмованою структурою створюють велику конкуренцію стандартним ІС, то питання, які пов'язані з необхідністю застосування існуючих та розробки нових методів діагностування ПЛІС є актуальними.

Аналіз останніх досліджень і публікацій. На сьогоднішній день питанням тестування цифрових пристроїв (інтегральних схем) присвячено велику кількість досліджень. Так, у роботі [8] розглядається проблема низької ефективності відомих детермінованих методів побудови тестів для цифрових схем, головною причиною є некерований перебір всіх комбінацій сигналів, які забезпечують прояв несправності на виході елемента з дефектом. В якості переваги в роботі запропонований метод фокусування пошуку для побудови тестів.

У роботі [9] представлена кубічна технологія аналізу цифрових схем для генерації тестів та оцінки їх якості. Запропоновані кубічний метод обробки несправностей моделі примітиву та метод генерації тестів для константових несправностей, що викори-

стовують кубічні покриття списків несправностей для побудови шляху активізації.

В [10] проведено аналіз двох конкуруючих методів контролю та діагностування цифрових схем: спектральний метод та метод сигнатурного аналізу. Доведено, що спектральний метод є найефективнішим. Це пов'язано з тим, що спектрограми, на відміну від сигнатур, найбільш пристосовані до побудови словника несправностей. Крім цього, використовуючи спектральний метод, можна варіювати в широких межах інформативності діагностичних ознак залежно від необхідної глибини діагнозу.

У [11] запропоновано для тестування цифрових пристроїв застосовувати генетичні алгоритми, що становлять адаптивний пошуковий метод, заснований на селекції найкращих елементів популяції, тобто логічних елементів на інтегральній схемі. При цьому на ефективність даного алгоритму істотно впливає правильний вибір способу кодування особин, і навіть правильний вибір параметрів. У [12] розглянуто відносно новий безконтактний, індукційний метод діагностування радіоелементів технічно складних цифрових радіоелектронних блоків, який дозволяє визначати не тільки технічний стан контрольованого блоку, а й знайти радіоелектронний компонент, який спровокував несправність чи відмову блоку.

Проте слід зазначити, що в розглянутих роботах тестування цифрових схем, а також оцінка технічного стану цифрових пристроїв (справний, працездатний і правильно функціонуючий) розглядається з позиції того, що дані цифрові пристрої реалізовані на стандартних ІС, на відміну від ПЛІС, де структура носить реконфігураційний характер. При цьому необхідно відзначити, що тестування цифрових інтегральних схем з структурою, яка програмується, не залежить від функціонування кінцевого, створеного користувачем цифрового пристрою [13].

Враховуючи вищезазначене, **метою статті** є обґрунтування підходу до діагностування мікропроцесорних систем у базисі ПЛІС, який є подальшим розвитком методу самодіагностики, застосування якого дозволить підвищити живучість і як наслідок надійність складних технічних систем, що функціонують в умовах несприятливого впливу, у тому числі і кібератак.

Проблема тестування цифрових пристроїв у базисі ПЛІС

Програмовані логічні інтегральні схеми – перспективна архітектура ВІС/НВІС, яка на сьогоднішній день знаходить все більш широке застосування у якості типових елементів функціонально складних цифрових систем високого ступеня інтеграції.

Однією з переваг ПЛІС є те, що логіка роботи не визначається при виготовленні, а задається шляхом програмування, що дає можливість спростити процес синтезу складних цифрових схем. Другою перевагою стало те, що завдяки тій же регулярності структури відбувається більш ефективно використовувати площу кристала, підвищуючи тим самим ступінь інтеграції. І, нарешті, третьою перевагою є те, що володіючи властивістю реконфігації, ПЛІС дозволяють проектувати

високонадійні, відмовостійкі мікропроцесорні системи, тим самим підвищуючи живучість, і як наслідок надійність складних технічних систем. Однак, незважаючи на низку переваг ПЛІС, таких, як регулярність структури та концептуальна простота процесу синтезу, їх тестування є трудомістким та складним процесом. Зростаюче застосування ВІС/НВІС в апаратурі, що виконують життєво важливі функції, такі, наприклад, як управління атомними електростанціями, навігація, системи управління технологічними процесами тощо, вимагає дуже глибокого тестування.

З іншого боку, навіть при використанні ефективних алгоритмів і потужних процесорних вимірювальних засобів для схем практичної складності не завжди вдається досягти 100% покриття несправностей. Так, згідно [13] одним із шляхів вирішення цієї проблеми є зміна та доповнення канонічної структури ПЛІС з таким розрахунком, щоб забезпечити потрібний рівень тестування. При цьому переслідуються такі цілі: спрощення генерації тестів; забезпечення генерації тестових послідовностей простими апаратними засобами, у випадку створення самотестованих ВІС/НВІС; підвищення покриваючої здатності тестових послідовностей; полегшення аналізу результату тестування. При цьому подальша реалізація цілей можлива, принаймні, трьома підходами [13].

Перший підхід – при проектуванні ПЛІС враховується необхідність її тестування, до схеми вносяться корективи, що полегшують цей процес. Побудовані на основі цього підходу ПЛІС прийнято називати зручнотестованими. При цьому всі методи побудови зручнотестованої ПЛІС можна розбити на два типи: які не змінюють канонічну дворівневу структуру ПЛІС; які використовують введення додаткових функціональних елементів, призначених для покращення тестування.

Другий підхід, заснований на запровадженні структурної надмірності, яка робить ПЛІС стійкою до відмови в тому сенсі, що несправності в ній або маскуються, або виявляються в момент виникнення.

Третій підхід передбачає обов'язкове додавання до структури ВІС/НВІС функціональних елементів, які здійснюють автономне тестування ПЛІС, в результаті виходить ВІС/НВІС із вбудованою діагностикою.

На сьогоднішній день ці підходи досить повно описані та реалізовані для ПЛІС першого покоління (*PAL – Programmable Array Logic, GAL – Generic Array Logic*) зі структурою комбінаційного типу. Дані типи структур є сукупністю двох матриць І та АБО, кожна з яких містить систему взаємно ортогональних шин, у точках перетину яких встановлюються напівпровідникові елементи зв'язку [13]. Однак з появою ПЛІС зі структурами, основу яких складають програмовані логічні блоки – макрокомірки, з'єднані із зовнішніми виводами та внутрішніми шинами (*CPLD – структура*) та *CLB* блоки (*Configurable Logic Block*) з таблицею відповідності *LUT (Look Up Table)*, за допомогою якої задаються булеві функції від кількох аргументів (*FPGA – структура*) – виникають складності тестування даного типу ПЛІС. Крім цього, враховуючи той факт, що на базі даних структур створено

абсолютно новий напрямок розвитку мікроелектроніки – універсальні мікропроцесорні системи на кристали (*System-on-Chip – SoC, SoPC – System-on-a-Programmable-Chip, MPSoC – Multiprocessor System-on-Chip*), то виникають складності тестування не просто цифрових пристроїв, а мікропроцесорних систем у базисі ПЛІС. При цьому технологія *System-on-Chip* складається з двох самостійних функціональних частин *FPGA (Field-Programmable Gate Array –* програмована вентилярна матриця) та *HPS (Hard Processor System –* жорстка процесорна система), які з'єднані між собою інтерфейсом обміну даними. Розглянемо найефективніші методи тестування ПЛІС зі структурами *CPLD* та *FPGA* [14]:

Метод використання схем самодіагностики (BIST – Built-In-Self-Test). Цей метод полягає у реалізації на рівні прошивки ПЛІС спеціалізованої схеми (пристрою), що дозволяє проводити внутрішнє самотестування. На рис. 1 представлена структура самотестуючого пристрою, що складається з генератора тестових сигналів (*TPG*), аналізатора відгуку (*ORA*) та логіки управління. Суть даного методу полягає в подачі на вхід тестованого пристрою псевдовипадкових тестових наборів та подальше порівнювання результуючого значення з еталонним значенням у сигнатурному аналізаторі. Реалізація алгоритму на основі даного методу передбачає використання певного відсотка внутрішньої логіки ПЛІС, а при проведенні функціонального контролю необхідно перевіряти весь обсяг внутрішньої логіки. Тому цей метод найбільш зручно застосовувати для тестування внутрішніх блоків ПЛІС.

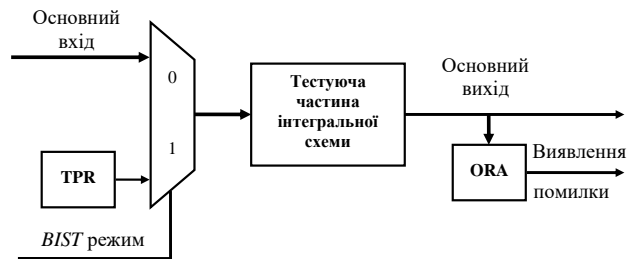


Рис. 1. Архітектура стандартної схеми самодіагностики (Fig. 1. The architecture of self-diagnosis standard scheme)

Метод повторювальних тестових множин (ILA – Iterative Logic Arrays). Його суть полягає в тому, що конфігураційні логічні блоки (*CLB*), з яких складається ПЛІС, конфігуруються таким чином, що вихідні сигнали попереднього блоку *CLB* є вхідними сигналами наступного блоку, утворюючи тим самим тестовий ланцюжок. Цей метод є оптимальним для функціонального контролю внутрішньої логіки ПЛІС. Однією з переваг методу є висока спостережувальність, що дозволяє визначити місце виникнення відхилення. Основним недоліком методу є необхідність використання спеціалізованого обладнання для проведення тестування, а також низька, порівняно з *BIST*, швидкість тестування.

Метод зчитування вихідних відгуків у вигляді вбудованих інтерфейсів. Цей метод використовується для проведення функціонального контролю вбу-

дованих блоків ПЛІС. Перевагами даного методу є висока швидкість та можливість проведення функціонального контролю без використання дорогого обладнання. Основним недоліком є те, що даний метод застосовується тільки для ПЛІС нового покоління за наявності спеціалізованого апаратного інтерфейсу *JTAG (Joint Test Action Group)* на базі стандарту *IEEE 1149.1*. При цьому тестування всієї ПЛІС є досить трудомістким і полягає в поєднанні даного методу з методом вбудованих схем самодіагностики у вигляді конфігураційної прошивки.

Таким чином, можна зробити висновок, що найефективнішим методом тестування є метод, заснований на техніці перевірки *BIST – Built-In-Self-Test*, тобто вбудованій самоперевірці. Виходячи з цього, подальший розгляд мікропроцесорних систем у базисі ПЛІС, з точки зору діагностування, спиратиметься саме на метод використання схем самодіагностики.

Підходи до діагностування мікропроцесорних систем у базисі ПЛІС

Мікропроцесорна система, як об'єкт діагностики, є складною обчислювальною структурою з шинною організацією. Вона складається з чотирьох основних груп ВІС/НВІС – мікропроцесора, пам'яті, контролерів введення-виводу та контролерів зв'язку з об'єктами. Кожна з цих функціональних підсистем, у свою чергу, є досить складною з точки зору діагностування. Тому при організації тестового та функціонального діагностування мікропроцесорних систем використовується декомпозиційний підхід, при якому як об'єкт діагностування виступають окремі функціональні пристрої: арифметико-логічний пристрій, процесор, оперативно-запам'ятовувачий пристрій, пристрій введення-виводу. На рис. 2, 3 представлені узагальнені схеми процесів діагностування мікропроцесорних систем, що складаються з об'єкта діагностування, засобів діагностування та системи усунення несправностей, тобто системи оперативного відновлення правильного функціонування.

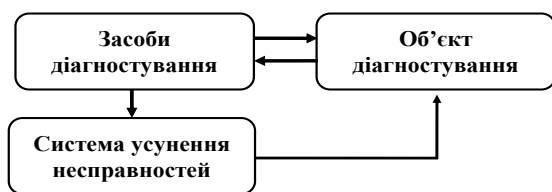


Рис. 2. Схема організації тестового діагностування (Fig. 2. The scheme of the organization of test diagnosing)

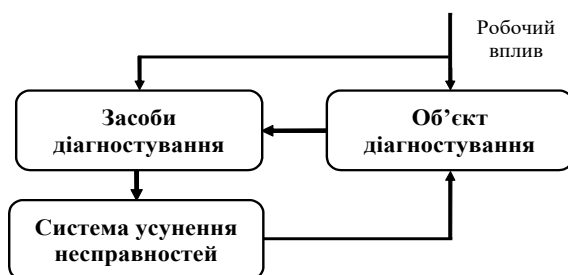


Рис. 3. Схема організації функціонального діагностування (Fig. 3. Scheme of organization of functional diagnostics)

Схема організації тестового діагностування (рис. 2) відрізняється від схеми організації функціонального діагностування (рис. 3) можливістю подачі на об'єкт діагностування спеціально організованих (тестових) впливів, у той час як у процесі функціонального діагностування використовується тільки робочий вплив. При цьому застосування даних схем у мікропроцесорних системах вирішують завдання оперативного виявлення помилок та пошуку місця несправності. Крім цього, при діагностуванні мікропроцесорних систем необхідно враховувати труднощі, які пов'язані з високим ступенем інтеграції ВІС/НВІС, розгалуженими зв'язками між елементами системи, а також відсутністю повної інформації про структуру системи. Окремо слід виділити труднощі діагностування мікропроцесорних систем у базисі ПЛІС, які пов'язані, як було зазначено вище, з реконфігураційним характером елементної бази систем, які розглядаються.

Саме з цих причин згідно [15, 16] існуючі методи (конструктивні та абстрактні) побудови діагностичних тестів для мікропроцесорних систем на стандартних ІС (метод таблиці функцій несправностей, метод існуючих шляхів, *D*-алгоритм, метод еквівалентної нормальної форми, булеве диференціювання, моделі регістрових передач, мережева модель Петрі тощо) не завжди можуть бути ефективними для діагностування мікропроцесорних систем у базисі ПЛІС. Крім цього, згідно [17] використання зовнішніх засобів контролю та діагностування в мікропроцесорних системах є, як правило, невиправданим. Це пов'язано з тим, що втручання у роботу мікропроцесорної системи призводить до її ускладнення і зниження, як продуктивності, так і надійності системи.

Виходячи зі сказаного, для визначення технічного стану мікропроцесорної системи в базисі ПЛІС, а також діагностування несправностей у ній, є доцільним використання принципів взаємодії процесорів між собою шляхом введення до складу багатопроцесорної системи сервісного процесора [18]. Основною функцією такого процесора є діагностика багатопроцесорної системи та оперативне автоматичне відновлення шляхом реконфігурації системи, тобто перепрограмування ПЛІС.

Таким чином, подальше підвищення живучості і як наслідок надійності, а також розширення функціональних можливостей багатопроцесорної системи в базисі ПЛІС пов'язане з використанням апаратної надмірності, шляхом введення до складу системи спеціальних апаратних засобів, що реалізують функцію спостереження, тобто використання принципу взаємодії.

Розглянемо найбільш поширені моделі взаємодії (*P*-, *B*-, *R*-, *K*- моделі), що описують діагностування та контроль у багатомашинних системах [17]. Так, *P* – модель (*Preparato-Metze-Chien*) заснована на представленні процесу діагностування взаємодією пар модулів, наслідком якого є одержання множини результатів діагностування. Діагностична модель представляється у вигляді діагностичного графа, де модулі – це вершини, а міжмодульні зв'язки – дуги [19].

У моделі модуль \mathcal{G}_i контролює модуль \mathcal{G}_j , а загальна кількість модулів – n . Перевірка за допомогою тестової послідовності h_{ij} модулем \mathcal{G}_i технічного стану модуля \mathcal{G}_j називається елементарної перевіркою, а її результат позначається r_{ij} . Сукупність тестів h_{ij} утворює множину J , а сукупність елементарних перевірок r_{ij} – множину R , $R = \{r_{ij}\}$, яка називається синдромом. Уся множина J перевірок здійснюється за один цикл діагностування. Результат елементарної перевірки позначається «0», якщо тест «пройшов», і «1» – в іншому випадку.

B – модель (Barsi-Grandoni-Maestrini) відрізняється від P – моделі лише тим, що недійсним вважається отримання результату контролю $r_{ij} = 0$ за наявності відмов одночасно в елементах \mathcal{G}_i і \mathcal{G}_j . При цьому застосування B – моделі призводить до меншої кількості допустимих станів працездатності системи [20].

R – модель (Russel-Kime) передбачає опис перевірки одного елемента декількома іншими апаратно або за допомогою програмних процедур [21]. У системах, що описуються такими моделями, n елементам ставиться у відповідність множина F_0 допустимих поодиноких несправностей $f_i : F_0 = (f_1, \dots, f_n)$, для яких задається множина J тестів $h_j, j = 1, \dots, p, J = \{h_1, \dots, h_p\}$. При цьому передбачається, що кожен тест є повним для однієї несправності з множини F_0 .

І нарешті K – модель (Makam-Avizienis) виділяє такі види несправностей [22]: $A = a_1, \dots, a_{|A|}$ – множина атомарних несправностей; $M = m_1, \dots, m_{|M|}$ – множина макронесправностей; $P = p_1, \dots, p_{|P|}$ – множина групових дефектів, де $|A|, |M|, |P|$ – потужність множин A, M, P відповідно.

Множина атомарних несправностей A описує відмови на найнижчому логічному рівні системи. Вони можуть бути описані переліком функціональних елементів чи постійними несправностями. Групові дефекти множини M являють собою фізично або логічно замінні частини системи, які визначають глибину пошуку дефекту в системі. Макронесправності є наслідком виникнення підмножини атомарних несправностей з множини несправностей A .

Між множинами несправностей існують певні відношення. Зокрема, введена функція f_{ap} з A в P і відображення зв'язку r_{am} між A і M , а також r_{pm} між M і P . Функція f_{ap} із A в P , що позначається $f_{ap} : A \rightarrow P$, призначає кожній атомарній несправності один і тільки один груповий дефект. Вважається, що груповий дефект виник, якщо є атомарна несправність, тобто $\forall a_i \in A$, $\exists p_i \in P : f_{ap}(a_i) = p_j$.

Відображення r_{am} описує зв'язок між елементами множини A і M , при якій макронесправність m_j виникає щоразу, як тільки з'являється атомарна несправність a_i . У багатьох випадках відображення r_{am} вдається представити функцією f_{am} . Зв'язок між елементами множини M і P полягає в тому, що якщо m_j – дефект з групового дефекту, то принаймні один з дефектів є наслідком макронесправності.

Елементи множин A, M, P є поодинокими несправностями системи. Для завдання наборів несправностей для кожної з цих множин визначають множини A', M', P' підмножин всіх елементів відповідно з A, M, P , тобто. $M' \subseteq C(M)$; $A' \subseteq C(A)$; $P' \subseteq C(P)$. Для множин A', M', P' визначають функції f_{AP} і f_{AM} таким чином:

$$f_{AP} : A' \rightarrow C(P), f_{AM} : A' \rightarrow C(M), \forall A^i \in A',$$

$$f_{AP}(A^i) = P^j = \{p_k | p_k = f_{ap}(a_i), a_i \in A^i\};$$

$$f_{AM}(A^i) = M^j = \{m_k | a_k r_{am} m_k, a_i \in A^i\},$$

діагностування в системі проводиться за допомогою атомарних тестів u_i множин $U = \{u_1, \dots, u_{|U|}\}$ і макротестів h_j множин $T = \{h_1, \dots, h_j, \dots, h_{|T|}\}$, а також вхідних послідовностей множин $J = \{J_1, \dots, J_{|J|}\}$.

Атомарний тест полягає в спостереженні одного виходу в момент часу при подачі на вхід схеми вхідної послідовності. Макротест являє собою множину атомарних тестів. Таким чином, порівнюючи моделі між собою, можна відзначити, що P - і B - моделі орієнтовані переважно на діагностування з використання результатів взаємодії елементів між собою. В R - і K - моделях при пошуку несправностей (дефектів) більшою мірою враховується функціональна структура об'єкта діагностування, діагностичні ознаки та можливості програмних засобів. Крім цього, реалізація даних моделей взаємодії дає можливість за наявності діагностичної інформації про технічний стан мікропроцесорної системи приймати сервісним процесором рішення про її реконфігурацію, тобто перерозподіл задач (функцій) шляхом перепрограмування ПЛІС.

Так, на сьогоднішній день згідно [23] існує два підходи до організації перерозподілу задач у системі при відмовах модулів. Перший підхід – статичний, передбачає, що відома підмножина $S = \{s_v\}$ станів системи, для якої необхідно здійснювати перерозподіл завдань, і потрібно знайти множину $\Gamma = \{G_v\}$ розподілів завдань таку, що кожен розподіл G_v , який відповідає стану s_v , задовольняє вимогам, які висуваються до показників живучості. При цьому оптимальні розподіли G_v для всіх станів $s_v \in S$ знаходяться при проектуванні системи, і кожен модуль при побудові системи забезпечується ресурсами, необхідними для виконання завдань, як у стані s_0 , так і в кожному стані $s_v \in S$ відповідно до

знайдених розподілів G_V . При переході системи в стан $s_V \in S$ та виявленні факту відмови у всіх працездатних модулях системи починається виконання завдань, що відповідають розподілу G_V .

Другий підхід, який називається динамічним, заснований на тому, що завдання знаходження оптимального розподілу G_V для стану s_V вирішується щоразу (за допомогою спеціального вирішального блоку, тобто сервісного процесора) під час переходу системи в стан s_V і результат її вирішення може залежати від попередніх станів системи. У загальному випадку, як при статичному, так і при динамічному підходах отримання оптимального розподілу може вимагати перерозподілу завдань як модулів, які відмовили, так і працездатних модулів.

Резюмуючи сказане, можна констатувати, що реалізація принципу взаємодії процесорів шляхом введення до складу багатопроекторної системи в базисі ПЛІС сервісного процесора дає можливість оцінювати не тільки технічний стан системи, а також приймати рішення щодо автоматичного відновлення системи шляхом перепрограмування ПЛІС.

Висновки

Розглянуто проблему діагностування мікропроцесорних систем у базисі ПЛІС, як основу побудови складних технічних систем, які функціонують в умовах несприятливого впливу, у тому числі і кібератак. Проведено аналіз існуючих методів тестування інтегральних схем із програмованою структурою, відзначено їх переваги та недоліки. Запропоновано в якості діагностичного пристрою використовувати сервісний процесор як невід'ємний елемент багатопроекторної системи. Такий процесор дозволить не лише збирати діагностичну інформацію, а й приймати рішення про реконфігурацію системи з метою оперативного автоматичного відновлення її функціонування шляхом перепрограмування ПЛІС.

Загалом, запропонований підхід до діагностування мікропроцесорних систем у базисі ПЛІС є подальшим розвитком методу самодіагностики. Крім цього якщо наділити складну технічну систему властивістю адаптації до умов функціонування, що змінюються, то таку систему можна розглядати як систему з елементами штучного інтелекту.

СПИСОК ЛІТЕРАТУРИ (REFERENCES)

1. R. Joost and R. Salomon, "Advantages of FPGA-based multiprocessor systems in industrial applications," 31st Annual Conference of IEEE Industrial Electronics Society, 2005. IECON 2005., 2005, pp. 6 pp., doi: 10.1109/IECON.2005.1568946.
2. Mousavi, M., De, S., Pourshaghagh, H. R., & Corporaal, H. (2019). Fault Tolerant FPGAs: where to spend the effort? In N. Konofaos, & P. Kitsos (Eds.), Proceedings – Euromicro Conference on Digital System Design, DSD 2019 (pp. 651-654). [8875164] Institute of Electrical and Electronics Engineers. doi.org/10.1109/DSD.2019.00103
3. Самохвалов Ю.Я. Забезпечення кібербезпеки АСУ ТП шляхом застосування ПЛІС технології / Ю.Я. Самохвалов, С.В. Толюпа, С.С. Штаненко. – Науковий журнал Київського національного університету імені Тараса Шевченка. Безпека інформаційних систем і технологій № 1, 2021 стр. 44 – 52. ISSN 2707-1758.
4. Самохвалов Ю.Я. & Штаненко С.С. Технологія System-on-Chip як основа підвищення живучості складних технічних систем. *Сучасна спеціальна техніка*. ДНДІ МВС України. Київ: №3(66), 2021. С. 31 – 43.
5. Фізичні основи теорії надійності: підручник / М. К. Жердев, С. В. Ленков, Б. П. Креденцер та ін.; за ред. М. К. Жердева. – К.: Видавничо-поліграфічний центр «Київський університет», 2008. – 215 с.
6. Діагностування аналогових і цифрових пристроїв радіоелектронної техніки / [В.В. Вишнівський, М.К. Жердев, С.В. Ленков, В.А. Проценко]; под ред. М.К. Жердева, С.В. Ленкова – М.: Сов. радио, 2009. – 224 с.
7. Гуляев В.А. Техническая диагностика управляющих систем. – Киев: Наук. думка, 1983. – 208 с.
8. Куліков В.М. Порівняльний аналіз методів побудови тестів для цифрових пристроїв. Збірник наукових праць ІСЗІ НТУУ «КПІ», Information Technology and Security, № 2(2), 2012. С. 34 – 44.
9. Хаханов В.И., Сысенко И.Ю., Хак Х.М. Джахирул, Масуд М.Д. Мехеди. Кубическое моделирование неисправностей цифровых проектов на основе FPGA, CPLD. *Радиоэлектроника, информатика, управление*. 2001. № 1. С. 123-129.
10. Ручко В.В. Контроль цифровых схем с помощью спектрограмм. / В.В. Ручко, Ю.Г. Савченко, А.В. Хмелевая // Управляющие системы и машины. – К.: Наукова думка. – №3(71), 1984. – С. 28 – 31.
11. Гордилов А.Ю. Постановка задачи диагностирования программируемых логических интегральных схем и анализ возможности применения генетических алгоритмов для ее решения. Научный журнал «Фундаментальные исследования». – №10, 2013. – С. 598 – 562. ISSN 1812-7339
12. Кузавков В. В., Гайдур Г. І., Серих С. О., Редзюк С. В. Безконтактний індукційний метод визначення технічного стану цифрового блока: розрахунок потужності випромінювання провідника. *Зв'язок*. 2016. №1. С. 32–39.
13. Бутаков Е.А. Диагностика программируемых логических матриц / Е.А. Бутаков, М. Б. Волынский, В.Г. Новоселов. – М.: Радио и связь, 1991. – 157 с.
14. Огурцов А.А. Проведение функционального контроля программируемых логических интегральных схем // Вестник Самарского университета. Аэрокосмическая техника, технологии и машиностроение. 2017. Т. 16, № 4. С. 137–146.
15. Слесарев М. В. Особенности разработки и применения аппаратуры тестирования и разбраковки многоядерных серверных микропроцессоров / М. В. Слесарев, С. В. Юрлин // Вопросы радиоэлектроники. 2017. № 3. С. 62 – 68.
16. Микони С. В. Общие диагностические базы знаний вычислительных систем, СПб.: СПИИРАН. – 1992. – 234 с.
17. Коваленко А. Е., Гула В. В. Отказоустойчивые микропроцессорные системы. К.: Техніка, 1986. 150 с.
18. Погребинский С. Б. Проектирование и надежность многопроцессорных ЭВМ / С.Б. Погребинский, В.П. Стрельников. – М.: Радио и связь, 1988. – 168 с.
19. Preparato F. P. On the Connection Assignment Problem of Diagnosable Systems / F. P. Preparato, G. Metze, R. J. Chien // IEEE Trans. 1967. № 6. P. 848–854.
20. Barsi T., Crandoni T., Maestrini P. A. Thecry of Diagnosability of Digital Systems / IEEE Trans., 1976. № 6. P. 585 – 593.
21. Russel J., Kime C. Sistem Fault Diagnosability Without Repair // IEEE Trans. 1975. vol. C-24. №11. P. 1078 – 1089.
22. Kime C. K. An Abstract Model for Digital System Diagnosis // IEEE Trans. vol. C-28. № 8. P. 754 – 767.

23. Лазарев В.Г. Построение программируемых управляющих устройств / В.Г. Лазарев, Е. И. Пийль, Е. Н. Турута. – М.: Энергоатомиздат, 1984. – 192 с.

Received (Надійшла) 28.11.2021

Accepted for publication (Прийнята до друку) 26.01.2022

ABOUT THE AUTHORS / ВІДОМОСТІ ПРО АВТОРІВ

Штаненко Сергій Станіславович – кандидат технічних наук, доцент, докторант науково-організаційного відділу, Військового інституту телекомунікацій та інформатизації імені Героїв Крут, Київ, Україна;

Serhii Shtanenko – Candidate of Technical Sciences, Associate Professor, Doctoral Student of the Scientific and Organizational Department, Heroes of Kruty Military Institute of Telecommunications and Informatization, Kyiv, Ukraine; e-mail: sh_sergei@ukr.net; ORCID ID: <https://orcid.org/0000-0001-9776-4653>.

Самохвалов Юрій Якович – доктор технічних наук, професор, професор кафедри інтелектуальних технологій Київського національного університету імені Тараса Шевченка, Київ, Україна;

Yurii Samokhvalov – Doctor of Technical Sciences, Professor, Professor of the Department of Intellectual Technologies, Taras Shevchenko National University of Kyiv, Kyiv, Ukraine; e-mail: yu1953@ukr.net; ORCID ID: <https://orcid.org/0000-0001-5123-1288>.

Іохов Олександр Юрійович – доктор технічних наук, доцент, начальник кафедри військового зв'язку та інформатизації, Національна академія Національної гвардії України, Харків, Україна;

Oleksander Iokhov – Doctor of Technical Sciences, Associate Professor, Head of the Department of Military Communication and Informatization, National Academy of the National Guard of Ukraine, Kharkiv, Ukraine; e-mail: iokhov@ukr.net; ORCID ID: <https://orcid.org/0000-0002-1718-0138>.

Малюк Віктор Григорович – кандидат технічних наук, доцент, професор кафедри військового зв'язку та інформатизації, Національна академія Національної гвардії України, Харків, Україна;

Victor Maliuk – Candidate of Technical Sciences, Associate Professor, Professor of the Department of Military Communication and Informatization, National Academy of the National Guard of Ukraine, Kharkiv, Ukraine; e-mail: vgmaluk@gmail.com; ORCID ID: <https://orcid.org/0000-0001-6510-3025>.

Микропроцессорные системы на программированных логических интегральных схемах как объект диагностики

С. С. Штаненко, Ю. Я. Самохвалов, А. Ю. Иохов, В. Г. Малюк

Аннотация. Предметом исследования в статье являются методы тестирования цифровых устройств, реализуемых на программируемых логических интегральных схемах (ПЛИС). **Целью работы** является обоснование подхода к диагностированию микропроцессорных систем в базе ПЛИС, который является дальнейшим развитием метода самодиагностики. **В статье решаются следующие задачи:** проведен анализ существующих методов тестирования цифровых устройств, раскрыты их преимущества и недостатки; проанализированы существующие подходы к диагностированию микропроцессорных систем, основу которых составляют интегральные схемы с программируемой структурой; предложен подход к диагностированию микропроцессорных систем в базе ПЛИС, в основу которого положен принцип взаимодействия процессоров между собой путем ввода в многопроцессорную систему сервисного процессора. **Получены следующие результаты:** доказано, что реализация предлагаемых моделей взаимодействия процессоров между собой в многопроцессорной системе наделит сервисный процессор диагностической информацией о техническом состоянии системы. Отмечено, что наличие диагностической информации о техническом состоянии микропроцессорной системы является основанием для принятия решения на восстановление системы путем перепрограммирования ПЛИС. **Выводы:** рассмотрена проблема диагностирования микропроцессорных систем в базе ПЛИС. Проведен анализ существующих методов тестирования интегральных схем с программируемой структурой, отмечены их преимущества и недостатки. Предложено в качестве диагностического устройства использовать встроенный сервисный процессор, основной функцией которого является сбор диагностической информации и принятие решения о реконфигурации микропроцессорной системы с целью оперативного автоматического восстановления ее функционирования.

Ключевые слова: микропроцессорная система; программируемая логическая интегральная схема; диагностика цифровых устройств.

Microprocessor systems based on programmable logic devices as an object of diagnostics

Serhii Shtanenko, Yurii Samokhvalov, Oleksander Iokhov, Victor Maliuk

Abstract. The subject of research in the article is the methods of testing digital devices implemented on programmable logic devices (PLDs). **The aim of the research** is to substantiate the approach to diagnosing microprocessor systems based on PLD, which is a further development of the self-diagnostics method. **The following tasks are solved in the article:** the analysis of existing methods for testing digital devices is conducted, their advantages and disadvantages are disclosed; analyzed the existing approaches to the diagnosis of microprocessor systems, which are based on integrated circuits with a programmable structure; an approach to diagnosing microprocessor systems based on PLDs is proposed, which is based on the principle of interaction between processors by introducing a service processor into a multiprocessor system. **The following results have been obtained:** it has been proved that the implementation of the proposed models of interaction between processors in a multiprocessor system will provide the service processor with diagnostic information about the technical state of the system. It is noted that the availability of diagnostic information about the technical condition of the microprocessor system is the basis for making a decision to restore the system by reprogramming the PLD. **Conclusions:** the problem of diagnosing microprocessor systems in the PLD basis is considered. The analysis of existing methods for testing integrated circuits with a programmable structure has been conducted, their advantages and disadvantages have been noted. It is proposed to use the built-in service processor as a diagnostic device, the main function of which is to collect diagnostic information and make a decision on the reconfiguration of the microprocessor system in order to quickly restore its functioning automatically.

Keywords: microprocessor system; programmable logic devices; diagnostics of digital devices.