

УДК 519.718.2

канд. техн. наук Штаненко С. С. ORCID: 0000-0001-9776-4653 (ВІТІ ім. Героїв Крут)
д-р техн. наук Самохвалов Ю. Я. ORCID: 0000-0001-5123-1288 (КНУ ім. Тараса Шевченка)
д-р техн. наук Толюпа С. В. ORCID: 0000-0002-1919-9174 (КНУ ім. Тараса Шевченка)

МЕТОДИЧНИЙ ПІДХІД ДО ВІДНОВЛЕННЯ ПРАВИЛЬНОГО ФУНКЦІОНУВАННЯ ВБУДОВАНИХ СИСТЕМ НА РІВНІ ПРОГРАМОВАНОЇ ЕЛЕМЕНТНОЇ БАЗИ

У статті запропоновано методичний підхід до відновлення правильного функціонування спеціалізованих мікропроцесорних систем управління на рівні програмованої елементної бази. Цей підхід включає два етапи.

На першому етапі вирішується основна задача технічної діагностики, а саме розпізнавання стану спеціалізованої мікропроцесорної системи, яка включає в себе: оцінку технічного стану, прогнозування, виявлення та локалізацію несправностей. Дана задача вирішується шляхом виявлення помилок в роботі спеціалізованої мікропроцесорної системи управління, які спричинені відмовою та збоєм цифрових пристроїв, помилками в програмному забезпеченні, або іншими причинами, застосовуючи існуючі методи контролю, а також локалізацією несправностей спеціалізованої мікропроцесорної системи управління, застосовуючи методи тестового та функціонального діагностування цифрових пристроїв, які є складовою системи, яка розглядається.

На другому етапі здійснюється відновлення правильного функціонування спеціалізованої мікропроцесорної системи управління шляхом реконфігурації її внутрішньої структури на рівні логічних елементів. При цьому в основу реконфігурації внутрішньої структури покладене положення прескриптивної теорії, яка розглядає питання цілеспрямованого управління об'єктами різної природи, які перебувають у стані «конфлікту» з іншими об'єктами.

Реалізація запропонованого підходу в подальшому може бути основою для проектування активних відмовостійких систем, які будуть здатні протидіяти відмовам та збоєм апаратного та програмного характеру в наслідок внутрішніх або зовнішніх несприятливих впливів.

Ключові слова: прескриптивна теорія, реконфігурація, спеціалізована мікропроцесорна система управління, вбудована система, контроль і діагностування цифрових пристроїв, відновлення правильного функціонування.

S. Shtanenko, Y. Samokhvalov, S. Toliupa Methodological approach to recovery correct functioning of embedded systems at the level of programmable element base.

The article proposes a methodical approach to restoring the proper functioning of specialized microprocessor control systems at the level of the programmable element base. This approach includes two stages.

At the first stage, the main task of technical diagnostics is solved, namely the recognition of the state of a specialized microprocessor system, which includes: assessment of the technical state, forecasting, detection and localization of malfunctions. This task is solved by detecting errors in the operation of a specialized microprocessor control system, which are caused by the failure and failure of digital devices, errors in software, or other reasons, using existing control methods, as well as localization of malfunctions of a specialized microprocessor control system, using methods of test and functional diagnostics digital devices that are part of the system under consideration.

At the second stage, the correct functioning of the specialized microprocessor control system is restored by reconfiguring its internal structure at the level of logical elements. At the same time, the reconfiguration of the internal structure is based on the position of the prescriptive theory, which considers the issue of purposeful management of objects of various nature, which are in a state of "conflict" with other objects.

The implementation of the proposed approach in the future can be the basis for the design of active fault-tolerant systems that will be able to counteract failures a hardware and software nature as a result of internal or external adverse actions.

Keywords: prescriptive theory, reconfiguration, specialized microprocessor control system, embedded system, control and diagnostics of digital devices, restoration of proper functioning.

Постановка завдання. Вбудовані системи (Embedded Systems) являють собою спеціалізовані мікропроцесорні системи управління, концепція розробки яких ґрунтується на тому, що такі системи взаємодіють з об'єктом управління або контролю, будучи вбудованими безпосередньо у пристрій, яким вони управляють [1; 2].

На сьогоднішній день вбудовані системи широко використовуються в різних галузях діяльності, таких як: машинобудування та верстатобудування, авіація, автомобілебудування, атомна енергетика, банківська сфера, військово-промисловий комплекс, а також

застосовуються як основа побудови автоматизованих систем управління, засобів автоматичного регулювання та управління технологічними процесами.

Слід зазначити, що перші вбудовані системи розроблялися як спеціалізовані цифрові пристрої на базі інтегральних схем малого та середнього ступеня інтеграції. Однак, з появою мікроконтролерної та мікропроцесорної техніки, а пізніше інтегральних схем із програмованою структурою, поняття вбудованої системи сильно трансформувалося. Так, якщо перші вбудовані системи являли собою спеціалізовану структуру, яка мала у своєму складі центральний процесор, окремі інтегральні схеми контролерів периферійного обладнання, цифрових запам'ятовуючих пристроїв, то сучасні вбудовані системи реалізують вже технологію System-on-Chip (SoC) – система на кристалі [3; 4].

Під системою на кристалі (System on Chip – SoC) мають на увазі обчислювальну систему, реалізовану в інтегральному виконанні, до складу якої входить високопродуктивний процесор або кілька процесорів, математичний процесор обробки даних та цифрової обробки сигналів, додаткові модулі пам'яті, набори периферійних пристроїв (контролерів) тощо. Така організація обчислювальної системи набула широкого поширення за допомогою своєї універсальності, малого енергоспоживання і навіть можливості реконфігурації її алгоритмічної структури. Зазначимо, що на сьогодні системи на кристалі витісняють громіздкі обчислювальні структури, реалізовані за допомогою набору інтегральних схем, замінюючи їх сучасними мікроконтролерами (PIC, AVR, MSP430, STM32, Cortex-M, TSP32 тощо), програмованими логічними інтегральними схемами (ПЛІС – CPLD, FPGA, FLEX) та одноплатними комп'ютерами типу Raspberry Pi [5].

Крім цього, слід врахувати, що створення сучасних систем, які використовують технологію System-on-Chip, засноване на застосуванні високотехнологічних САПР функціональних цифрових пристроїв, що вимагає від розробників глибоких знань не тільки цифрової схемотехніки та архітектури обчислювальних систем, але і знання методів синтезу спеціалізованих пристроїв з мікропрограмним управлінням, знання мов опису апаратури та розробки програмного коду, а також методів контролепридатного синтезу. Таким чином, враховуючи вищесказане, актуальною науковою задачею є визначення технічного стану вбудованої системи реалізованої відповідно до концепції System on Chip та відновлення її правильного функціонування з припущенням, що в якості елементної бази використовуються програмовані логічні інтегральні схеми.

Аналіз останніх публікацій. На сьогодні питанням проектування та функціонування вбудованих систем присвячено велику кількість наукових праць. Так, у роботі [6] розглянуто задачу підвищення якості функціонування мікропроцесорів, що використовуються в системах контролю та управління доступом. Виділено вимоги, а також запропоновано варіант складу команд, які необхідні для якісної побудови мікропроцесорів, що працюють на основі системи залишкових класів, для контролю та управління доступом. У [7] наведено огляд засобів проектування вбудованих мікропроцесорних систем, що реалізуються на основі програмованих логічних інтегральних мікросхем та розглянуті інструменти налагодження програмного забезпечення мікропроцесорних систем на основі ядер сімейства Pico Blaze, Micro Blaze та Power PC. У [8] розглянуто задачу організації апаратної частини вбудованих мікропроцесорних систем, а також синтез елементів вбудованих систем на програмованій логіці на основі моделі програмно-керуючого автомата. У роботі [9] розглянуто модельно-орієнтоване проектування за допомогою програмних продуктів Math Works, що дозволяють автоматизувати процес розробки, налагодження та верифікації програмного забезпечення для систем управління. Крім цього у роботі [10] представлений спосіб відбору інтегральних схем по стійкості до електрорушійної сили з використанням критичної напруги живлення. У роботі [11] розглядається технологія застосування методу власного випромінювання для відзначення несправного радіоелектронного компонента в складі цифрового блока сучасного радіоелектронного обладнання за допомогою автономної автоматизованої системи діагностування.

Однак проведений аналіз показує, що на сьогодні не в повному обсязі висвітлені питання, які пов'язані з виявленням та локалізацією відмов спеціалізованих мікропроцесорних систем управління, а також оперативного автоматичного відновлення правильного їх функціонування. Крім цього необхідно враховувати, що існуючі методи контролю та діагностування, як правило, розробляються для конкретного типу інтегральних схем, що не є завжди прийнятним для застосування щодо конкретного екземпляра.

Таким чином, **метою статті** є розробка методичного підходу до відновлення правильного функціонування спеціалізованої мікропроцесорної системи управління на рівні програмованої елементної бази за результатами самодіагностування.

Виклад основного матеріалу.

Контроль та діагностування мікропроцесорних систем. Під контролем мікропроцесорних (обчислювальних) систем розуміється процес отримання інформації, що дозволяє визначити технічний стан обчислювальної системи, застосувавши апаратні, програмні та комбіновані методи та засоби контролю, встановити її відповідність вимогам, що висуваються до даного типу систем [12].

При цьому для оцінки ефективності методів контролю можна використовувати коефіцієнт якості обчислювальної системи з контролем, який визначається як ймовірність видачі безпомилкового результату перетворення інформації

$$K(t) = 1 - P_{\text{пом}}$$

де $P_{\text{пом}}$ – ймовірність пропуску помилок системою контролю при видачі результату перетворення інформації в обчислювальній системі.

На рисунку 1 представлені основні види контролю обчислювальних систем.



Рис. 1. Класифікація видів контролю обчислювальних систем

Однак слід враховувати, що представлені види контролю зазвичай застосовуються в універсальних обчислювальних системах. Разом з тим у спеціалізованих обчислювальних системах, зокрема, у вбудованих системах, що виконують обмежену кількість функціональних програм, широко застосовується контроль правильності виконання програм, який називається програмно-логічним контролем, до якого входять: контроль тривалості виконання, послідовності виконання, метод контрольних функцій і контроль гладкості [13]. Розглянемо її більш детально.

1. Контроль тривалості виконання програми заснований на тому, що для кожної програми заздалегідь відома максимальна тривалість виконання та будь-яке її перевищення означає, що програма зациклася, зупинилася або виконана неправильно. Перевищення тривалості виконання не обов'язково пов'язане з помилками у програмі, часто причиною може бути спотворення адресної інформації збоями. Можливі навіть випадки, коли почне

виконуватися одна програма, а потім результат збою виконання переходить помилково до іншої програми. Тому іноді застосовують контроль послідовності виконання підпрограм, порівнюючи номери фактично виконаних підпрограм з потрібними значеннями.

2. Метод контрольних функцій полягає в тому, що результати роботи програми повинні відповідати певним функціональним співвідношенням. Наприклад, рішення системи диференціальних рівнянь можуть бути перевірені за критерієм задоволення контрольного рівняння, утвореного у вигляді рівнянь вихідної системи.

3. Контроль гладкості заснований на тому, що якщо ряд результатів обчислень є більш-менш гладкою функцією, то будь-які різкі відхилення результату від екстрапольованого значення свідчать про помилку.

Перераховані види контролю правильності виконання програм здійснюються переважно програмними засобами. Вони дозволяють виявляти помилки в роботі обчислювальних систем із затримкою, порівнянню з часом виконання програми або підпрограми. Спільно з апаратними програмні види контролю допомагають виявити помилки, які не були виявлені апаратними засобами. Якщо ж за цільовим призначенням обчислювальної системи не потрібно швидке виявлення помилок, достатньо обмежитися програмними засобами контролю.

Далі, під діагностуванням мається на увазі процедура локалізації несправності об'єкта, тобто встановлення того, яка частина об'єкта, що діагностується, несправна. При діагностуванні проводиться встановлення несправності об'єкта на більш нижчому ієрархічному рівні, ніж під час контролю. У деяких випадках контроль обчислювальних систем сприймається як окремий випадок діагностування. Продовжуючи спускатися ієрархічною структурою об'єкта діагностування, можна доходити до будь-якого бажаного рівня ієрархії, до окремих контактних з'єднань, логічних елементів або навіть частин їх конструкції. Мірою проникнення ієрархією об'єкта є глибина діагностування. При цьому глибина діагностування вирішується, виходячи з організації процесу відновлення. З метою швидкого відновлення системи доцільно обмежитися спочатку встановленням пристрою, що відмовив. Це завдання вирішується здебільшого засобами апаратного контролю, без залучення програмних методів тестування. При цьому процедуру контролю можна розглядати як процедуру діагностування за найменшої глибини. Розглянемо найпоширеніші методи діагностування обчислювальних систем.

На сьогодні за характером взаємодії між об'єктом та засобом діагностування розрізняють тестове та функціональне діагностування. При тестовому діагностуванні на об'єкт подають спеціально підготовлені тестові дії та порівнюють реакції об'єкта на ці дії з еталонними відповідями. Цей вид діагностування застосовується тоді, коли необхідно перевірити справність функціонування або виявити несправність (дефект), що впливають на працездатність об'єкта, що перевіряється. При тестовому діагностуванні реалізуються спеціальні алгоритми, які складаються з елементарних етапів контролю. Остаточний діагноз ставиться за результатами елементарного контролю обчислювальної системи. При цьому використовуються евристичні підходи, діагностичні моделі аналітичних описів або графо-аналітичних уявлень основних властивостей об'єкта та розроблені на їхній основі алгоритми діагностування у вигляді сукупності послідовних операцій. Слід зазначити, що методи тестового діагностування містять дуже громіздкі та дорогі підготовчі операції з розробки детермінованих тестів та еталонних реакцій. При цьому виділяються три типи тестування [14]:

статистичне, коли зміна тестових наборів на виході та зняття реакцій значно нижча за частоту при роботі обчислювальної системи в реальних умовах;

динамічне, коли тестові набори подаються, а вихідні реакції аналізуються на граничних частотах роботи обчислювальної системи;

параметричне, коли перевіряються параметри обчислювальної системи, а саме: статичні – напруга, струм, опір, коефіцієнт передачі; динамічні – зміна напруги, струму, провідності, коефіцієнта передачі, тимчасові затримки тощо. Основні методи тестового діагностування обчислювальних систем представлені на рисунку 2.

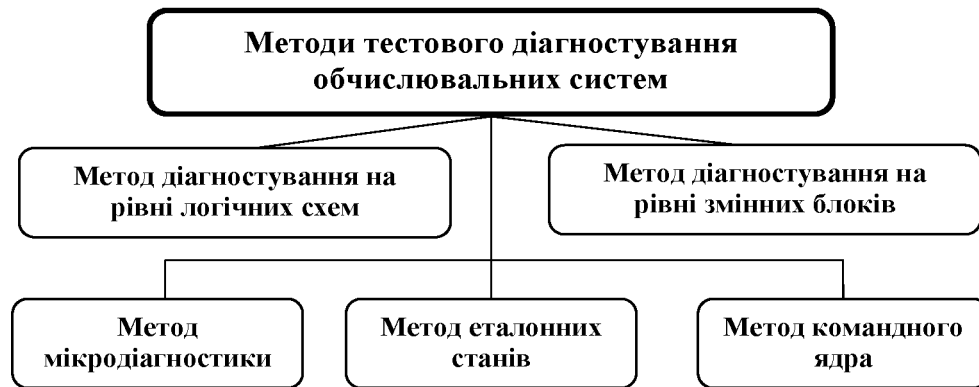


Рис. 2. Методи тестового діагностування обчислювальних систем

Другий – ймовірнісну (стохастичну) модель, яка полягає у подачі на вхід обчислювальної системи шумоподібних (випадкових та псевдовипадкових) впливів, що генеруються вбудованими генераторами, та аналіз вихідних реакцій. Основні методи функціонального діагностування обчислювальних систем представлені на рисунку 3.

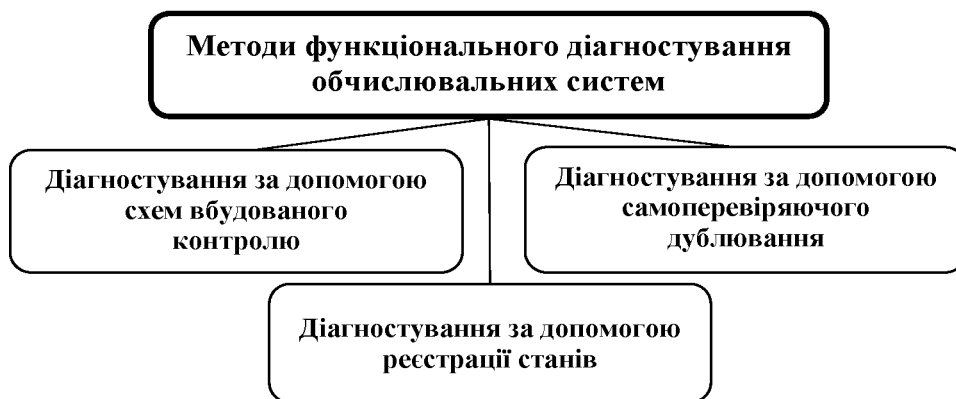


Рис. 3. Методи функціонального діагностування обчислювальних систем

Окремо слід виділити компактне тестування (сигнатурний аналіз), який відноситься як до ймовірнісних методів контролю, так і методів тестового діагностування. Суть даного методу полягає у порівнянні результатів тестування з еталоном (стиснутою довгою бітовою послідовністю з високою точністю в короткі коди – сигнатури). Це здійснюється за допомогою сигнатурних реєстрів, що реалізують поліном згортки бітових послідовностей з високою точністю. Отримані сигнатури порівнюються з еталонними, занесеними до словника сигнатур, реалізованих у вигляді дерева пошуку дефекту. А також слід виділити конкуруючий з методом сигнатурного аналізу – метод спектрограм, який дає можливість використовувати в якості діагностичних ознак розподіл відносних частот появи окремих комбінацій, утворених вихідними символами в послідовні моменти часу. Такі розподіли називають спектрограмами обчислювального пристрою, що діагностується. При цьому спектрограма може бути отримана аналітично або шляхом моделювання роботи пристрою на фіксованій вхідній послідовності [15].

Отже, розглянуті методи контролю та діагностування технічних засобів є загальною методологічною та технічною основою для прогнозування та діагностування відмов, а це у свою чергу дозволяє забезпечити необхідні показники надійності та ефективності обчислювальних систем при найменших витратах сил та засобів.

Підхід до усунення відмов шляхом реконфігурації спеціалізованої мікропроцесорної системи управління із самодіагностуванням. Мікропроцесорна система як об'єкт діагностування є складною функціональною структурою, яка містить велику кількість електронних елементів і безліч розгалужених зв'язків. Крім цього велика різноманітність мікропроцесорів відрізняються між собою набором та порядком виконання команд, часовою

організацією роботи, мають різну продуктивність, тактову частоту, розрядність, обсяг кеш-пам'яті та мікро- та макроархітектуру. Виходячи з цього, згідно з [16] при організації контролю, а також тестового та функціонального діагностування мікропроцесорних систем використовується декомпозиційний підхід, при якому в якості об'єкта контролю та діагностування виступають окремі функціональні пристрої: арифметико-логічний прилад, процесор, оперативно-запам'ятовуючий пристрій, постійно запам'ятовуючий пристрій, пристрої введення-виведення тощо. При цьому необхідно враховувати і труднощі, що виникають у процесі контролю та діагностування мікропроцесорних систем, які пов'язані з високим ступенем інтеграції ВІС/НВІС, розгалуженими зв'язками між елементами, відсутністю повної інформації про внутрішню структуру мікропроцесорної системи, а також відсутністю апаратних засобів контролю самого процесора. А враховуючи той факт, що одним з найбільш перспективних напрямків при проєктуванні мікропроцесорних систем є технологія System-on-Chip, в якості елементної база якої використовуються мікроконтролери, інтегральні схеми з програмованою структурою та одноплатні комп'ютери типу Raspberry Pi, то проблема контролю та діагностування набуває абсолютно нового характеру.

Так, у роботах [16; 17] для визначення технічного стану мікропроцесорних систем, реалізованих на ПЛІС, запропоновано використовувати засоби самодіагностування цифрових пристроїв, при цьому реалізувавши принцип взаємодії (тестування) мікропроцесорів між собою шляхом введення до складу багатопроцесорної системи сервісного процесора. Основною функцією такого процесора є контроль та діагностування багатопроцесорної системи, а також оперативне автоматичне відновлення шляхом реконфігурації системи. При цьому реалізація запропонованого принципу взаємодії та використання засобів самодіагностування наділить мікропроцесорну систему властивістю адаптації, тобто можливістю зміни параметрів, структури, управляючого впливу з метою досягнення оптимального функціонування системи при початковій невизначеності і в умовах роботи, що змінюються [18; 19]. Розглянемо детальніше принцип взаємного тестування процесорів.

Нехай система S складається з n пристроїв, причому кожен пристрій тестується підмножиною інших пристроїв. Така система може бути описана через граф $G(V, E)$, де V – множина вершин графа, що відповідають окремим пристроям, а E – множина дуг графа, направлених від пристрою, який тестує, до пристрою, який тестується, які зображують проведення тестів. Нехай ваги дуг означають: 1 – відмова пристрою, який тестується, виявлена; 0 – відмова пристрою, який тестується, не виявлена. Система S називається t_0 -діагностованою, якщо вона при заданому $G(V, E)$ забезпечує виявлення пристроїв, що відмовили, за наявності не більше ніж t_0 пристроїв, що відмовили.

Твердження. Система S є t_0 -діагностованою, якщо: 1) кількість вхідних дуг у кожній вершини графа $G(V, E)$ не менше t_0 , 2) для будь-якого цілого p за умови $0 \leq p < t_0$ і кожного $V' \subset V$ при числі вершин $|V'| = n - 2t_0 + p$, числі вершин $|\Gamma V'|$ підмножини вершин $\Gamma V'$, куди входять вихідні V' з дуги графа, більше p . При цьому $|V'|$ означає число вершин, що входять до підмножини V' .

Зазначимо, що оскільки з твердження випливає, що $n > 2t_0 + |V'| - |\Gamma V'|$, а завжди знаходиться V' , яке задовольняє умові $|V'| = |\Gamma V'|$, то $n \geq 2t_0 + 1$ – умова необхідна, але не достатня.

В якості прикладу розглянемо граф, де $t_0 = 2$ (рис. 4). Тоді $p \in \{0, 1\}$ і $|V'| = 6 - 4 + \{0, 1\} = \{2, 3\}$. З рисунка. 4 видно, що для кожного V' при $|V'| = 2$ величина

$|GV'| = 2 > 0$ і для кожного $|V'| = 3$ величина $|GV'| = 2 > 1$. Легко переконалися, що при $t_0 = 3$ ці умови не задовольняються. Отже, система на рисунку 4 є t_0 -діагностованою при $t_0 = 2$.

Визначення. Система, зображувана графом $G(V, E)$ при $t_0 \leq \text{ent}(n-1)/2$, де ent – ціла частина, є $D(n, t_0, X)$ -системою при $\vartheta_i, \vartheta_j \in E$ тоді й лише тоді, коли $(i-j) \bmod n \in X$, де $X = (x_1, x_2, \dots, x_{t_0})$ – деяка множина цілих чисел, таких як $1 \leq x_i \leq \text{ent}(n-1)/2$ для всіх $1 \leq i \leq t_0$, а $x_i < x_{i+1}$. Символ $(\vartheta_i, \vartheta_j)$ тут означає дугу графа G , яка виходить з вершини ϑ_i і входить в вершину ϑ_j .

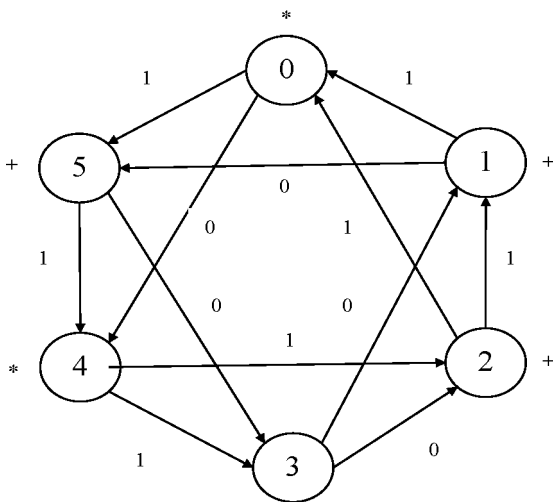


Рис. 4. Граф багатопроцесорної системи із взаємним діагностуванням для всіх $i \in [1, t_0]$. В останньому вираженні індекси у ϑ додаються за $\bmod n$.

Нехай $V^{\alpha_1 \alpha_2} = \{\vartheta_a \in V_a \mid \omega(\vartheta_b, \vartheta_a) = \alpha_1, \omega(\vartheta_a, \vartheta_i) = \alpha_2, \vartheta_b \in V_b\}$. Тоді, якщо $|V_a^{00}| \geq |V_a^{01}|$ робиться висновок про те, що вершина ϑ_i не відмовила і позначається через (+), інакше переходити до виконання етапу 3, вважаючи, що вершина ϑ_i відмовила (позначається *). При цьому $\omega(\vartheta_b, \vartheta_a)$ означає вагу дуги, з вершини ϑ_b до вершини ϑ_a . Операція встановлення працездатності вузла ϑ_i , яка записана вище в короткій символічній формі, означає, що спочатку встановлюється підмножина V_a вершин, перевіряючих вершину ϑ_i , а потім – підмножина V_b різних вершин, перевіряючих кожен по одній вершині з V_a . Вершина ϑ_i вважається працездатною тоді і лише тоді, коли кількість ланцюгів з елементів V_b через елементи V_a в ϑ_i з вагами (0, 0) більше або дорівнює, ніж кількість ланцюгів з вагами (0, 1), тобто кількість ланцюгів, де працездатність вершини ϑ_i підтверджується працездатною вершиною, більше чи дорівнює кількості ланцюгів, де непрацездатність вершини ϑ_i підтверджується вершиною з підтвердженою працездатністю.

Етап 2. Позначити кожен вершину $\vartheta \in \Gamma_{\vartheta_i}$ знаком +, якщо $\omega(\vartheta_i, \vartheta) = 0$, знаком *, якщо $\omega(\vartheta_i, \vartheta) = 1$. При цьому Γ_{ϑ_i} – підмножина вершин, до яких відносяться дуги графа, які виходять із ϑ_i .

Етап 3. Підставити $i := i - 1$. Якщо $i < 0$, то виконувати етап 4, якщо ϑ_i позначений +, то здійснити перехід до етапу 2, інакше до етапу 1.

Іншими словами, $D(n, t_0, X)$ -система – це t_0 -діагностована n -процесорна кільцева система з діагональними зв'язками, що визначаються X .

Система на рисунку 4 є $D(6, 2, \{1, 2\})$ -системою. Далі представлений алгоритм для діагностування $D(n, t_0, X)$ системи. Передбачається, що заданий граф системи з вагами дуг і потрібно визначити вершини, що відмовили і не відмовили. Алгоритм складається з наступних етапів.

Етап 1. Встановити $i = n - 1$ та побудувати підграф $G^i(V^i E^i) \subset G$, де $V_i = \{\vartheta_i\} \cup V_a \cup V_b$ і $E^i = \{(\vartheta, \vartheta_i)\}$ для всіх $\vartheta \in \{V_a\} \cup \{\vartheta_{i+x_j+x_{t_0}}, \vartheta_{i+x_j}\}$

Етап 4. Якщо кількість позначених * вершин більше t_0 , то система не діагностується, інакше підмножина вершин, позначених *, – підмножина вершин, що відмовили.

Етап 5. Кінець.

Далі представлений метод реконфігурації структури цифрових пристроїв (складових мікропроцесорної системи), який полягає в зміні внутрішніх зв'язків між логічними елементами з появою відповідного сигналу від засобів самодіагностування. Даний метод розглядає цифрові пристрої, які є складовими мікропроцесорної системи в якості динамічних управляючих систем, що функціонують в умовах несприятливих впливів. Для компенсації дій зовнішніх впливів на правильне функціонування таких систем використовується положення прескриптивної теорії, яка розглядає питання цілеспрямованого управління об'єктами різної природи, що перебувають у стані «конфлікту» з іншими об'єктами [20].

Суть даного методу полягає у відшукуванні такої надмірності структури B_i , яка при підключенні до входу модуля A (при відключеному несправному вузлі A_j) призводить до відновлення правильного функціонування модуля A (рис. 5).

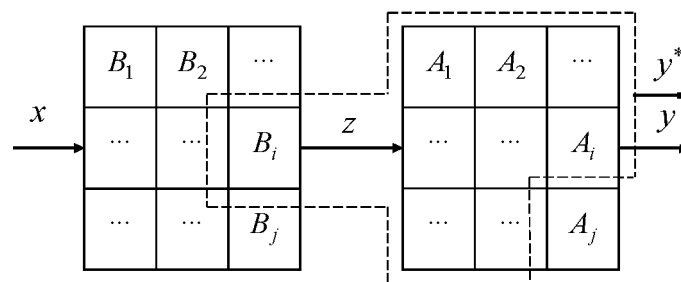


Рис. 5. Структура відновленого модуля

Модуль A , що розглядається, являє собою сукупність логічних елементів a_i , або ІР-блоків (Intellectual Property – інтелектуальна власність) для ПЛІС, який реалізує функцію

$$Y_{n-1} = \Phi_0(X_n, Y_n),$$

де $X(x_1, x_2, \dots, x_n)$ – вхідне слово, $Y(y_1, y_2, \dots, y_n)$ – вихідне слово, n – часові такти.

Потрібно синтезувати деяку систему $S \supseteq A$, що також реалізує задану функцію $\Phi_0(X_n, Y_n)$, за умови виходу з ладу будь-якої з підсистем A_j системи $A(a_i \in A)$ заданої складності розбиття C_j . Систему A можна подати у вигляді матриці M_A будь-якої з її підсистем A_j . Вилучення з матриці M_A будь-якої з її підсистем призводить до утворення матриці спотворень $M\{A_{0j}\}$, $A_{0j} = (A_0 \setminus A_j)$, що результує сукупність нових функцій $M\{\Phi_{0j}(X_n, Y_n)\}$.

Для відновлення правильного функціонування системи (реалізації функції Y) необхідно утворити матрицю $\{M_{B_j}\}$, що відновлюється. При цьому кожна підсистема B_j матриці $\{M_{B_j}\}$ підключається на вхід підсистем A_{0j} матриці спотворень $M\{\Phi_{0j}(X_n, Y_n)\}$. У загальному випадку для всіх B_i , як правило, існує перетин структур

$$\bigcap_i B_i = B_1 B_2 \dots B_j,$$

що володіє функціональними властивостями, загальними для всіх B_1, B_2, \dots, B_j або більшості з них. Але можуть існувати й окремі структури, які не містять перетинів. У цьому випадку для кожної несправності та підсистем A_j , що відключається, формування надлишкової

структури B_i на основі узагальненого модуля $\bigcap_i B_i$ утворюється відповідним підключенням вхідних $\{X\}$ і вихідних $\{Z\}$ сигналів цього модуля (рис. 6).

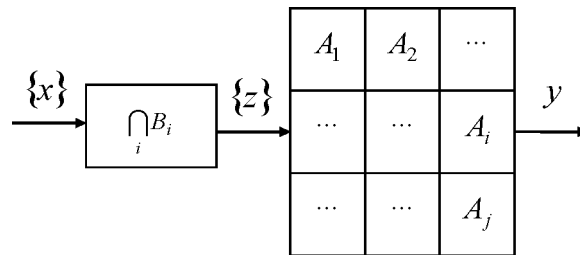


Рис. 6. Механізм відновлення модуля

Змістом функціонально-надійного синтезу системи A_0 є визначення правил Ψ опису підсистеми B_j матриці $M\{B_j\}$ при вилученні будь-якої підсистеми A_j матриці M_A . При цьому зауважимо, що правило Ψ повинно індукувати як підсистеми B_j , що відновлюються, за A_{0j} : $B_j = \Psi A_{0j}$, так і матрицю $M\{B_j\}$, яка відновлюється, підматриці спотворень:

$$M\{B_j\} = \Psi\{M_{A_j}\}.$$

Таким чином, розглянутий метод реконфігурації цифрових пристроїв дозволяє визначити структуру надмірних підсистем B_j залежно від відключених несправних підсистем A_j даного пристрою A_0 . При цьому розрахована структура B_j й частина, яка залишилася справною, $A_{0j} = A_0 \setminus A_j$ реалізує задану функцію Y_{n+1} .

Зазначимо, що при реконфігурації цифрових пристроїв обов'язковим є контролюючий (діагностуючий) пристрій. А враховуючи той факт, що його структура і функціональні завдання, які він виконує, досить складні, то доцільно розробити самодіагностуючий пристрій. Принцип побудови таких пристроїв може бути заснований на методі реконфігурації надмірних цифрових пристроїв. В цьому випадку цифровий пристрій A також поділяється на вузли A_1, A_2, \dots, A_j і залежно від цього поділу будується структура надмірного пристрою B , що складається з схем B_1, B_2, \dots, B_j . Пристрій із самодіагностуванням і самовідновленням, зображений на рисунку 7, містить реконфігуратор R , що забезпечує відповідну реконфігурацію пристроїв A і B , та два контрольні регістри запису результатів розрахунків P_1 і P_2 .

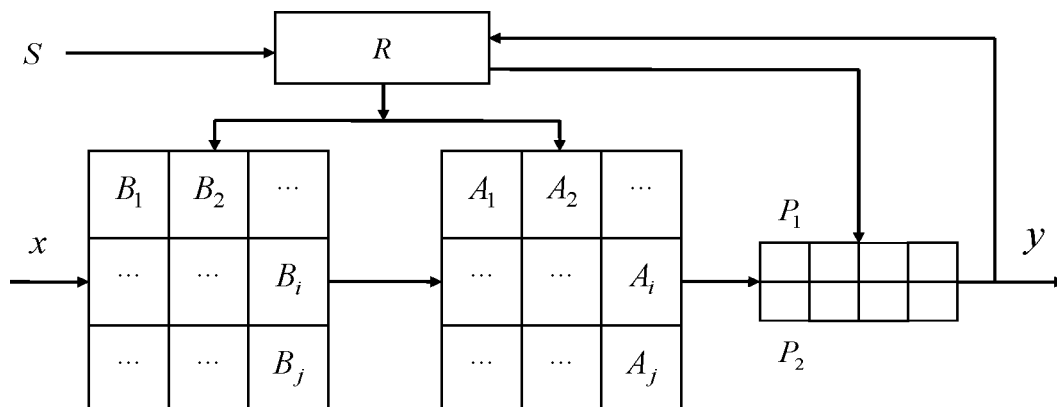


Рис. 7. Структура надмірного цифрового пристрою із самодіагностуванням

Принцип роботи полягає в тому, що цифровий пристрій A , що складається з вузлів A_1, A_2, \dots, A_j , може бути розділений на три укрупнені блоки A_I, A_{II}, A_{III} (рис. 8, *а*). Далі, за відсутності несправностей функціонує весь пристрій A . Після певного кроку обчислень в реконфігуратор R подається сигнал S самоконтролю. У цьому випадку в пристрій вводиться тестова програма і в перший регістр P_1 записується проміжний результат обчислень (перший крок). Перед другим кроком обчислень реконфігуратор R відключає частину пристроїв, наприклад A_I , і підключає до нього відповідно пристрій B , наприклад B_I . На вхід пристрою, що утворився, $B_I - A_{II} - A_{III}$ подається та ж тестова програма, що і на першому кроці обчислень, а результат обчислень записується в другий регістр P_2 (другий крок). Реконфігурація пристроїв цього типу представлена на рисунку 8, *б*.

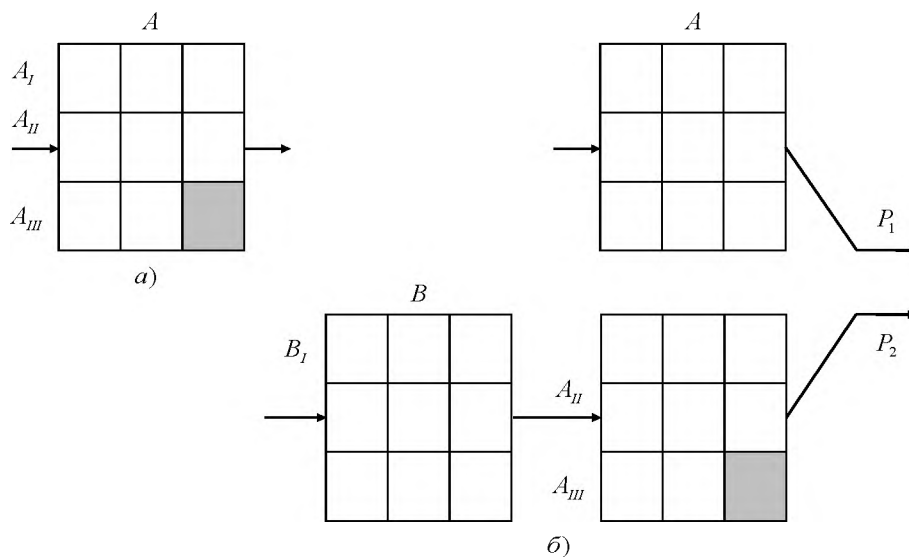


Рис. 8. Принцип реконфігурації цифрового пристрою із самодіагностуванням:
а – укрупнений цифровий пристрій A ; *б* – цифровий пристрій A з надмірною структурою B

Якщо вміст регістрів P_1 і P_2 збігаються, то пристрій A продовжує функціонувати. Якщо ж результати обчислень P_1 і P_2 різні, то це означає, що несправний один з блоків пристрою A . Зазначимо, що в якості сигналу S може бути використаний один з методів тестового діагностування, представлених на рисунку 2.

Отже, розглянутий метод реконфігурації із самодіагностуванням дає можливість не тільки визначати технічний стан цифрового пристрою, але й відновлювати правильне його функціонування шляхом перебудови внутрішньої структури.

Висновки. Запропоновано методичний підхід відновлення правильного функціонування цифрових пристроїв, які є основою побудови спеціалізованої мікропроцесорної системи управління, реалізованої за технологією «система на кристалі». В основі цього підходу лежать методи контролю та діагностування спеціалізованих мікропроцесорних систем, а також метод реконфігурації цифрових надлишкових структур із засобами самодіагностування на рівні булевих рівнянь. Реалізація даного підходу при проектуванні спеціалізованих мікропроцесорних систем управління на сучасній елементній базі надасть змогу підвищити надійність не лише системи, яка розглядається, а й усієї системи управління складними об'єктами та технологічними процесами в загалом.

Напрямами подальшої роботи є проектування адаптивної мікропроцесорної системи управління з вбудованою інтелектуальною системою розпізнавання технічного стану та системою оперативного автоматичного відновлення правильного функціонування, здатної протидіяти несприятливим впливам, як навмисного так і ненавмисного характеру.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. A. Crespo, P. Albertos, J. Simó, Embedded control systems: from design to implementation, *Ifac Proceedings Volumes, Volume 40, Issue 1, 2007, Pages 25–32, ISSN 1474-6670, ISBN 9783902661210. DOI: 10.3182/20070213-3-CU-2913.00006.*
2. Smit, Wim & Hendriksen, Wim. (2004). Embedded systems: Smart and intelligent tools in an increasingly interconnected globalised world. *International Journal of Technology Policy and Management. Vol. 4. pp. 309–323. DOI:10.1504/IJTPM.2004.006614.*
3. H. De Man, System-on-Chip design: impact on education and research, in *IEEE Design & Test of Computers*, vol. 16, no. 3, pp. 11–19, July – Sept. 1999. DOI: 10.1109/54.785820.
4. Иванюк А. А. Проектирование встраиваемых цифровых устройств и систем: монография / А. А. Иванюк. Минск: Бестпринт, 2012. 337 с. ISBN 978-985-6873-47-1.
5. Штаненко С. С. Технологія System-on-Chip як основа підвищення живучості складних технічних систем / С. С. Штаненко, Ю. Я. Самохвалов // *Сучасна спеціальна техніка: ДНДІ МВС України. Київ. 2021. № 3(66). С. 31–43. ISSN: 2411-3816.*
6. Магомедов Ш. Г. Проектирование микропроцессорных устройств, разработанных для систем контроля и управления / Ш. Г. Магомедов. *Cloud of Science. 2019. Т. 6. № 4. С. 752–761. UBL: http://cloudofscience.ru.*
7. Зотов В. Ю. Средства проектирования встраиваемых микропроцессорных систем, реализуемых на основе ПЛИС фирмы Xilinx. М.: Современная электроника, № 9. 2006. С. 64–70.
8. M. N. Granieri and F. J. Levy, "Embedded diagnostic system design using an automated diagnostic tool set," *AUTOTESTCON 93*, San Antonio, TX, USA, 1993, pp. 645–649. DOI: 10.1109/AUTEST.1993.396294.
9. Топораш Г. К. Модельно-ориентированное проектирование программного обеспечения для встраиваемых систем в среде Matlab/Simulink / Г. К. Топораш, А. В. Мазур, Д. А. Ковальчук, А. А. Пушкин // *Автоматизація технологічних і бізнес-процесів. 2014. № 17. С. 26–29. DOI: 10.15673/АТБП2312-3125.17/2014.26326.*
10. Горлов М. И. Диагностический контроль интегральных схем по измерению критического напряжения питания / М. И. Горлов, А. В. Строгонов, А. В. Арсентьев, А. А. Винокуров // *Энергия. Воронеж: ЗАО «Орбита», 2017. С. 29–46.*
11. Кузавков В. В. Методика локалізації несправного радіоелектронного компоненту / В. В. Кузавков, О. Г. Янковський // *Збірник наукових праць Одеської державної академії технічного регулювання та якості, 2015. Вип. 1. С. 36–41.*
12. Гуляев В. А. Техническая диагностика управляющих систем. Киев: Наукова думка, 1983. 208 с.
13. Jäger, Reinhold. (1991). Computer diagnostics – a survey: Practical applications of computerized assessment: Theoretical principles and perspectives. *European Review of Applied Psychology / Revue Européenne de Psychologie Appliquée. 41. 247–268.*
14. Иьуду К. А. Надежность, контроль и диагностика вычислительных машин и систем. М.: Высшая школа, 1989. 216 с.
15. Ручко В. В. Контроль цифровых схем с помощью спектрограмм / В. В. Ручко, Ю. Г. Савченко, А. В. Хмелевая // *Управляющие системы и машины. Киев: УСИМ. 1984. № 3 (71). С. 28–31.*
16. Штаненко С. С. Мікропроцесорні системи на програмованих логічних інтегральних схемах як об'єкт діагностики / С. С. Штаненко, Ю. Я. Самохвалов, О. Ю. Іюхов, В. Г. Малюк // *Сучасні інформаційні системи = Advanced Information Systems. 2022. Т. 6. № 1. С. 81–87. DOI: 10.20998/2522-9052.2022.1.14.*
17. Погребинский С. Б. Проектирование и надежность многопроцессорных ЭВМ / С. Б. Погребинский, В. П. Стрельников. М.: Радио и связь, 1988. 168 с.
18. Герасимов Б. М. Інтелектуальні системи підтримки прийняття рішень / Б. М. Герасимов, В. М. Локазюк, О. Г. Оксіюк, О. В. Поморова. К: Вид-во Європ. ун-ту, 2007. 335 с.
19. Штаненко С. С. Адаптація мікропроцесорних систем управління до несприятливих впливів / С. С. Штаненко, Ю. Я. Самохвалов // *Сучасна спеціальна техніка: ДНДІ МВС України. Київ. 2022. № 3 (70). С. 89–100. ISSN: 2411-3816.*
20. Обухов В. Е. Синтез избыточных дискретных устройств с реконфигурацией структуры / В. Е. Обухов, В. В. Павлов. К.: Наукова думка, 1979. 156 с.