

Б. Ю. Волочій, доктор технічних наук, професор кафедри теоретичної радіотехніки та радіовимірювань Національного університету "Львівська політехніка"

Л. Д. Озірковський, кандидат технічних наук, доцент кафедри теоретичної радіотехніки та радіовимірювань Національного університету "Львівська політехніка"

А. В. Мащак, аспірант кафедри теоретичної радіотехніки та радіовимірювань Національного університету "Львівська політехніка"

О. П. Шкілюк, аспірант кафедри теоретичної радіотехніки та радіовимірювань Національного університету "Львівська політехніка"

МЕТОДИКА ПОБУДОВИ ДЕРЕВА ВІДМОВ СКЛАДНОЇ ТЕХНІЧНОЇ СИСТЕМИ НА ОСНОВІ ГРАФУ СТАНІВ І ПЕРЕХОДІВ

Запропоновано методику, за якою мінімальні перерізи визначаються не з дерева відмов об'єкта дослідження, а з його моделі у вигляді графу станів і переходів. Власне дерево відмов будується на основі мінімальних перерізів. Такий підхід дає змогу перевірити достовірність оцінок безпечності програмно-технічних комплексів.

Ключові слова: безпечність експлуатації; дерево відмов; складна технічна система; автоматизація побудови дерева відмов; оцінка безпечності.

In this paper the technique, which allows to obtain a minimal cut sets not from fault tree, but from the model as graph of states and transitions is presented. The fault tree is based on minimum cross sections. This approach allows to validate software and hardware systems safety assessment.

Key words: safety operation; fault tree; complex technical system; building automation fault tree; safety assessment.

Постановка проблеми. Умовою ефективного функціонування сучасних складних технічних систем (СТС) є забезпечення заданого рівня надійності та безпечності. Наслідки відмови СТС потенційно можуть завдавати великих фінансових втрат і призводити до людських жертв. Тому актуально забезпечити заданий рівень надійності та оцінки потенційних наслідків аварійних ситуацій ще на етапі системотехнічного проектування таких систем [1; 2]. Оцінку надійності та безпечності СТС здійснюють за допомогою моделювання, на основі якого розраховують відповідні показники.

Для отримання показників надійності нині розроблено велику кількість методів, а саме: методи імітаційного моделювання, методи логіко-імовірнісного моделювання (ЛІМ), метод простору станів. Найчастіше застосовують методи ЛІМ [3]. До позитивних рис методів ЛІМ можна зарахувати те, що для них розроблено формальні алгоритми отримання моделей. Таким чином, методи ЛІМ отримали можливість їх автоматизації і створення відповідного програмного забезпечення.

© Б. Ю. Волочій, Л. Д. Озірковський, А. В. Мащак, О. П. Шкілюк, 2014

Найпоширеніший метод логіко-імовірнісного моделювання СТС передбачає побудову дерева відмов (ДВ) [4]. Дерево відмов – різновид структурних схем, який дає змогу дедуктивним способом графічно описати події, котрі призводять до катастрофічної відмови системи [5]. Результат аналізу ДВ – якісне та кількісне зображення подій, які можуть призвести до катастрофічної відмови системи, у вигляді мінімальних перерізів (МП) і ймовірностей виникнення МП. Отже, мінімальні перерізи – це така мінімальна комбінація подій, що призводить до катастрофічної відмови системи. Якщо з МП вилучити хоча б одну подію, то відмова системи в цілому унеможливилюється.

Аналіз останніх досліджень і публікацій. У сучасній літературі багато праць присвячено побудові та аналізу ДВ. Так, у дослідженнях [4–7] подано загальні принципи та особливості побудови ДВ. У працях [8–14] розглянуто методики побудови та аналізу ДВ із подальшим їх застосуванням на конкретних системах.

У праці [8] автори подають методику побудови та аналізу динамічних ДВ. Позитивна особливість методики полягає у тому, що розробник отримує ДВ з урахуванням послідовності виникнення подій у часі та подій, що повторюються. Після побудови динамічного ДВ для подальшого дослідження системи науковці аналізують даний ДВ із використанням імітаційного моделювання (ІМ) методом Монте-Карло. Однак слід зазначити, що використання імітаційного моделювання для аналізу динамічного ДВ значно збільшує часові затрати порівняно з аналізом статичних ДВ.

Автори праці [9] пропонують методику побудови статичного ДВ. Особливість методики побудови ДВ – введення до дерева блоків із часовими залежностями. Цю методику застосовували для побудови моделі телекомунікаційної мережі. Систему перевірено на надійність.

У [10] досліджено методику, що дозволяє проводити аналіз шляхом декомпозиції громіздких, складних ДВ на прості. У результаті декомпозиції одного складного ДВ отримують декілька дерев меншої розмірності, котрі піддаються аналізу. Після проведення аналізу отриманих менших ДВ за допомогою запропонованого авторами методу проводиться процедура об'єднання результатів аналізу менших ДВ в один. Результат даного методу – мінімальні перерізи системи (МП) і показник надійності – ймовірність відмови системи. Зазначимо, що через проведення декомпозиції та об'єднання ДВ подана методика вкладає похибку у певне значення ймовірності відмови системи.

Динамічні ДВ, в яких закладено функціональну взаємозалежність подій, автори подають у праці [11]. До позитивних особливостей цієї методики побудови дерева відмови можна зарахувати використання у ДВ блоків з пріоритетом і блоків повторення. За допомогою зазначеного ДВ отримуємо ймовірність виникнення катастрофічної відмови системи. Отримані ДВ враховують кореляцію між відмовами, однак їх подальший аналіз потребує використання методів ІМ або методу простору станів.

У дослідженні [12] подано аналіз некогерентних ДВ, які ґрунтуються на перерозподілі ймовірностей виникнення подій у ДВ. У моделі закладено процедуру зміни інтенсивності виникнення відмови за певної умови. Такий метод застосовано під час розробки моделі системи аварійного відключення на ядерній електростанції. Даний підхід досить складний і потребує від розробника значних інтелектуальних та часових затрат.

Два методи: аналіз на основі ДВ та аналіз видів, причин і наслідків відмов порівнюються в науковій праці [13], де також подано результат порівняння цих двох методів, із зазначенням внеску кожного для реалізації структурованої інтелектуальної розробки системи управління гідравлічної турбіни та її технічного обслуговування.

Основою методики аналізу надійності телекомунікаційних систем у дослідженні [14] є побудова й аналіз ДВ. У запропонованих методах автор подає принципи розробки моделей телекомунікаційних систем і моделей систем їх технічної експлуатації для забезпечення їх надійності та

визначення критичності відмов. Побудова моделей технічної експлуатації підкреслює важливість урахування в них не лише структури технічної системи, а також її функціональної поведінки.

У праці [15] автори пропонують методику побудови ДВ із урахуванням перерозподілу навантаження елементів. Запропонована методика складна у використанні й потребує великих часових затрат у розробці моделі.

Отже, сучасні методики побудови та аналізу ДВ у більшості випадків потребують від розробника “ручної” побудови дерева. Слід зауважити, що ДВ статичні, оскільки вважають, що всі події в системі взаємозалежні. Для використання методів побудови динамічних ДВ, у котрих закладено взаємозалежність між подіями, необхідно для подальшого аналізу залучати складні методи. Це може бути метод ІМ чи метод простору станів, що потребує значних часових та інтелектуальних затрат.

У відомих методиках побудови та аналізу ДВ в одній моделі не поєднується надійнісна та функціональна поведінка СТС, не враховується технічне обслуговування й обмеження на кількість ремонтів, наявність і можливість контролю та діагностики, надійність програмного забезпечення.

Тому актуальне завдання розробки методики, котра дала б можливість урахувати вищезазначені особливості СТС зі ступенем формалізації, що дозволяє автоматизувати процес отримання ДВ і МП системи.

Мета статті – запропонувати методику, за якою мінімальні перерізи визначаються не з дерева відмов об’єкта дослідження, а з його моделі у вигляді графу станів і переходів.

Виклад основного матеріалу. Новий підхід до автоматизації побудови дерева відмов СТС базується на тому, що знайдено метод визначення МП безпосередньо з моделі об’єкта дослідження у вигляді графу станів і переходів [16]. Створити методику побудови ДВ з високим ступенем формалізації дозволив метод простору станів [17], а саме його вдосконалена версія [18; 19], де високий рівень формалізації має методика розробки графу станів і переходів, що дала можливість автоматизувати процес побудови математичних моделей поведінки СТС у вигляді дискретно-неперервних стохастичних систем.

Автоматизація процесу побудови математичних моделей поведінки СТС базується на структурно-автоматній моделі (САМ) об’єкта дослідження. Структурно-автоматна модель СТС – це формалізоване подання її структури та поведінки, що включає в себе: параметри об’єкта дослідження, які мають бути відображені в моделі; вектор стану (ВС) об’єкта дослідження; дерево правил модифікації компонент вектора стану. На основі САМ формується граф станів і переходів. Однак ступінь деталізації подання об’єкта дослідження в такому графі недостатній для побудови на його основі дерева відмов. Для отримання ДВ удосконалено метод побудови САМ об’єкта дослідження для отримання бінарної САМ.

Бінарна САМ – це структурно-автоматна модель об’єкта дослідження, в якій усі елементи структури системи відображаються окремими компонентами ВС і можуть набувати лише двох значень – 0 та 1. За допомогою бінарної САМ, на відміну від звичайної САМ [15; 16], можна описати структуру й поведінку СТС без об’єднання станів для елементів її структури. Крім цього, бінарна САМ дає можливість отримати розщеплений стан відмови, в якому розрізняють стани непрацездатних підсистем СТС з необхідним рівнем деталізації.

За допомогою програмного модуля ASNA [17], для якого бінарна САМ надає вхідні дані про об’єкт дослідження, проєктант отримує граф станів і переходів з необхідним для побудови ДВ рівнем деталізації. На основі цього графу формується аналітична модель СТС у вигляді системи диференціальних рівнянь Колмогорова–Чепмена. Розв’язок системи диференціальних рівнянь дає розподіл імовірностей перебування СТС у всіх можливих станах. Використанням запропонованої процедури фільтрації непрацездатних станів формується масив мінімальних перерізів [14]. Кількісні показники МП отримують шляхом додавання ймовірностей перебування системи у відповідних станах. Методику побудови ДВ зі ступенем формалізації, який робить її придатною для автоматизації, подано на рис. 1.

Вхідними даними для побудови ДВ є масив МП. Відповідно до запропонованої процедури розробник отримує логічну функцію, за допомогою якої відбувається графічна побудова ДВ.

Автоматизована побудова ДВ виконується у два етапи. На першому етапі отримується логічна функція ДВ шляхом послідовного перебирання масиву МП. На другому – на основі отриманої логічної функції відбувається графічна побудова ДВ.

Слід зазначити, що графічна побудова ДВ статична, але ймовірнісні показники МП ураховують динамічні процеси, котрі відбуваються в СТС.



Рис. 1. Методика побудови дерева відмов

За поданою методикою розроблено алгоритм, який будує логічну функцію ДВ і реалізований у пілотному програмному засобі.

А. Алгоритм отримання логічної функції дерева відмов

У даному алгоритмі реалізації першого етапу відбувається послідовний запис логічної функції ДВ.

Скорочення, які використовуються в описі алгоритму автоматизованої побудови ДВ:

n – вказівник порядкового номера МП;

i – вказівник порядкового номера компонента ВС;

j – лічильник компонентів ВС, значення яких дорівнює нулю;

ПКВС – постійна кількість компонентів вектора стану;

ЛНС n – лічильник нулів перерізу, фіксує для МП з порядковим номером n кількість компонентів ВС, значення яких дорівнює нулю;

ПКМС – постійна кількість МП;

ВС n [i] – значення i -го компонента вектора стану МП з порядковим номером n ;

ФЛФ – форма логічної функції.

Крок 1. Беремо перший мінімальний переріз. У (n) записується 1, а лічильник нулів січення (ЛНС1) першого МП фіксує кількість компонентів ВС, значення яких дорівнює нулю. Лічильник компонентів ВС (j) отримує значення кількості компонентів ВС, значення, що дорівнюють нулю $-j = \text{ЛНС1}$.

Крок 2. Беремо перший компонент вектора стану першого МП системи $-i = 1$. У ФЛФ записуємо символ “(”.

Крок 3. На цьому кроці слід визначити значення i -го компонента вектора стану МП з порядковим номером n (під час першої ітерації – першого компонента ВС першого МП), причому можливі два варіанти:

Якщо значення i -го компонента вектора стану МП дорівнює нулю ($\text{ВС}n[i] = 0$), то до форми логічної функції (ФЛФ) записується значення вказівника поточного компонента ВС – $V “i”$. Далі треба зменшити на одиницю стан лічильника компонентів ВС, значення яких дорівнює нулю $-j = j - 1$ та перевірити чи $j < 0$.

Якщо $j > 0$, то це означає що в даному МП ще існують компоненти ВС, значення яких дорівнює нулю, тому в ФЛФ записуємо знак множення “ \times ” та переходимо до кроку 4. Якщо $j = 0$, то це означає, що в даному МП більше немає компонентів ВС, значення яких дорівнює нулю, тому слід перейти до кроку 4.

Якщо значення i -го компонента вектора стану МП більше нуля ($\text{ВС}n[i] > 0$), то слід перейти до кроку 4.

Крок 4. Збільшуємо вказівник компонента ВС (i) поточного МП з порядковим номером n на одиницю $-i = i + 1$ та перевіряємо чи вказівник не зайшов за останнє значення компонентів ВС. Перевірка відбувається шляхом порівняння вказівника компонента ВС зі значенням постійної компонентів вектора стану (ПКВС):

Якщо $i \leq \text{ПКВС}$, то необхідно перейти до кроку 3.

Якщо $i > \text{ПКВС}$, то переходимо до кроку 5.

Крок 5. Збільшуємо вказівник МП n на одиницю $-n = n + 1$. Тепер слід перевірити чи не перевищило поточне значення вказівника порядкового номера МП кількість МП. Перевірка проводиться шляхом порівняння поточного інкрементованого вказівника МП із постійною кількістю МП (ПКМП):

Якщо $n \leq \text{ПКВС}$, то в ФЛФ записуємо символ “)” і знак додавання “+”, переходимо до кроку 2.

Якщо $n > \text{ПКВС}$, то у ФЛФ записуємо символи “);”. На цьому кроці процедуру запису логічної функції ДВ закінчено.

В. Алгоритм графічної побудови дерева відмов

Крок 1. Побудова нижнього рівня ДВ. Для отримання графічного зображення ДВ усі елементи, які в дужках і між якими є знак множення “ \times ”, необхідно об’єднати логічними елементами AND так, щоб із кожної групи елементів, між якими були знаки “ \times ”, мали по одному виходу. В результаті отримуємо нижній рівень ДВ. Після побудови нижнього рівня слід перейти до кроку 2.

Крок 2. Формуються такі рівні ДВ і вершинна подія. Для цього слід об’єднати всі виходи нижнього рівня ДВ логічними елементами OR згідно з правилом об’єднання елементів. Так отримується ДВ з однією подією найвищого рівня.

Правило об’єднання елементів: одним логічним елементом AND чи OR можна об’єднати лише два виходи з елементів нижнього рівня дерева.

Об’єкт дослідження – відмовостійка програмно-апаратна система (ПАС), яка складається з чотирьох основних модулів: вхідного приймального, операційного, обробки інформації та виконавчого, котрі виконують цільову функцію. Вхідний приймальний модуль і модуль обробки інформації зарезервовані. Відмова ПАС може виникнути як внаслідок від-

мови модулів системи, так і внаслідок відмови програмного забезпечення. Структурну схему відмовостійкої ПАС зображено на рис. 2.

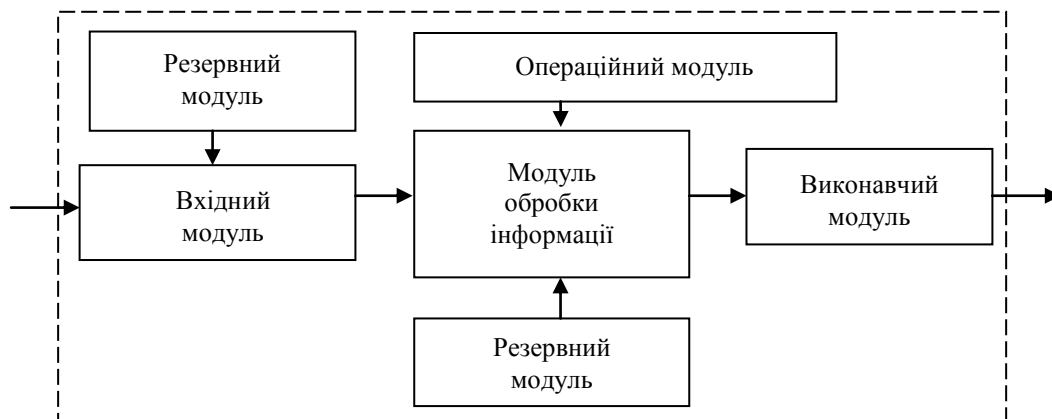


Рис. 2. Структурна схема відмовостійкої програмно-апаратної системи

За методикою, поданою в статті [14], для відмовостійкої ПАС отримано масив МП (табл. 1).

Таблиця 1

Масив мінімальних перерізів

Порядковий номер МП	Компоненти вектора стану
1	V1=1; V2=1; V3=1; V4=1; V5=1; V6=1; V7=0
2	V1=1; V2=1; V3=1; V4=1; V5=0; V6=0; V7=1
3	V1=1; V2=1; V3=0; V4=0; V5=1; V6=1; V7=1
4	V1=0; V2=0; V3=1; V4=1; V5=1; V6=1; V7=1

Вхідні дані:
 ПКВС – присвоюється 7.
 ПКМС – присвоюється 4.
А. Алгоритм автоматизованої побудови дерева відмов
 Крок 1. $n = 1; j = 1$.
 Крок 2. $i = 1$.
 У ФЛФ записуємо символ “(”, переходимо до кроку 3.
 Крок 3. Беремо значення 1-го компонента вектора стану 1-го МП. Так як $(BC1 [1] = 1) > 0$, то переходимо до кроку 4.
 Крок 4. $i = i + 1; i = 1 + 1 = 2$. Оскільки $2 < 7 (i \leq ПКВС)$, переходимо до кроку 3.
 Крок 3. Беремо значення 2-го компонента вектора стану 1-го МП. $(BC1 [2] = 1) > 0$, переходимо до кроку 4.
 Крок 4. $i = i + 1; i = 2 + 1 = 3; 3 < 7 (i \leq ПКВС)$, переходимо до кроку 3.
 Крок 3. Беремо значення 3-го компонента вектора стану 1-го МП; $(BC1 [3] = 1) > 0$, переходимо до кроку 4.

Крок 4. $i = i + 1; i = 3 + 1 = 4; 4 < 7 (i \leq \text{ПКВС})$, переходимо до кроку 3.
Крок 3. Беремо значення 4-го компонента вектора стану 1-го МП; $(\text{BC1} [4] = 1) > 0$, переходимо до кроку 4.
Крок 4. $i = i + 1; i = 4 + 1 = 5; 5 < 7 (i \leq \text{ПКВС})$, переходимо до кроку 3.
Крок 3. Беремо значення 5-го компонента вектора стану 1-го МП; $(\text{BC1} [5] = 1) > 0$, переходимо до кроку 4.
Крок 4. $i = i + 1; i = 5 + 1 = 6; 6 < 7 (i \leq \text{ПКВС})$, переходимо до кроку 3.
Крок 3. Беремо значення 6-го компонента вектора стану 1-го МП; $(\text{BC1} [6] = 1) > 0$, переходимо до кроку 4.
Крок 4. $i = i + 1; i = 6 + 1 = 7; 7 = 7 (i \leq \text{ПКВС})$, переходимо до кроку 3.
Крок 3. Беремо значення 7-го компонента вектора стану 1-го МП. $\text{BC1} [7] = 0$;
У ФЛФ записуємо $V_i: V_7$
ФЛФ: (V_7)
 $j = j - 1; j = 1 - 1 = 0; j = 0$, переходимо до кроку 4.
Крок 4. $i = i + 1; i = 7 + 1 = 8; 8 > 7 (i > \text{ПКВС})$, переходимо до кроку 5.
Крок 5. $n = n + 1; n = 1 + 1 = 2$; Оскільки $(2 < 4) n \leq \text{ПКВС}$, у ФЛФ записуємо символ “)”, далі у ФЛФ записуємо знак додавання “+”, переходимо до кроку 2.
ФЛФ: $(V_7) +$
Крок 2. $i = 1; j = 2$; у ФЛФ записуємо символ “(”;
ФЛФ: $(V_7) + ($
Крок 3. Беремо значення 1-го компонента вектора стану 2-го МП. $(\text{BC2} [1] = 1) > 0$, переходимо до кроку 4.
Крок 4. $i = i + 1; i = 1 + 1 = 2; 2 < 7 (i \leq \text{ПКВС})$, переходимо до кроку 3.
Крок 3. Беремо значення 2-го компонента вектора стану 2-го МП; $(\text{BC2} [2] = 1) > 0$, переходимо до кроку 4.
Крок 4. $i = i + 1; i = 2 + 1 = 3$; оскільки $3 < 7 (i \leq \text{ПКВС})$, то переходимо до кроку 3.
Крок 3. Беремо значення 3-го компонента вектора стану 2-го МП. $(\text{BC2} [3] = 1) > 0$, переходимо до кроку 4.
Крок 4. $i = i + 1; i = 3 + 1 = 4; 4 < 7 (i \leq \text{ПКВС})$ переходимо до кроку 3.
Крок 3. Беремо значення 4-го компонента вектора стану 2-го МП. $(\text{BC2} [4] = 1) > 0$, переходимо до кроку 4.
Крок 4. $i = i + 1; i = 4 + 1 = 5; 5 < 7 (i \leq \text{ПКВС})$ переходимо до кроку 3.
Крок 3. Беремо значення 5-го компонента вектора стану 2-го МП. $\text{BC2} [5] = 0$; у ФЛФ записуємо значення $V_i: V_5$
ФЛФ: $(V_7) + (V_5 \times$
 $j = j - 1; j = 2 - 1 = 1$;
 $j > 0$ записується знак множення “ \times ”; далі потрібно перейти до кроку 4.
Крок 4. $i = i + 1; i = 5 + 1 = 6; 6 < 7 (i \leq \text{ПКВС})$, переходимо до кроку 3.
Крок 3. Беремо значення 6-го компонента вектора стану 2-го МП. $\text{BC2} [6] = 0$; у ФЛФ записуємо значення вектора V_i :
ФЛФ: $(V_7) + (V_5 \times V_6$
 $j = j - 1; j = 1 - 1 = 0$; потрібно перейти до кроку 4.
Крок 4. $i = i + 1; i = 6 + 1 = 7; 7 = 7 (i \leq \text{ПКВС})$ переходимо до кроку 3.
Крок 3. Беремо значення 7-го компонента вектора стану 2-го МП. $(\text{BC2} [7] = 1) > 0$, переходимо до кроку 4.
Крок 4. $i = i + 1; i = 7 + 1 = 8$; якщо $i > \text{ПКВС}$, то переходимо до кроку 5.
Крок 5. $n = n + 1; n = 2 + 1 = 3$; оскільки $3 \leq 4$ ФЛФ, записуємо символ “)”; далі виконуємо умову $n < \text{ПКВС}$, записуємо знак додавання “+”, переходимо до кроку 2.
ФЛФ: $(V_7) + (V_5 \times V_6) +$

Крок 2. $i = 1; j = 2$; у ФЛФ записуємо символ “(”); переходимо до кроку 3:
 ФЛФ: $(V7) + (V5 \times V6) + ($

Крок 3. Беремо значення 1-го компонента вектора стану 3-го МП. $(BC3 [1] = 1) > 0$, переходимо до кроку 4.

Крок 4. $i = i + 1; i = 1 + 1 = 2; 2 < 7 (i \leq PKBC)$, переходимо до кроку 3.

Крок 3. Беремо значення 2-го компонента вектора стану 3-го МП. $(BC3 [1] = 1) > 0$, переходимо до кроку 4.

Крок 4. $i = i + 1; i = 2 + 1 = 3; 3 < 7 (i \leq PKBC)$, переходимо до кроку 3.

Крок 3. $BC3 [3] = 0$; у ФЛФ записуємо значення вектора V_i :
 ФЛФ: $(V7) + (V5 \times V6) + (V3$

$j = j - 1; j = 2 - 1 = 1$; оскільки $j > 0$, то записуємо “ \times ”, далі потрібно перейти до кроку 4.
 ФЛФ: $(V7) + (V5 \times V6) + (V3 \times$

Крок 4. $i = i + 1; i = 3 + 1 = 4; 4 < 7 (i \leq PKBC)$, переходимо до кроку 3.

Крок 3. $BC1 [4] = 0$; у ФЛФ записуємо значення вектора V_i :
 ФЛФ: $(V7) + (V5 \times V6) + (V3 \times V4$

$j = j - 1; j = 2 - 1 = 1; j = 0$, то потрібно перейти до кроку 4.

Крок 4. $i = i + 1; i = 4 + 1 = 5; 5 < 7 (i \leq PKBC)$, переходимо до кроку 3.

Крок 3. Беремо значення 5-го компонента вектора стану 3-го МП. $BC3 [5] = 1$; далі слід перейти до кроку 4.

Крок 4. $i = i + 1; i = 5 + 1 = 6; 6 < 7 (i \leq PKBC)$, переходимо до кроку 3.

Крок 3. Беремо значення 6-го компонента вектора стану 3-го МП. $(BC3 [6] = 1) > 0$, переходимо до кроку 4.

Крок 4. $i = i + 1; i = 6 + 1 = 7; 7 = 7 (i \leq PKBC)$, переходимо до кроку 3.

Крок 3. Беремо значення 7-го компонента вектора стану 3-го МП. $(BC3 [7] = 1) > 0$, переходимо до кроку 4.

Крок 4. $i = i + 1; i = 7 + 1 = 8; 8 > 7 (i > PKBC)$, переходимо до кроку 5.

Крок 5. $n = n + 1; n = 3 + 1 = 4$; оскільки $4 = 4$ ФЛФ записуємо символ “)””; $n \leq PKBC$, записуємо знак “+”; переходимо до кроку 2.
 ФЛФ: $(V7) + (V5 \times V6) + (V3 \times V4) +$

Крок 2. $i = 1; j = 2$; у ФЛФ записуємо символ “(”); переходимо до кроку 3.
 ФЛФ: $(V7) + (V5 \times V6) + (V3 \times V4) + ($

Крок 3. Беремо значення 1-го компонента вектора стану 4-го МП. $BC4 [1] = 0$, у ФЛФ записуємо значення вектора $i (V1); j = j - 1; j = 2 - 1 = 1$; оскільки $j > 0$, то записуємо знак множення “ \times ”; далі потрібно перейти до кроку 4.
 ФЛФ: $(V7) + (V5 \times V6) + (V3 \times V4) + (V1 \times$

Крок 4. $i = i + 1; i = 1 + 1 = 2; 2 < 7 (i \leq PKBC)$, переходимо до кроку 3.

Крок 3. Беремо значення 2-го компонента вектора стану 4-го МП. $BC4 [2] = 0$; у ФЛФ записуємо значення вектора V_i ;
 ФЛФ: $(V7) + (V5 \times V6) + (V3 \times V4) + (V1 \times V2$

$j = j - 1; j = 1 - 1 = 0$; оскільки $j = 0$, то переходимо до кроку 4.

Крок 4. $i = i + 1; i = 2 + 1 = 3; 3 < 7 (i \leq PKBC)$, переходимо до кроку 3.

Крок 3. Беремо значення 3-го компонента вектора стану 4-го МП. $(BC4 [3] = 1) > 0$, переходимо до кроку 4.

Крок 4. $i = i + 1; i = 3 + 1 = 4; 4 < 7 (i \leq PKBC)$, переходимо до кроку 3.

Крок 3. Беремо значення 4-го компонента вектора стану 4-го МП. $(BC4 [4] = 1) > 0$, переходимо до кроку 4.

Крок 4. $i = i + 1; i = 4 + 1 = 5; 5 < 7 (i \leq PKBC)$, переходимо до кроку 3.

Крок 3. Беремо значення 5-го компонента вектора стану 4-го МП. $(BC4 [5] = 1) > 0$, переходимо до кроку 4.

Крок 4. $i = i + 1; i = 5 + 1 = 6; 6 < 7 (i \leq PKBC)$, переходимо до кроку 3.

Крок 3. Беремо значення 6-го компонента вектора стану 4-го МП. ($BC4 [6] = 1$) > 0 , переходимо до кроку 4.

Крок 4. $i = i + 1$; $i = 6 + 1 = 7$; $7 = 7$ ($i \leq PKBC$), переходимо до кроку 3.

Крок 3. Беремо значення 7-го компонента вектора стану 4-го МП. ($BC4 [7] = 1$) > 0 , переходимо до кроку 4.

Крок 4. $i = i + 1$; $i = 7 + 1 = 8$; $8 > 7$ ($i > PKBC$), переходимо до кроку 5.

Крок 5. $n = n + 1$; $n = 4 + 1 = 5$; оскільки $5 > 4$, у ФЛФ записуємо знак “)”, $n > PKBC$, то у ФЛФ записуємо знак “;”.

ФЛФ: $(V7) + (V5 \times V6) + (V3 \times V4) + (V1 \times V2)$;

Процедуру запису логічного подання дерева відмов закінчено.

Згідно з алгоритмом графічної побудови графічного ДВ на основі сформованої ФЛФ було отримано дерево, що зображено на рис. 3.

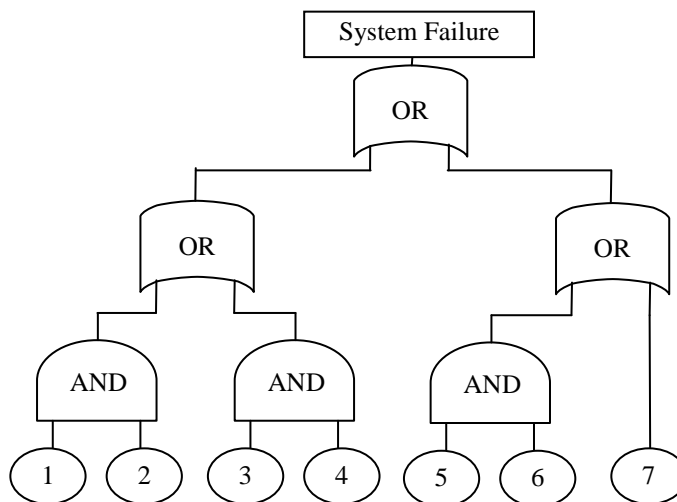


Рис. 3. Дерево відмов відмовостійкої програмно-апаратної системи

Висновки з даного дослідження і перспективи подальших розвідок у даному напрямку. Автоматизована побудова дерева відмов складної технічної системи відбувається завдяки програмному засобу, розробленому на основі запропонованої методики. Використання програмного засобу зменшує затрати часу для аналізу надійності та безпечності експлуатації системи. Особливо відчутне це зменшення під час розгляду різних варіантів забезпечення надійності та безпечності експлуатації складної технічної системи. Це зумовлено тим, що кожен варіант реалізації складної технічної системи потребує побудови свого дерева відмов.

Запропонована методика побудови та аналізу дерева відмов забезпечує високу достовірність оцінок рівня безпечності експлуатації складної технічної системи.

Список використаних джерел:

1. Резчиков А. Ф. Безопасность критических инфраструктур: математические и инженерные методы анализа и обеспечения / Резчиков А. Ф. ; под ред. В. С. Харченко. – Харьков : Нац. аэрокосмический университет им. Н. Е. Жуковского “ХАИ”, 2011. – 641 с.
2. Применение компьютерных технологий для оценивания надежности и безопасности программно-технических комплексов / [Соколов Ю. Н., Харченко В. С., Илюшко В. М.

и др.] ; под ред. Ю. Н. Соколова, В. С. Харченко. – Х. : Нац. аэрокосмический университет им. Н. Е. Жуковского “ХАИ”, 2013. – 458 с.

3. Можаяв А. С. Теоретические основы, опыт применения и направления развития общего логико-вероятностного метода и программного комплекса “Арбитр” моделирования надежности, живучести, безопасности и риска систем / А. С. Можаяв // *Промышленная безопасность* : 17-й научный семинар НТЦ. – М. – 23 ноября 2009. – 11 с.

4. Щербовських С. В. Математичні моделі та методи для визначення характеристик надійності багатотермінальних систем із урахуванням перерозподілу навантаження : монографія / С. В. Щербовських. – Львів : НУ “Львівська політехніка”, 2012. – 296 с.

5. Henley E. J. Reliability engineering and risk assessment / E. J. Henley, H. Kumamoto // Prentice-Hall. – 1981. – 568 p.

6. Kececioglu D. Reliability Engineering Handbook / D. Kececioglu // Prentice-Hall. – 1991. – Vol. 2. – 541 p.

7. Liggesmeyer P. Fault Tree analysis, Current Research Issues, Tutorial / P. Liggesmeyer, B. Kaiser // SAFECOMP 2004, Potsdam 2004.

8. Dynamic Fault Trees Resolution: A Conscious Trade-Off Between Analytical And Simulative Approaches / F. Chiacchio, L. Compagno, D. D’Urso and oth. // *Reliability Engineering & System Safety*. – 2011. – Vol. 96. – № 11. – P. 1515–1526.

9. Skrobanek P. Analysis Of Timing Requirements For Intrusion Detection And Prevention Using Fault Tree With Time Dependencies / P. Skrobanek, M. Woda // *Intrusion Detection Systems* / Edited by P. Skrobanek. – Saneza : InTech, 2011. – P. 307–324.

10. Contini S. Analysis Of Large Fault Trees Based On Functional Decomposition / S. Contini, V. Matuzas // *Reliability Engineering & System Safety*. – 2011. – Vol. 96. – № 3. – P. 383–390.

11. Probabilistic Algebraic Analysis of Fault Trees With Priority Dynamic Gates and Repeated Events / G. Merle, J.-M. Roussel, J.-J. Lesage, A. Bobbio // *Reliability, IEEE Trans. on*. – 2010. – Vol. 59. – № 1. – P. 250–261.

12. Lixuan L. Joint failure importance for noncoherent fault trees / L. Lixuan, J. Jiang // *Reliability, IEEE Trans. on*. – 2007. – Vol. 56. – № 3. – P. 435–443.

13. Rodrigo de Queiroz Souza FMEA and FTA Analysis For Application of The Reliability Centered Maintenance Methodology: Case Study on Hydraulic Turbines / Rodrigo de Queiroz Souza, Alberto José Álvares // *ABCM Symposium Series in Mechatronics*. – Vol. 3. – P. 803–812.

14. Маевский Л. С. Методы обеспечения надежности информационно-телекоммуникационных систем на различных этапах жизненного цикла / Л. С. Маевский. – СПб. : Издатель Барзилович З. П., 1999. – 112 с.

15. Mandziy B. Mathematical model for failure cause analysis of electrical systems with load-sharing redundancy of component / B. Mandziy, O. Lozynsky, S. Shcherbovskykh // *Przeegląd Elektrotechniczny*, 89 (2013). – № 11. – P. 244–247.

16. Волочій Б. Ю. Методика розрахунку мінімальних січень для відмовостійких систем на основі структурно-автоматної моделі / Б. Ю. Волочій, Л. Д. Озірковський, А. В. Машак та ін. // *Вісник НТУУ “КПІ”. Серія: “Радіотехніка. Радіоапаратобудування”*. – 2013. – № 52. – С. 38–45.

17. Тараканов К. В. Аналитические методы исследования систем / Тараканов К. В., Овчаров Л. А., Тырышкин А. Н. – М. : Сов. радио, 1974. – 240 с.

18. Волочій Б. Ю. Технологія моделювання алгоритмів поведінки інформаційних систем / Б. Ю. Волочій. – Львів : Національний університет “Львівська політехніка”, 2004. – 220 с.

19. Математичні моделі та методи аналізу надійності радіоелектронних, електротехнічних та програмних систем / Бобало Ю. Я., Волочій Б. Ю., Лозинський О. Ю. та ін. – Львів : Національний університет “Львівська політехніка”, 2013. – 300 с.