

Винахід відноситься до галузі автоматики та обчислювальної техніки і призначений для моделювання характеристик надійності програмного забезпечення відмовостійких обчислювальних систем.

Відомий пристрій для моделювання імовірнісного графа, що містить перший та другий регістри, перший, другий та третій постійні запам'ятовуючі пристрої, формувач, першу, другу та третю схеми порівняння, перший та другий генератори випадкових сигналів, першу та другу схеми збігу, перший та другий елементи АБО, лічильник об'єму моделювання, реверсивний лічильник, перший та другий накопичуючі суматори, перший та другий блоки задання константи, перший та другий блоки ділення та індикатор [1]. Цей пристрій дозволяє моделювати статистичні характеристики процесу автоматичного відновлення за допомогою імовірнісного графа. Недоліком пристрою є те, що він не дозволяє визначити характеристики надійності програмних засобів відмовостійких обчислювальних систем.

Відомий пристрій для моделювання оцінки надійності програмних засобів, що містить регістр, генератор випадкових сигналів, перший, другий, третій та четвертий постійні запам'ятовуючі пристрої, першу та другу схеми порівняння, формувач, першу та другу схеми збігу, елемент АБО, блок задання константи, накопичуючий суматор, блок ділення, лічильник та індикатор [2]. Цей пристрій дозволяє моделювати статистичні характеристики надійності програмних засобів. Недоліком пристрою є те, що елементарні показники надійності кожної вершини графа є сталими величинами, що зменшує достовірність результатів моделювання, що отримуються.

В основу винаходу поставлена задача вдосконалення пристрою для моделювання статистичних характеристик надійності програмного забезпечення відмовостійкої обчислювальної системи за допомогою імовірнісного графа, в якому за рахунок моделювання випадкових значень елементарних показників надійності і багаторазового проходження різних маршрутів графа підвищується достовірність оцінки середньої імовірності безвідмовного виконання програми.

Поставлена задача вирішується тим, що до пристрою, який містить регістр, перший генератор випадкових сигналів, перший, другий, третій та четвертий запам'ятовуючі пристрої, першу та другу схеми порівняння, формувач, першу та другу схеми збігу, елемент АБО, блок задання константи, накопичуючий суматор, блок ділення, лічильник та індикатор, причому вихід першого запам'ятовуючого пристрою (ПЗП) підключений до першого входу першої схеми порівняння, другий вхід якої з'єднаний з виходом першого генератора випадкових сигналів, а перший вихід з'єднаний з першим входом першої схеми збігу, другий вхід якої підключено до виходу другого ПЗП, другий вихід першої схеми порівняння підключено до другого входу другої схеми збігу, перший вхід якої з'єднаний з виходом третього ПЗП, адресний вхід якого з'єднаний з відповідними адресними входами першого, другого та четвертого ПЗП, із входом формувача і підключений до виходу першого регістра, інформаційний вхід якого з'єднаний з першим входом другої схеми порівняння і з виходом елемента АБО, перший і другий входи якого підключені до виходів відповідно першої та другої схеми збігу, вхід запуску першого регістра є входом пристрою, а вхід скидання першого регістра з'єднаний із входом скидання лічильника і підключений до виходу другої схеми порівняння, другий вхід якої підключено до виходу блока задання константи, вихід лічильника підключений до першого входу блока ділення, вихід якого з'єднаний із входом індикатора, а другий вхід підключено до виходу накопичуючого суматора, у відповідності з винаходом додатково введені другий генератор випадкових сигналів, блок масштабування, суматор і блок множення, вхід другого генератора випадкових сигналів з'єднаний з першим входом суматора і входом блока масштабування і підключений до виходу четвертого ПЗП, а вихід підключений до першого входу блока множення, другий вхід якого з'єднаний з виходом блока масштабування, а вихід підключений до другого входу суматора, вихід якого з'єднаний із входом накопичуючого суматора.

На фіг.1 показано блок-схему пристрою, а на фіг.2 наведено приклад імовірнісного графа програми.

Вихід першого регістра 1 з'єднаний з адресними входами відповідно першого 2, другого 3, третього 4, четвертого 5, ПЗП і з входом формувача 6. Вихід першого ПЗП 2 з'єднаний з першим входом першої схеми порівняння 7, другий вхід якої підключено до виходу першого генератора випадкових сигналів 8, вхід якого підключений до виходу формувача 6. Перший і другий входи першої схеми порівняння 7 підключені відповідно до першого і другого входів першої 9 і другої 10 схем збігу. Другий вхід першої схеми збігу 9 підключений до виходу другого ПЗП 3, а перший вхід другої схеми збігу 10 підключений до виходу третього ПЗП 4. Виходи першої 9 і другої 10 схем збігу підключені відповідно до першого і другого входів елемента АБО 11, вихід якого підключений до першого входу другої схеми порівняння 12 і інформаційного входу першого регістра 1, вхід скидання якого з'єднаний із входом лічильника 13. Вхід запуску першого регістра 1 є входом пристрою. Вихід другого генератора 14 випадкових сигналів підключений до першого входу блока множення 15, вихід якого з'єднаний з першим входом суматора 16, а другий вхід з'єднаний із виходом блока масштабування 17, вхід якого з'єднаний із входом генератора 14 випадкових сигналів і першим входом суматора 16 і підключений до виходу четвертого ПЗП 5, вихід суматора 16 підключений до входу накопичуючого суматора 18. Вихід накопичуючого суматора 18 з'єднаний із входом блока 19, другий вхід якого підключений до виходу лічильника 13, а вихід з'єднаний із входом індикатора 20. Вихід блока 21 задання константи з'єднаний із другим входом другої схеми порівняння 12.

Принцип роботи пристрою такий.

Програмний модуль з програмного забезпечення відмовостійкої обчислювальної системи може бути поданий графовою моделлю програми. При моделюванні обчислювального процесу за такою моделлю передбачається надання кожній вершині графа деякого значення, наприклад, елементарного показника d_i , що дорівнює логарифму ймовірності правильної роботи операторів, пов'язаних з даною вершиною. Динаміка функціонування програми визначається вибором визначеного маршруту на графі. Цей вибір обумовлюється сукупністю реалізацій передач керування в логічних вершинах, які зв'язані з випадковим процесом надходження на вхід програми різних векторів вхідних даних, що приводить до випадкового вибору маршрутів на графі. Вибір маршрутів залежить від значення транзитивних ймовірностей переходів P_{ij} від i -ї до j -ї вершини графа, якими навантажуються дуги графа.

Очевидно, що окремий маршрут L реалізується з деякою ймовірністю

$$P(L) = \prod_{i \in L} P_{ij},$$

а середня оцінка ймовірності відмови може бути визначена як

$$Q = \sum_L \left\{ P(L) \times \sum_{i \in L} d_i \right\},$$

тому що параметр надійності Q є дискретною випадковою величиною і його середнє значення визначається на множині реалізацій як середнє по ймовірності. Але для багаторозгалужених графових моделей, які мають багато логічних вершин (вершин з двома виходами), кількість можливих маршрутів різко зростає. Наприклад, якщо граф має 20 логічних вершин, то при певних умовах кількість маршрутів досягає $2^{20} \approx 10^6$. В таких випадках оцінку надійності функціонування програми можна одержати тільки за допомогою імітаційного моделювання. Кількість реалізацій маршрутів визначає об'єм моделювання M . Тоді

$$Q \cong \frac{\sum_{i \in L} d_i}{M},$$

Ма внесок маршрутів в оцінку Q буде пропорційним ймовірності їх реалізацій.

Однак значення цих показників d_i отримують за результатами статистичних даних, які, як правило, не повні як за обсягом, так і за кількістю враховуваних факторів. Наприклад, елементарним показником може бути величина $d_i = -\ln r_i$,

де r_i - ймовірність безвідмовного виконання послідовності операторів, що асоціюються з i -ю вершиною, яка може бути надана фірмами - розробниками програмного забезпечення.

Звичайно, використовують статистичні дані, які накопичені під час відлагодження та експлуатації подібних програмних засобів, і отримують деяку величину q , як оцінку можливої кількості помилок в програмі, нормовану на одну команду (оператор). Тоді

$$d_i = m_i q,$$

де m_i - кількість команд, пов'язаних з i -ю вершиною.

Тому, для підвищення достовірності результатів моделювання оцінки надійності програмного забезпечення, елементарному показникові d_i пропонується надати розкид, наприклад, за нормальним законом з математичним сподіванням, яке дорівнює d_i , і дисперсією $0,1 d_i^2$, що еквівалентно моделюванню шуму реального процесу функціонування алгоритму програми.

Отримана під час такого моделювання середня оцінка надійності буде більш достовірною. Закон розподілу та його параметри можуть бути узгоджені з представниками фірм - розробників програмного забезпечення.

На фіг.2 показані один з можливих варіантів ймовірнісного графа програми.

Блоки 1-12, що показані на фіг.1, визначають ту чи іншу реалізацію маршруту на ймовірнісному графі.

Перший генератор випадкових сигналів 8 формує реалізацію неперервної випадкової величини, яка має рівномірний розподіл на інтервалі $[0, 1]$.

В першому ПЗП зберігаються значення ймовірностей переходів у відповідності з послідовними номерами вершин графа. Наприклад, у першій комірці, яка відповідає вершині Z_1 (номер вершини - 1) записане число, яке дорівнює найменшому значенню ймовірності переходу (P_{12} чи P_{13}), у другій комірці ПЗП 2 записане значення P_{24} , або P_{25} . У четвертій комірці (відповідає четвертій вершині) записується одиниця і т.д.

У другому ПЗП 3 записується найменший номер вершини графа, яка пов'язана з поточною вершиною, а у третьому ПЗП 4 записується номер другої вершини, що зав'язана з поточною (при відсутності такої вершини записується 0). Нехай, наприклад, на виході регістра 1 встановлено код поточної вершини Z_5 (п'ять), а $P_{56} < P_{57}$. Тоді на виході першого ПЗП 2 буде значення P_{56} , на виході другого ПЗП 3 - номер 6, а на виході третього ПЗП 4 - номер 7, тобто номери вершин, які пов'язані з поточною вершиною під номером 5.

У четвертому ПЗП 5 записується значення показників вершин d_i ймовірнісного графа.

Таким чином, в чотирьох ПЗП кодується конкретна реалізація ймовірнісного графа.

Другий генератор 14 формує нормальну випадкову величину з параметрами $m=0$, $\sigma^2=1$, тобто $N(0; 1)$. Блоки 14-16 формують нормальну випадкову величину $N(d_i; 0,1 d_i^2)$.

Пристрій працює таким чином.

При надходженні сигналу запуску на вхід запуску першого регістра 1 на його виході встановлюється код першої вершини графа Z_1 (одиниця), який надходячи на адресні входи усіх ПЗП встановлює на виходах відповідно значення ймовірності P_{12} (при умові, що $P_{12} < P_{13}$), код числа 2 (ПЗП 3), код числа 3 (ПЗП 4), а також значення d_1 для першої вершини (ПЗП 5). Крім того, сигнал з виходу регістра 1 за допомогою формувача 6 запускає генератор випадкових сигналів 8, на виході якого одержується випадкове число r_i з інтервалу $[0, 1]$.

Якщо $r_i \leq P_{12}$, то це значить, що у графі реалізувалась дуга, що відповідає ймовірності переходу P_{12} . Тоді для випадку $r_i \leq P_{12}$ на першому виході першої схеми порівняння 7 з'являється сигнал, який дозволяє проходження коду номера другої вершини через першу схему збігу 9. Сигнал на другому виході схеми порівняння блокує другу схему збігу 10. На випадок $r_i > P_{12}$ активізується другий вихід схеми порівняння 7.

Крім того, значення показника надійності d_1 першої вершини надходить до входів генератора 14 випадкових сигналів і блока масштабування 17 та до першого входу суматора 16. Генератор 14 формує випадкове число, яке

множиться у блоці множення 15 на число $10^{-\frac{1}{2}} d_i$ з виходу блока масштабування 17, так що на виході суматора 16

утворюється число $d_i + 10^{-\frac{1}{2}} d_i \times r_{12}$, r_{12} - випадкове число, що формує генератор 14. Сума з виходу суматора 16 підходить до інформаційного входу накопичуючого суматора 18.

Далі код з виходу першої схеми збігу 9 через елемент АБО 11 (на виході елемента АБО утворюється код поточної вершини графа при проходженні маршруту) надходить до інформаційного входу першого регістра 1 і далі подається на адресні входи усіх ПЗП. На виході першого ПЗП 2 встановлюється код P_{24} (якщо $P_{24} < P_{25}$), на виходах другого та третього ПЗП - код вершин Z_4 і Z_5 .

Якщо $t_i > P_{34}$, то код вершини Z_5 передається на вихід елемента АБО 11 і це вершина стає поточною тощо. Після проходження маршруту на виході елемента АБО 11 встановлюється код останньої вершини Z_{14} , а на виході суматора 18 утворюється код, відповідний сумарному показнику надійності реалізованого маршруту.

Крім того поточний номер вершини на виході елемента АБО 11 надходить до першого входу другої схеми порівняння 12, де порівнюється з кодом останньої вершини Z_{14} , який надходить з виходу блока 21 задання константи. Якщо коди співпадають, то на виході схеми порівняння 12 утворюється сигнал, який додає одиницю у лічильнику 13. Крім того, цей сигнал встановлює перший регістр 1 в початковий стан, що відповідає номеру першої вершини Z_1 . Далі процес повторюється.

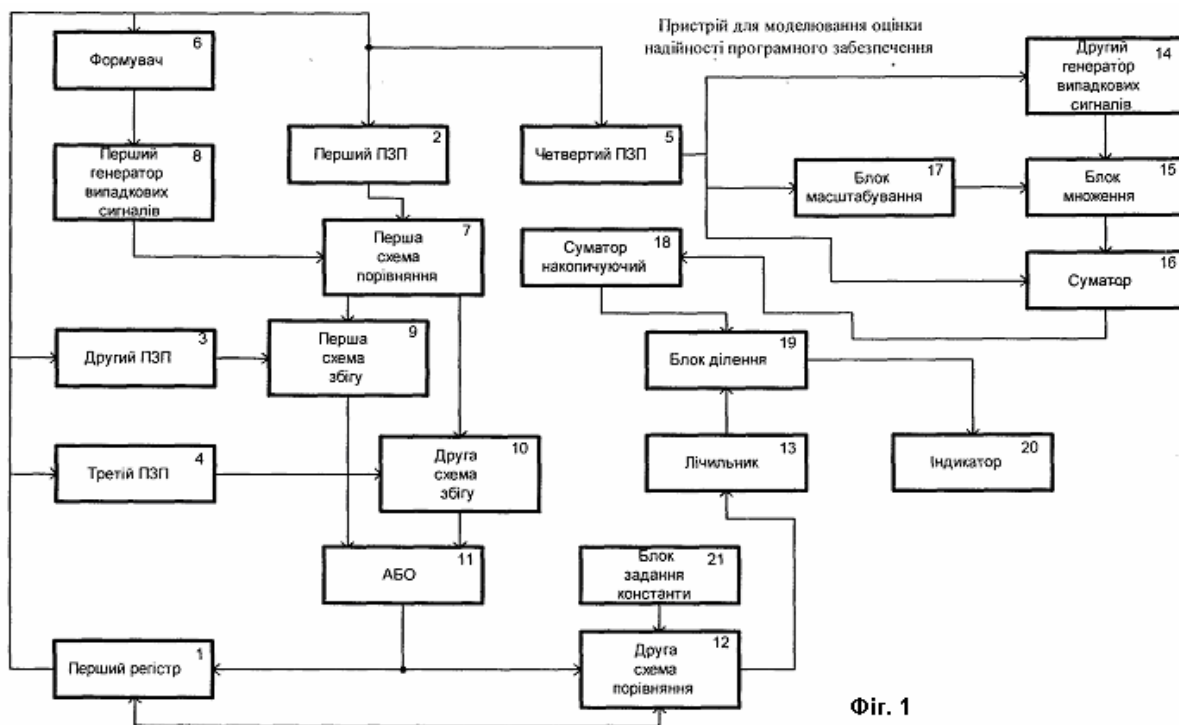
Таким чином на виході накопичуючого суматора 18 утворюється поточна сума показників надійності для вершин реалізованих маршрутів, так що на виході блока ділення 19 утворюється поточна оцінка середньої імовірності відмови програми. Це значення відображається на індикаторі 20.

З метою спрощення схеми та її опису в блок-схемі відсутні елементи затримки (у вигляді ланцюгів елементів І), які необхідні для забезпечення стійкої роботи пристрою.

Пристрій може бути реалізований у цифровому варіанті на стандартних мікросхемах.

Література:

1. Патент України 34114А, G06G7/48, Пристрій для моделювання імовірнісного графа (аналог).
2. Патент України 36399А, G06G7/48, Пристрій для моделювання оцінки надійності програмних засобів (прототип)



Фіг. 1

