4 **Кабецкий, А. Г.** Методы и инструментальные средства построения логических устройств электрической централизации на базе программируемых логических интегральных схем / А. Г. Кабецкий, Д. С. Марков // Известия Петербургского университета путей сообщения. – 2010. – № 2. – С. 168–173.

5 **Dobiáš, R.** FPGA Based Design of Railway's Interlocking Equipment / R. Dobiáš, H. Kubátová // Proceedings of EUROMICRO Symposium on Digital System Design. – 2004. – P. 467–473.

6 **Сапожников, Вл. В.** Синтез систем управления движением поездов на железнодорожных станциях с исключением опасных отказов / Вл. В. Сапожников**. –** М. : Наука, 2021. – 229 с.

7 **Сапожников, В. В.** Коды с суммированием для систем технического диагностирования. Т. 1. Классические коды Бергера и их модификации : [монография] / В. В. Сапожников, Вл. В. Сапожников, Д. В. Ефанов. – М. : Наука, 2020. – 383 с.

8 **Сапожников, В. В.** Коды с суммированием для систем технического диагностирования. Т. 2. Взвешенные коды с суммированием : [монография] / В. В. Сапожников, Вл. В. Сапожников, Д. В. Ефанов. – М. : Наука, 2021. – 455 с.

УДК 656.25

ОЦЕНКА НЕЗАВИСИМОСТИ ОТКАЗОВ В ДИВЕРСИТЕТНОМ ПРОГРАММНОМ ОБЕСПЕЧЕНИИ С ПОМОЩЬЮ EL-МОДЕЛИ

С. Н. ХАРЛАП, А. Ю. КУЛАЖЕНКО
Белорусский государственный университет транспорта, г. Гомель

Современные микроэлектронные системы железнодорожной автоматики реализуются как многоканальные системы. В таких системах технологические алгоритмы, связанные с функциональной безопасностью, выполняются параллельно в различных каналах с последующим сравнением результатов работы. Управляющие воздействия вырабатываются только в том случае, если каналы выдали одинаковые результаты. Безопасность таких систем базируется на предположении, что любой отказ или ошибка в программном обеспечении приведут к различной реакции каналов на одни и те же входные воздействия, что будет обнаружено схемой сравнения и переведет систему в защитное состояние.

Такое поведение системы при отказах или ошибках в программном обеспечении можно обеспечить только в случае независимости возникновения отказов и проявления ошибок программного обеспечения в разных каналах. В соответствии с МЭК61508 основным способом обеспечения независимости отказов в различных каналах многоканальной системы является диверситет (разнообразие). При этом диверситет может быть функциональным, основанным на различных способах решения одной и той же задачи, и диверситетом технологий.

К сожалению, в стандарте не даются методы оценки достигнутого диверситета, что приводит к убежденности, что любой диверситет обеспечивает необходимую независимость отказов. В то же время имеются публикации, показывающие ложность данного утверждения, в частности предложенная в конце прошлого века формализованная модель (EL-модель) позволяет получить вероятностную оценку диверситетных программ и математически доказывает, что невозможно достичь независимости отказов в различных версиях программы, даже если версии разработаны независимо в соответствии с действующими стандартами.

Рассмотрим в качестве примера оценку независимости отказов управляющей программы, выполняющей вычисление функции алгебры логики $Y=A\cdot(B+\bar{C}+D\cdot A\cdot\bar{D})+D\cdot\bar{A}\cdot(B+\bar{B})$. Функция реализована следующими способами:

π_1 – непосредственное вычисление ФАЛ;

π_2 – минимизация (преобразование) ФАЛ и ее непосредственное вычисление. Функция после минимизации: $Y=A\cdot(B+\bar{C})+D\cdot\bar{A}$;

π_3 – вычисление методом бинарных программ;

π_4 – вычисление по таблице истинности (метод адресных переходов).

Методы π_1 и π_2 реализованы непосредственным вычислением ФАЛ с помощью логических операций AND, OR, NOT. Метод π_3 представляет собой вычисление ФАЛ с помощью проверки ряда условий, например, если $A = 0$ и $D = 0$, то $Y = 0$. Метод π_4 основан на предварительном вычислении таблицы истинности функции и записи ее в определенную область памяти. Алгоритм предусматривает вычисление по значениям входных переменных адреса ячейки таблицы истинности, в которой хранится результат. Методы π_1, π_3 и π_4 считаются взаимно диверситетными. Для методов π_1 и π_2 взаимный диверситет считается недостаточным.

Для оценки корректности программ применяется метод функционального тестирования по критериям:

1) проверка всех классов входных данных, когда тест должен содержать по одному представителю из каждого класса;

2) проверка всех классов выходных данных, когда при исполнении тестовых примеров должно быть получено по одному представителю из каждого класса.

Тестовая последовательность (*ABCD*) представлена в таблице 1.

Таблица 1 – **Тестовая последовательность**

№	Входные значения				Ожидаемый результат
	A	B	C	D	Y
T1	**0**	**1**	**0**	**1**	**1**
T2	**1**	**0**	**1**	**0**	**0**

Предположим, что при разработке программ были допущены ошибки. Статистическая вероятность возникновения ошибок в программном обеспечении составляет примерно от 2 до 15 ошибок на 1000 строк кода. Для реализации указанных версий ПО объем кода составляет не более 50 строк, поэтому можно предположить, что вероятность ошибки составит не менее 0,1. Типы ошибок определены экспертно, основываясь на том, что основными причинами ошибок являются невнимательность человека при создании программы и использование неисправных аппаратных средств или некорректно работающих программных инструментальных средств. Рассмотрены следующие типы ошибок:

1) невыполнение (пропуск) инверсии логической переменной;

2) преобразование одной логической операции в другую (например AND в OR и обратно);

3) неверное значение флага сравнения результатов (при выполнении операций сравнения в методе бинарных программ);

4) невыполнение (неполное выполнение) сдвига при формировании адреса в методе адресных переходов.

При оценке версий ПО с помощью EL-модели учитывались только версии программ, наличие ошибок в которых не обнаруживалось тестами.

Для указанных условий были рассчитаны вероятности формирования программами ошибочных результатов. В соответствии с методикой были получены следующие результаты.

Вероятность того, что случайно выбранная программа для случайно выбранного входного значения отработает ошибочно, составляет $Q_1 = 0{,}013$.

Если использовать две случайно выбранные версии с последующим сравнением результатов, то вероятность ошибочного выходного воздействия для случайно выбранного входного значения составляет $Q_2 = 0{,}0006$.

Если же использовать предположение о полной независимости отказов для двух каналов, то это даст более низкую вероятность отказа системы $Q_3 = 0{,}0002$.

Таким образом, результаты исследований подтвердили тезис о том, что полная независимость отказов даже для диверситетных версий ПО не может быть достигнута, т. е. оценку безопасности ПО невозможно получить просто из факта наличия диверситета: нельзя утверждать, что если одна система имеет вероятность опасного отказа λ, то диверситетная система будет иметь вероятность опасного отказа равную λ^2.

УДК 656.25

УРОВНИ ФОРМАЛИЗАЦИИ ФУНКЦИИ БЕЗОПАСНОСТИ ПРИ ВЕРИФИКАЦИИ МИКРОПРОЦЕССОРНЫХ СИСТЕМ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ ДВИЖЕНИЯ ПОЕЗДОВ

С. Н. ХАРЛАП, Б. В. СИВКО

Белорусский государственный университет транспорта, г. Гомель

Микропроцессорные системы железнодорожной автоматики и телемеханики (СЖАТ) используются на железнодорожном транспорте, где главным аспектом является безопасность движения поездов. К таким системам предъявляются повышенные требования по безопасности, надёжности и