АВТОМАТИКА, ТЕЛЕМЕХАНИКА И СВЯЗЬ

УДК 004.312.466

Б. В. СИВКО, ассистент, Белорусский государственный университет транспорта, г. Гомель

ДОКАЗАТЕЛЬСТВО КОРРЕКТНОСТИ БЛОКА ТЕЛЕУПРАВЛЕНИЯ 16-1 ДИСПЕТЧЕРСКОЙ ЦЕНТРАЛИЗАЦИИ «НЁМАН»

Приведены результаты верификации блока телеуправления 16-1 диспетчерской централизации «Неман», которая выполнялась с помощью доказательства корректности. Рассмотрены ключевые шаги и особенности проводимой верификации. Показано, что данным способом можно находить ошибки в программном обеспечении, которые обнаружить иначе крайне затруднительно.

Одной из серьезных проблем построения безопасных микроэлектронных систем железнодорожной автоматики и телемеханики (СЖАТ) является организация сопряжения с исполнительными объектами, которые имеют разнообразные характеристики, что затрудняет их унификацию и стандартизацию. Поэтому для решения проблемы чаще используется способ, заключающийся в создании специализированных функциональных блоков, к которым предъявляются специфичные требования безопасности, не позволяющие использовать выпускаемые промышленностью устройства сопряжения с объектами (УСО) [1].

В связи с этим разработка и эксплуатация каждого из вышеназванных устройств является уникальным процессом, при котором результат должен соответствовать как поставленным требованиям с точки зрения технического задания, так и требованиям безопасности функционирования, предъявляемым микроэлектронным СЖАТ.

Примерами таких устройств являются блоки телеуправления (ТУ16-1) и телесигнализации (ТС32-1) диспетчерской централизации (ДЦ) «Нёман», безопасные блоки управления напольными объектами ТУ-8Б и ТС-16Б процессорно-релейной централизации «Ипуть».

В лаборатории БЭМС ТС БелГУТа (далее лаборатория) проводятся испытания устройств данного типа на безопасность их функционирования, во время которых анализируется наличие ошибок в программном обеспечении (ПО), при этом одним из способов поиска является доказательство корректности, применяемое как инструмент верификации готового ПО, когда требуется наличие соответствия имеющейся спецификации программы и её реализации.

В процессе верификации потенциально могут быть обнаружены ошибки, не выявленные ранее. Кроме того, при проведении доказательства корректности могут быть найдены ошибки этапа валидации, предложены рекомендации к исправлениям, определены характеристики работы ПО, а полученные результаты – использованы для других задач, таких как повышение качества разработки новых устройств, более точная формализация верификации, согласование протоколов взаимодействия подсистем аппаратно-программного комплекса (АПК).

Одним из испытываемых устройств в лаборатории являлся блок ТУ16-1 ДЦ «Нёман», для которого была проведена верификация с помощью доказательства корректности [2, 3]. Блок ТУ16-1 входит в состав СЖАТ ДЦ «Нёман» и является микроэлектронным УСО, осуществляющим получение информации от управляющей ЭВМ и установку соответствующих выходных воздействий [4]. При приеме данных происходит получение посылок по последовательному интерфейсу, согласно которым изменяется каждый из 16 выходных ключей, которые могут принимать одно из двух состояний – 1 или 0 (рисунок 1).

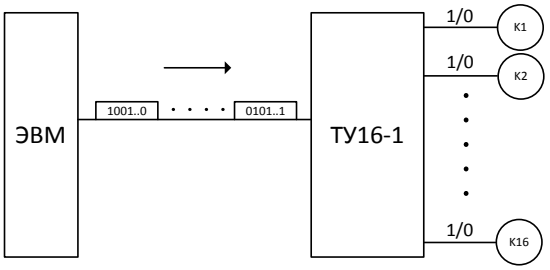

Рисунок 1 – Схема работы блока ТУ16-1

ПО блока ТУ16-1 состоит из 200 строк ассемблера процессора PIC16C57, который не обладает большой сложностью и функционально выполняет формирование устойчивых управляющих воздействий, обеспечивает светодиодную индикацию состояния и проверку замыканий выходных контактов портов [3].

Так как рассматриваемое ПО не обладает большой сложностью, то это позволяет использовать доказательства корректности без применения каких-либо дополнительных средств или методик. Поэтому верификация данным методом проводилась вручную с использованием модели логики Хоара [5].

Порядок и содержимое сообщений, получаемых блоком, задает управляющая ЭВМ, в которых содержится информация для установки четырёх ключей и идентификатор одной из тетрад [4]. При приеме сообщения ПО проверяет его корректность (по длине и биту четности), и если все в порядке, то обновляются соответствующие управляющие ключи.

При синтезе безопасных схем в железнодорожной автоматике и телемеханике широко применяются логические элементы с несимметричными отказами (h_1 – надежные элементы), для работы с которыми спроекти-

рован блок ТУ16-1 [6]. В связи с этим функцией опасного отказа является функция, равная единице на тех запрещенных наборах, на которых ложное включение управляемого объекта является опасным [7]. Поэтому условия безопасности требуют, чтобы самопроизвольный переход из 0 в 1 на выходных управляющих ключах был недопустим. Кроме того, необходимо, чтобы обновление выходной информации происходило как можно быстрее, что особенно важно тогда, когда ключ находится в состоянии 1, а управляющая ЭВМ передает команду на его перевод в 0. Если время такого перехода из 1 в 0 окажется слишком большое, то это может привести к нарушению условий безопасности работы системы.

С точки зрения рассматриваемой верификации было сформулировано доказываемое условие: отсутствие самопроизвольного перехода из 0 в 1 и удержание выходного значения 1 в случае отсутствия команды от управляющей ЭВМ только в течение определенного времени.

Передача команд представляет собой множество событий в виде отправки сообщения от ЭВМ и его получения в блоке ТУ16-1. После успешного приема обновляется одна из тетрад, и, если был принят 0 на какой-либо ключ, то его выходное состояние должно обязательно перейти в 0 не позже, чем через некоторый безопасный интервал времени T, который определяется исходя из особенностей системы и предъявляемых требований безопасности. Например, может быть задано, что значение T должно быть меньше времени такта выполнения алгоритма управляющей ЭВМ или быть намного меньше времени реакции используемого выходного реле.

Для описания условий безопасности введем следующие определения:

– $E_{вх(i)}$ – i-е событие получения входного ключа при посылке-приеме от управляющей ЭВМ;

– $t(E_{вх(i)})$ – момент времени события при приеме значения от управляющей ЭВМ (в секундах);

– $K(E_{вх(i)})$ – значение ключа при приеме значения от управляющей ЭВМ (0 или 1);

– $K_{вых}(t)$ – значение в момент времени t выходного ключа, передаваемого на объект управления;

– T – значение минимального безопасного времени переключения состояния выходного ключа;

– $E_{вх(0\to1)}$ – событие перехода входного значения из 0 в 1, то есть такое $E_{вх(i)}$, при котором выполняется условие (1):

$$(E_{вх(i-1)} = 0)\ and\ (E_{вх(i)} = 1); \quad (1)$$

– $E_{вх(1\to0)}$ – событие перехода входного значения из 1 в 0, то есть такое $E_{вх(i)}$, при котором выполняется условие (2):

$$(E_{вх(i-1)} = 1)\ and\ (E_{вх(i)} = 0). \quad (2)$$

События $E_{вх}$, $E_{вх(0\to1)}$ и $E_{вх(1\to0)}$ упорядочены таким образом, что выполняются условия (3) и (4):

$$t(E_{вх(i)}) < t(E_{вх(i+1)}), \quad (3)$$

$$t(E_{вх(1\to0)(i)}) < t(E_{вх(0\to1)(i)}). \quad (4)$$

С помощью введенных определений требуемое к выполнению условие безопасности принимает вид (5):

$$(\forall K_{вых}) \in (t(E_{вх(1\to0)(i)} + T) < t(K_{вых}) < t(E_{вых(0\to1)(i)})),$$
$$K_{вых} = 0. \quad (5)$$

Графическое изображение данного условия показано на рисунке 2.

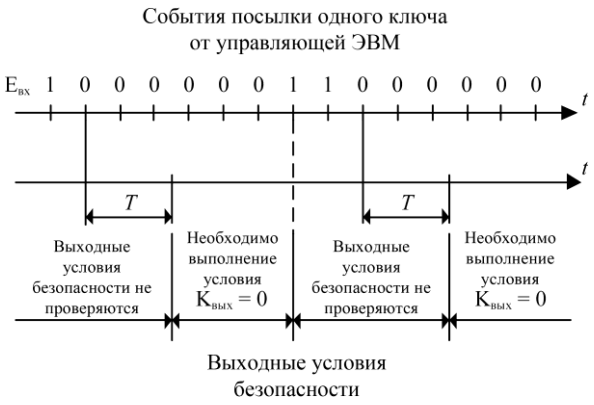

Рисунок 2 – Условия безопасности выходных ключей блока ТУ16-1

Таким образом, в некоторый момент времени управляющая ЭВМ посылает команду перехода состояния выходного ключа из 1 в 0, которую принимает блок ТУ16-1, и изменяет выходное значение, что должно завершиться за заданное фиксированное время T. Если после этого происходит прием команды перехода в 1, то условие безопасности ($K_{вых} = 0$) не обязано выполняться до следующей команды перехода в 0.

Внутренняя логика ПО блока ТУ16-1 такова, что происходит установка выходных ключей в соответствии с полученными на входе значениями параметров, переданных от управляющей ЭВМ. В случае разрыва канала связи, который фиксируется в случае отсутствия корректных посылок от управляющей ЭВМ в течение некоторого фиксированного времени, выходные ключи устанавливаются в нулевое значение.

В начале верификации была получена функция безопасности (5), выполнение которой требовалось доказать. В дальнейшем были определены программные конструкции, которые способны повлиять на условия безопасности. Это позволило установить, как происходит прием команд и как возможно изменение выходных значений ключей. Во время верификации состояние системы было представлено в виде объединения двух множеств: внутренних регистров процессора и внешних условий системы. Переходы между состояниями осуществлялись согласно рассматриваемым внутренним операторам программы в виде взаимосвязанных троек Хоара [5].

Работа внутреннего алгоритма была представлена в виде двух этапов:

1 Старт и выход в рабочий режим.

2 Постоянная работа в рабочем режиме.

Для каждого из этапов верификация проводилась в две фазы: доказательство факта завершения программы (то, что выполнение обязательно дойдет до заданной

вершины графа троек Хоара) и доказательства того, что условия безопасности (5) истинны (верны в рассматриваемых точках).

При проведении верификации первого этапа применялось доказательство с использованием предикатов абстракций (*Predicate Abstraction*) [8]. С помощью данного метода были быстро получены простые графы сценариев запуска для различных внешних условий при известном состоянии во время инициализации микроконтроллера. Так были получены все варианты выхода в рабочий режим, а в последующем полученные условия сравнивались с рассматриваемым на втором этапе инвариантом работы основного цикла.

В рабочем режиме микропроцессор постоянно выполняет один и тот же алгоритм, во время которого проверяется состояние внешних условий, проводится в случае события приема обработка входящей посылки и контролируется индикация. Этот процесс является бесконечным и выход из него возможен только при внутреннем сигнале сброса по таймеру, отсутствие которого доказывалось отдельно.

Во время верификации второго этапа выполнялись следующие действия. С помощью предикатов абстракций, полученных на первом этапе, проводились холостые циклы с возвратом в их начальную точку. Полученное условие было расширено (ослаблено) для определения инварианта цикла I, с которым в дальнейшем было проведено доказательство снизу-вверх. Так было найдено предусловие для предыдущего начала цикла $l_{\text{пред}}$. Выполнение ($l_{\text{пред}} \in l$) являлось доказательством того, что цикл бесконечен.

В дальнейшем было доказано, что после приема входящей посылки значение корректно принимается, проверяется на ошибки и выходные значения ключей устанавливаются согласно полученным данным. Для этого использовался вывод предусловия установки ключа от точки задания выходных значений до точки приема сообщения.

Согласно логике микроконтроллера сброс внутренних ключей при отсутствии сигнала связи определен как

$$t - t_{\text{п}} > T_{\text{предельное}}, \qquad (6)$$

где $t_{\text{п}}$ – время последней корректно принятой посылки; $T_{\text{предельное}}$ – предельное время отсутствия приема корректных посылок.

Полученное с помощью правил вывода условие (6) определено сразу для всех ключей. То есть, если для каждого из них команды отсутствуют, то только тогда выходные управляющие значения блок переводит в безопасное состояние. Но, согласно условию безопасности (5), каждый ключ должен контролироваться отдельно, что формулируется следующим образом (7):

$$t - t_{\text{п}(i)} > T, \qquad (7)$$

где $t_{\text{п}(i)}$ – время последней корректно принятой посылки по i-му ключу.

Из-за отличий между (6) и (7) доказать корректность работы алгоритма блока ТУ16-1 не удалось. Для безопасной работы необходимо, чтобы (7) было верным всегда, когда выполняется (6).

Сравнение причин разности условий (6) и (7) показывает, что реализованная логика блока ТУ16-1 не выполняет условие безопасности (5), которое может быть нарушено тогда, когда блок ТУ16-1 считает, что связь присутствует, но при этом выходные ключи не обновляются. Это возможно тогда, когда происходит прием корректных посылок, но для одной из тетрад они либо не приходят, либо оказываются некорректными.

Описанная ситуация проявляется в достаточно специфичных и редких случаях. Например:

– повышенный уровень помех в канале связи от управляющей ЭВМ к блоку ТУ16-1;

– нарушение частоты приема какой-либо тетрады из-за ошибки ПО управляющей ЭВМ;

– проявление одиночного сбоя во всем АПК.

Выявить подобное поведение с помощью других методов и на других этапах разработки и эксплуатации крайне затруднительно. Если ошибка была допущена на этапе валидации, то она может быть обнаружена только сторонними специалистами или теми, кто эксплуатирует систему. Если ошибка не была выявлена до эксплуатации, то диагностировать проблему и доказать её присутствие сложно в силу отсутствия детализированного логирования и каких-либо других средств мониторинга. Для выявления проблемы необходимо иметь знания о внутреннем алгоритме работы ПО, которым полноценно обладают только инженеры, ответственные за разработку. В силу сложности всего АПК и возможного случайного проявления отказа ошибка подобного рода может непредсказуемо влиять на поведение комплекса в целом и длительное время быть незаметной. Но, как показывает результат верификации в лаборатории, описанная проблема может быть обнаружена рассматриваемым методом. Найденные с применением доказательства корректности ошибки могут улучшить общее понимание работы системы и повлиять на весь жизненный цикл АПК – от постановки задачи до эксплуатации.

После испытаний лабораторией было сделано заключение, согласно которому блок ТУ16-1 может быть использован в случаях отсутствия передачи ответственных команд. В последующем проводилась модернизация блока ТУ16-1.

Очевидно, что похожая ошибка может проявиться и в других микроэлектронных устройствах, поэтому верификации процесса передачи параметров необходимо уделять особое внимание. В связи с этим при последующем рассмотрении в лаборатории микроэлектронных СЖАТ проверка корректности передачи управляющих воздействий стала выступать в качестве отдельной необходимой процедуры.

Полученный опыт может быть использован как для верификации готовых УСО СЖАТ, так и для разработки новых.

В настоящее время в лаборатории разрабатывается методика доказательства корректности для микроэлектронных СЖАТ, которая позволит эффективно осуществлять поиск ошибок с помощью описанного подхода.

Список литературы

1 **Сапожников, В. В.** Методы построения безопасных микроэлектронных систем железнодорожной автоматики / Вл. В. Сапожников, Х. А. Христов, Д. В. Гавзов; под ред. Вл. Сапожникова. – М. : Транспорт, 1995. – 272 с.

2 ТУ РБ 00047792.014-99. Система диспетчерской централизации и управления движением поездов ДЦ «Нёман». Технические условия. – Минск : КТБ Бел. ж. д., 1999.

3 ТУ РБ 00047792.004-99. Блок ТУ16-1 телеуправления. Технические условия. – Минск : КТБ Бел. ж. д., 1999.

4 ТО РБ БЧ 6102.808.01-98. Комплект линейный. Техническое описание и инструкция по эксплуатации. – Минск : КТБ Бел. ж. д., 1998.

5 **Hoare, C. A. R.** An axiomatic basis for computer programming / C. A. R. Hoare // CACM. – 12(10):576–580, 583. – October 1969. DOI:10.1145/363235.363259.

6 **Сапожников, В. В.** Дискретные устройства железнодорожной автоматики и телемеханики / В. В. Сапожников, Ю. А. Кравцов, Вл. В. Сапожников. – М. : Транспорт, 1988.

7 **Сапожников, В. В.** Теория дискретных устройств железнодорожной автоматики, телемеханики и связи: учеб. для вузов ж.-д. трансп. / В. В. Сапожников, Ю. А. Кравцов, Вл. В. Сапожников; под ред. Вл. Сапожникова. – 2-е изд., перераб. и доп. – М. : УМК МПС России, 2001. – 312 с.

8 **Graf, S.** Construction of abstract state graphs with PVS / S. Graf, H. Saidi // In Proc. of CAV'1997: Computer Aided Verification. – Vol. 1254 of LNCS. – 1997. – P. 72–83.

Получено 28.02.2012

B. V. Sivko. The proof of the correctness of the centralized dispatching control "Neman" telecontrol unit 16-1.

The verification results of the telecontrol unit 16-1 which is part of centralized dispatching control "Neman" are published. Verification order and features are considered. It is shown that proof of correctness allows to find software bugs which are extremely difficult to reveal by another methods.