УДК 004.312.466

Б. В. СИВКО, магистр технических наук, Белорусский государственный университет транспорта, г. Гомель

ДОКАЗАТЕЛЬСТВО КОРРЕКТНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МНОГОПРОЦЕССОРНЫХ УСТРОЙСТВ СВЯЗИ С ОБЪЕКТАМИ ЖЕЛЕЗНОДОРОЖНОЙ АВТОМАТИКИ И ТЕЛЕМЕХАНИКИ

Рассмотрены особенности доказательства корректности программного обеспечения многопроцессорных систем железнодорожной автоматики и телемеханики на примере блоков телеуправления 8Б и телесигнализации 16Б микропроцессорной централизации «Ипуть». Анализ опыта верификации позволил выделить категории доказываемых функций безопасности, определить последовательность этапов доказательства корректности и сформировать критерии оценки объема работ. Показано, что введение инварианта и доказательство его сохранения необходимо для исключения накопительных ошибок программного обеспечения. Установлено потенциальное наличие подлежащих верификации свойств, которые сложно точно сформулировать до проведения доказательства корректности, а их проверку должен осуществлять конечный пользователь.

Научно-технический прогресс немыслим без использования новейших достижений и передовых технологий, в том числе и на железнодорожном транспорте, где с начала 80-х годов XX века идет разработка и повсеместное применение аппаратно-программных комплексов (АПК) на микроэлектронной базе, что позволило поднять функциональность систем железнодорожной автоматики и телемеханики (СЖАТ) на новый качественный уровень. Однако использование широких возможностей микропроцессоров столкнулось с проблемой обеспечения безопасности, так как её решение для микроэлектронных устройств отличается от применяемых ранее подходов релейной схемотехники. В настоящее время отсутствуют универсальные и общепризнанные способы доказательства безопасности для микроэлектронных СЖАТ, из-за чего практикуется комплексный подход, заключающийся в применении множества методов и средств для повышения безопасности и надежности системы на всех этапах её жизненного цикла.

Доказательство корректности программного обеспечения (ПО) является одним из формальных методов его верификации и в настоящее время успешно применяется на Белорусской железной дороге для поиска ошибок, повышения качества проектирования устройств СЖАТ и их испытаний на безопасность [1–3].

Некорректное поведение СЖАТ влияет на безопасность движения поездов и может привести к большому экономическому ущербу, а в ряде случаев стоить жизни и здоровья людей, поэтому железнодорожные АПК относятся к критически важным объектам информатизации (*safety-critical systems*) [5, 6]. Для систем, связанных с безопасностью, стандарт IEC 61508 имеет градацию уровней полноты безопасности от SIL-1 до SIL-4, при этом железнодорожные системы должны соответствовать наиболее жесткому уровню эксплуатации SIL-4, который настоятельно рекомендует применение формальных методов для критически важных систем информатизации [7].

Из-за необходимости предоставления широкой функциональности и обеспечения высокого уровня эксплуатационной готовности современные микропроцессорные СЖАТ являются сложными и требуют развития методов обеспечения надлежащего уровня безопасности. Это обусловливает появление и использование систем, в которых для выполнения возложенных на них функций применяется несколько взаимодействующих друг с другом микропроцессоров, каждый из которых имеет отличное от других ПО. Безопасные блоки телеуправления (ТУ-8Б) и телесигнализации (ТС-16Б), разработанные для работы с напольными объектами микропроцессорной централизации «Ипуть», являются примерами АПК, обладающих описанными свойствами. Данные блоки проходили испытания на безопасность в лаборатории БЭМС ТС БелГУТа, в рамках которых доказательство корректности ПО являлось одним из пунктов верификации.

Рассмотрим устройство и особенности работы блока ТУ-8Б, общая схема которого и его взаимодействие с другими подсистемами показаны на рисунке 1.

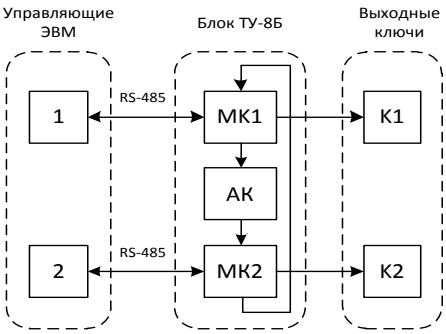

Рисунок 1 – Взаимодействие блока ТУ-8Б с другими подсистемами АПК

Блок ТУ-8Б имеет в своем составе три микроконтроллера (МК1, МК2, АК), каждый из которых выполняет отличные от других функции и использует разное ПО. Микроконтроллеры взаимодействуют друг с другом по внутреннему протоколу, передавая информацию о состоянии друг друга по кольцу (от МК1 идет передача посылок к АК, от АК – к МК2 и от МК2 – к МК1). Команды на управление блоку передает управляющая ЭВМ (это могут быть разные ЭВМ), взаимодействуя с микроконтроллерами МК1 и МК2 по интерфейсу RS-485 и внутреннему протоколу в асинхронном режиме, инициатором которого всегда выступает управляющая ЭВМ.

Блок ТУ-8Б имеет 8 выходов для управления, каждый из которых в состоянии включить одно управляемое устройство (реле первого класса надежности). У МК1 и у МК2 есть по 8 выходов, позволяющих инициировать включение выходных устройств безопасным образом, а

аппаратная схема построена так, что только в случае установки 1 от МК1 и от МК2 для каждого из выходных ключей произойдет включение управляемого объекта. Таким образом, описанное построение МК1 и МК2 задействует структурный метод обеспечения безопасности на основании аппаратного резервирования, реализуя дублированную систему с умеренными связями [8].

С точки зрения обеспечения безопасности АПК построен на основе стратегии применения логических элементов с несимметричными отказами (h_1-надежные элементы), когда ключ может принимать одно из двух состояний – 1 или 0, последнее из которых не может привести к опасному отказу [9]. Таким образом, если один из МК1 или МК2 выходит из строя (его показания расходятся с другим), то тот, который не был подвержен опасному отказу, переводит выходные ключи в 0, обеспечивая выполнение стратегии перехода в безопасное состояние в случае одиночного отказа [8]. В случае применения в АПК только одного микропроцессора подобное безопасное поведение получить невозможно, чем обусловливается необходимость использования более чем одного микроконтроллера.

ТУ-8Б имеет в наличии третий микроконтроллер АК, отвечающий за индикацию состояния всего блока и поддерживающий связь между МК1 и МК2.

Блок ТС-16Б с точки зрения постановки задач определения функции безопасности и последующего доказательства корректности обладает такими же особенностями, что и блок ТУ-8Б. ПО данных устройств представляет собой ассемблерный код нижнего уровня с количеством строк кода около двух тысяч (по определению Конта, *Conte*) для каждого из микроконтроллеров МК1 и МК2 [10]. Данные особенности позволяют эффективно использовать аксиоматический базис доказательства корректности в силу отсутствия дополнительных слоев (например, компилятора, внутренней виртуальной машины или операционной системы) между программной и аппаратной частями и при этом обладают приемлемой сложностью. Рассматриваемое ПО имело хороший уровень структурированности кода, что выражалось большим количеством процедур, абсолютное большинство из которых имело цикломатическую сложность по Мак-Кейбу не более 10, что позволило быстро определять их характеристики, предусловия и постусловия [11].

Для проведения доказательства корректности формулировались подлежащие верификации функции безопасности на основании:

– требований безопасности ко всей системе;

– выполняемых задач рассматриваемого АПК;

– особенностей протоколов взаимодействия с другими подсистемами;

– используемых стратегий обеспечения безопасности.

Соответственно, направления для задач верификации доказательства корректности ПО были сформулированы следующим образом:

– корректность запуска системы;

– безопасность передаваемых значений ключей;

– правильность индикации состояния блока;

– расчёт гарантированных временных пределов переходов между состояниями, влияющими на безопасность.

Для доказательства корректности ПО было представлено в виде графа троек логики Хоара и далее верифицировался созданный ориентированный граф, ребрами которого являются переходы, а вершинами – выполняемые программой действия [12].

Внутренний алгоритм ПО основан на циклическом выполнении функции *cyclic executive* (рисунок 2) [13].

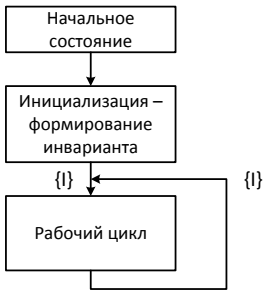

Рисунок 2 – Циклическое выполнение внутреннего алгоритма

Для верификации корректности запуска системы требовалось определить условия выхода в рабочий режим, после чего начиналось непосредственное выполнение рабочего цикла, во время которого ТУ-8Б реализовывал возложенную на него функцию.

Блок ТУ-8Б является многопроцессорным, и его старт сопровождается процедурой синхронизации, во время которой происходит обмен сообщениями между микроконтроллерами МК1, МК2 и АК. После обмена, если он прошел успешно, каждый из микроконтроллеров стартует по внутреннему рабочему циклу и повторяет его бесконечное количество раз (если не происходит сброса информации). Таким образом, доказательство корректности представляет собой последовательность следующих шагов:

1 Определение точки начала рабочего цикла. Это такая точка алгоритма, которая гарантированно выполняется на каждом витке цикла.

2 Выполнение доказательства корректности сверху вниз от начального условия до точки начала рабочего цикла. На данном этапе может эффективно использоваться метод предикатов абстракций (*Predicate Abstraction*) для облегчения процедуры вывода [14].

3 Для полученных условий на шаге 2 необходимо провести доказательство того, что в случае работы многопроцессорной системы выход на рабочий цикл всех микропроцессоров обязательно завершится.

4 Определение условий выхода системы в рабочий режим для всех микропроцессоров.

5 Согласование полученных условий на шагах 2 и 4 на уровне их валидации, то есть с конечным пользователем системы и с условиями безопасности. Данное действие необходимо потому, что на первых четырех шагах определяются свойства верифицируемого АПК и при этом отсутствуют критерии их проверки.

Проведение доказательства корректности ПО по запуску информации необходимо выполнять до последующих этапов, так как его результаты используются для выполнения дальнейших задач верификации.

Для рассматриваемых блоков ТУ-8Б и ТС-16Б следующим этапом доказательства корректности являлось определение инвариантов циклов рабочего режима для каждого из микроконтроллеров, что проводилось с использованием результатов верификации запуска. Особенность в том, что выведенное условие старта (P) при попадании в начальную точку рабочего цикла является основанием для формирования инварианта (I), так как должно выполняться условие $P \in I$. В дальнейшем инвариант

имеет тенденцию расширяться, так как состояние микроконтроллера может изменяться на каждом витке цикла. Расширением инварианта можно считать любое возможное предусловие для начальной точки рабочего режима. Другими словами, если происходит возврат в начальную точку, то это означает, что цикл выполняется, но если алгоритм не попадает в начальную точку, то это свидетельствует о выходе из рабочего режима.

Найденный таким образом инвариант подлежит проверке, что на каждом витке цикла он выполняется и не будет нарушен. Доказательство корректности ПО проводится с целью исключения накопительных ошибок, которые могут проявляться редко или в результате длительной эксплуатации. Источником такого рода ошибок является зависимость состояния микропроцессора от его предыдущих состояний, и если мы доказываем, что имеет место сохранение инварианта, то тем самым верифицируется отсутствие ошибок данного типа, которые крайне опасны, так как их чрезвычайно трудно выявить отличными от доказательства корректности методами.

Инвариант используется для следующего этапа верификации – доказательства свойств системы, в том числе и имеющих отношение к безопасности. Например, для блока ТУ-8Б одними из доказываемых свойств являлись гарантированный переход выходного ключа из 1 в 0 по истечении тайм-аута и отсутствие самопроизвольного перехода (без наличия соответствующей команды от управляющей ЭВМ в заданный период времени) выходного ключа из 0 в 1. Если найденный инвариант не прошел проверку определения или доказываемых свойств, то это свидетельствует либо о некорректности определения инварианта (он должен быть усилен, ослаблен или изменен), либо о наличии программной ошибки.

Для ПО блоков ТУ-8Б и ТС-16Б оказался возможным переход от дедуктивного анализа к верификации модели (*model checking*), что улучшило точность понимаемого поведения программы и уменьшило затраты на доказательство корректности [15]. В качестве модели рассматривались состояния ключей микроконтроллера и условия переходов между ними, когда для каждого из ключей на начало цикла было представлено его состояние (0 или 1), а в соответствии с логикой тела цикла формализованы переходы из одного состояния в другое, как показано на рисунке 3, где $\overline{\alpha}$ – состояние микроконтроллера и внешних условий во время выполнения всего тела цикла; $F_{(0\to1)(i)}$ – функция условия перехода из 0 в 1 для рассматриваемого i-го ключа (принимает значения: истина или ложь); $F_{(1\to0)(i)}$ – аналогичная функция перехода из 1 в 0 для того же ключа.

Используя предикат абстракций для $\overline{\alpha}$, можно определить варианты развития событий, при которых обязательно осуществляется переход или гарантированно не происходит перехода из одного состояния в другое.

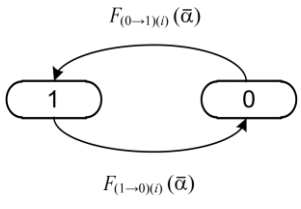

Рисунок 3 – Переходы между состояниями ключа

Определенные и проверенные на данном этапе функции F могут быть успешно использованы на следующем этапе для расчета временных характеристик переходов между состояниями.

Во время верификации временных параметров функционирования необходимо сначала доказать то, что при рассматриваемом предикате или в общем случае программа обязательно выполнится от начальной точки до конечной. После этого для выбранного подмножества $\overline{\alpha}$ функции F требуется провести анализ поведения и расчёт характеристик. Как правило, в верифицируемых микропроцессорных СЖАТ требуется определить только максимальное или минимальное время выполнения, что во многом облегчает задачу верификации. Кроме того, как показывает практика, в железнодорожных устройствах связи с напольными объектами вычислительные мощности микроконтроллеров для реализации функциональности используются на уровне единиц процентов для самых длительных сценариев поведения, поэтому для упрощения процесса верификации временных характеристик может оказаться удобным проводить расчет минимального и максимального времени работы тела цикла без учета логики программы и в дальнейшем использовать данные значения для определения периода возврата в начальную точку рабочего режима. При таком рассмотрении переход по состояниям функций F осуществляется за конечное время, пределы возможного диапазона которого достаточно ослаблены для того, чтобы доказательство корректности проводилось с минимальными усилиями, но при этом рамки достаточно жесткие для того, чтобы выполнение условий безопасности успешно доказывалось.

Кроме состояний ключей в блоках ТУ-8Б и ТС-16Б имеются другие внутренние состояния, от которых зависит поведение АПК, и переходы между ними могут быть проверены с помощью верификации модели [15]. Эти состояния и их описания показаны в таблице 1.

Таблица 1 – **Диагностические состояния ТУ-8Б и ТС-16Б**

Состояние	Тип	Условие перехода в состояние
ERR1	Ошибка связи	Отсутствие корректных посылок от управляющей ЭВМ
ERR2	Внутренняя ошибка	Замыкание кольца при проверке идентификатора; короткое замыкание выходов МК1 или МК2; тайм-аут МК1 → АК; несоответствие номера АК дополнению
ERR3	Ошибка выходных ключей	Обнаружение проблем выходных умножителей 4 В, 24 В
OK	Нет ошибок	Блок успешно выполняет свои функции

Каждый из блоков в любой из моментов функционирования находится в одном из состояний, которое определяет выдаваемую внешнему пользователю индикацию. Диагностика состояния блока непосредственно относится к вопросам безопасности, так как в случае её корректности возможна замена блока при возникновении отказа. Если же индикация работает неверно, то обнаружение отказавшего блока становится затруднительным и, кроме того, повышается вероятность возникновения второго отказа в блоке, включенном в эксплуатацию, что недопустимо, так как АПК проектировался и испытывался на основании стратегии обнаружения одиночных отказов, которая работает только в случае, когда первый отказ гарантированно обнаружи-

вается и его последствия ликвидируются за фиксированное предельное время [8].

Доказательство корректности индикации блоков проводилось в несколько этапов. Сначала были определены возможные состояния и условия переходов между ними (см. таблицу 1). Далее данные условия переходов проверялись на соответствие функции безопасности: условие перехода в более безопасное состояние должно быть достаточным в отношении необходимой функции безопасности, т. е. множество состояний выполнения функции безопасности должно быть включено во множество, определяемое условием перехода.

Следующим шагом верификации являлось определение свойств индикации в случае нахождения блока в соответствующем состоянии. Индикация представляла собой динамическое поведение светодиода, которое например в случае присутствия ошибок ERR1 и ERR3 можно представить в виде графика, показанного на рисунке 4.

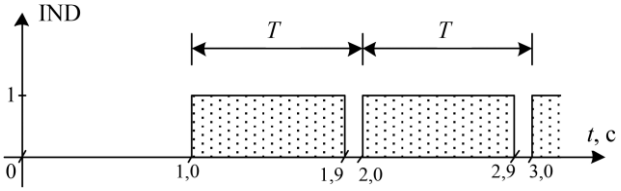

Рисунок 4 – Индикация ошибочного состояния ERR1 и ERR3

К описываемому поведению не прилагалось точных спецификаций, и корректность поведения мог определить только конечный пользователь. В связи с этим полученные графики для каждого из состояний рассматривались человеком на корректность. Например, проиллюстрированная индикация показывает, что светодиод первую секунду не горит, а в последующем выходит в стабильный режим повторения горения индикации в течение 0,9 с и отсутствия горения в течение 0,1 с. Таким образом, доказательство корректности ПО представляет собой трудноопределяемую потенциальную проблему, а пользователь системы принимает решение о том, насколько такое поведение корректно. В данном случае, так как переходы между ошибочными состояниями происходят крайне редко, а сам период индикации сравним с периодом запоздания, одна секунда задержки не считается существенной с точки зрения безопасности и корректности функционирования блока.

Во время верификации блоков ТУ-8Б и ТС-16Б были строго определены некоторые из свойств системы, а также потенциально ошибочные варианты поведения (такие как отсутствие реакции на разрывы связи между микроконтроллерами МК2 и МК1, АК и МК1, зависимость параметров синхронизации только от одного микроконтроллера МК2). Кроме того, одним из результатов являлась рекомендация о проведении испытаний для длительного периода между сеансами передачи между микроконтроллерами.

Опыт верификации систем подобной сложности показывает, что доказательство корректности способно находить потенциальные дефекты во всем АПК, более точно определять поведение системы и вырабатывать рекомендации по её улучшению.

Список литературы

1 **Сивко, Б. В.** Доказательство корректности блока телеуправления 16-1 диспетчерской централизации «Неман» / Б. В. Сивко // Вестник БелГУТа: Наука и транспорт. – 2012. – № 1 (24). – С. 18–21.

2 **Butler, R. W.** What is Formal Methods? / R. W. Butler. – NASA LaRC Formal Methods Program, 2001.

3 **Харлап, С. Н.** Верификация программного обеспечения микропроцессорной светооптической светодиодной системы / С. Н. Харлап, Б. В. Сивко // Вестник БелГУТа: Наука и транспорт. – 2012. – № 1 (24). – С. 22–25.

4 **Сивко, Б. В.** Проектирование безопасного программного обеспечения микропроцессорных устройств автоматики и телемеханики / Сивко Б. В. // Проблемы безопасности на трансп. : тез. докл. VI Междунар. науч.-практ. конф., Гомель, 29–30 ноября 2012 г. / М-во образования Респ. Беларусь, М-во трансп. и коммуникаций Респ. Беларусь, Бел. ж. д., Белорус. гос. ун-т трансп.; редкол. : В. И. Сенько (отв. ред.) [и др.]. – Гомель, 2012. – С. 205.

5 **Knight, John C.** Safety critical systems: challenges and directions / John C. Knight // ICSE '02 Proceedings of the 24th International Conference on Software Engineering, 2002. – P. 547–550.

6 **Sommerville, I.** Software engineering / Ian Sommerville. – Addison-Wesley, 2007. – P. 44.

7 **Smith, David J.** Safety Critical Systems Handbook. A Straightforward Guide to Functional Safety, IEC 61508 and Related Standards, Including Process IEC 61511 and Machinery IEC 62061 and ISO 13849 / David J. Smith and Kenneth G. L. Simpson. – Elsevier Ltd., 2010.

8 **Бочков, К. А.** Методы обеспечения безопасности в микропроцессорных системах железнодорожной автоматики и телемеханики : учеб. пособие / К. А. Бочков, С. Н. Харлап. – Гомель : БелГУТ, 2001.

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

10 **Conte, S. D.** Software Engineering Metrics and Models / S. D. Conte, H. E. Dunsmore, Y. E. Shen. – San Francisco : Benjamin Cummings Publishing Co., Inc. 1986.

11 **McCabe, T. J.** A Complexity Measure / T. J. McCabe // IEEE Transactions of Software Engineering. – 1976. – Vol. 2. – No. 4. – P. 308–320.

12 **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.

13 **Locke, D. C.** Software Architecture for Hard Real-Time Applications: Cyclic Executives vs. Fixed Priority Executives / D. C. Locke // Real-Time Systems 4(l) : 37–53 (1992).

14 **Graf, S.** Construction of abstract state graphs with PVS / Saidi H. // Proc. of CAV'1997: Computer Aided Verification, vol. 1254 of LNCS, p. 72–83. Springer, 1997.

15 **Clarke, Edmund M.** Model Checking / Edmund M. Clarke, Orna Grumberg and Doron A. Peled // MIT Press, 1999.

Получено 28.11.2012

B. V. Sivko. Proof of correctness of multiprocessor software of microprocessor railway systems.

The features of the software proof of correctness for multiprocessor railway systems on the example of remote control units 8Б and 16Б of the microprocessor railway signalling system "Iput" has been considered. Analysis of the experiences of the verification are proved possible to distinguish the category of safety functions for proof, determine the sequence of steps of the proof and to make criteria for assessing the amount of work for proof. It is shown that the determination of the invariant and the proof of it are necessary to avoid accumulation software errors.

Established the potential presence of the features for proof, which are difficult to formalize precisely before the proof of correctness take place, and the verification result must be checked by the end user, for showed instance, at the monitor diagnostic subsystem.