АВТОМАТИКА, ТЕЛЕМЕХАНИКА И СВЯЗЬ

УДК 004.312.466

К. А. БОЧКОВ, доктор технических наук, С. Н. ХАРЛАП, кандидат технических наук, Б. В. СИВКО, магистр технических наук, Белорусский государственный университет транспорта, г. Гомель

ОЦЕНКА ВРЕМЕННЫХ ПАРАМЕТРОВ ФУНКЦИОНИРОВАНИЯ МИКРОПРОЦЕССОРНЫХ УСТРОЙСТВ СВЯЗИ С ОБЪЕКТАМИ СИСТЕМ ЖЕЛЕЗНОДОРОЖНОЙ АВТОМАТИКИ И ТЕЛЕМЕХАНИКИ

Предложен способ оценки временных параметров функционирования микропроцессорных устройств сопряжения с напольными объектами железнодорожной автоматики и телемеханики с помощью доказательства корректности программного обеспечения с применением ослабления условия без учета логики программы. Показано, что данный метод эффективно применяется для получения строгих доказываемых характеристик системы и может быть с минимальными затратами использован во время проведения общей верификации.

Приведены результаты анализа микроэлектронных железнодорожных устройств сопряжения с напольными объектами с использованием описываемого подхода. Показано, что верификация программного обеспечения нижнего уровня с целью оценки временных параметров может быть быстро и эффективно проведена с помощью предлагаемого метода.

В настоящее время широко применяются системы железнодорожной автоматики и телемеханики (СЖАТ) на микроэлектронной базе: микропроцессорные централизации стрелок и сигналов, многозначные автоматические локомотивные сигнализации, микропроцессорные диспетчерские централизации и др. Принципиальным отличием данного этапа развития СЖАТ является то, что в них повсеместно используются программные способы реализации алгоритмов управления и контроля, позволяющие добиться большей функциональности и быстродействия, но микроэлектронная элементная база не позволяет решить важнейшую проблему – обеспечение безопасности – теми же способами, что и релейная [1]. В связи с отсутствием единого подхода для доказательства безопасности применяется комплекс методов, каждый из которых имеет свои достоинства и недостатки, что приводит к тому, что идет поиск наиболее эффективных подходов, позволяющих решить ключевые проблемы обеспечения безопасности.

К настоящему времени лабораторией БЭМС ТС БелГУТа накоплен опыт испытаний и анализа на безопасность микроэлектронных устройств СЖАТ низкого уровня, таких как блоки телеуправления (ТУ16-1) и телесигнализации (ТС32-1) диспетчерской централизации «Неман», безопасные блоки управления напольными объектами ТУ-8Б и ТС-16Б микропроцессорной централизации «Ипуть» и другие устройства, выполняющие функции контроля, сигнализации, сопряжения и диагностики [2]. Безопасность рассматриваемых аппаратно-программных комплексов (АПК) зависит от имеющегося в их составе программного обеспечения (ПО), что ведет к необходимости его верификации, которая, в свою очередь, требует проведения оценки временных параметров.

Строгое доказательство анализируемых временных характеристик АПК является одной из задач обеспечения безопасности функционирования микропроцессорных СЖАТ, так как они относятся к критически важным объектам информатизации, работающим в реальном времени. Примерами анализируемых свойств являются:

– длительность реакции АПК на внешние воздействия;

– тайм-ауты перехода системы в безопасное или неактивное состояние;

– время обработки информации в определенных условиях и параметрах функционирования;

– период обновления устройств индикации;

– длительность задержек во времени программным способом;

– гарантированное время перехода системы из одного состояния в другое.

Определение и строгое доказательство значений временны́х характеристик проводились с привлечением формальных методов (*Formal Methods*), а именно: с использованием доказательства корректности на основе логики Хоара [3, 4]. Созданная с помощью описанных методов модель ПО представляет собой ориентированный граф, узлами которого являются точки предусловий и постусловий выполняемых операторов, ребра – конкретные действия программы, а направления рёбер задаются согласно последовательности выполнения алгоритма.

В микроэлектронных СЖАТ, как правило, отсутствуют какие-либо специализированные средства строгой выдержки временных параметров, и для реализации требуемой функциональности могут использоваться как аппаратные, так и программные решения. В случае последних происходит передача контроля выполнения на предназначенное для этого ПО, время выполнения и задержки которого зависят от характеристик микроэлектроники, на которых оно работает, например от частоты тактового генератора. В таких случаях оценить

время работы последовательности операторов можно на основании документации по рассматриваемым устройствам. Таким образом, время выполнения каждой команды микроконтроллера может быть рассчитано в тактах микропроцессора, а в общем случае требуется учёт характеристик аппаратных средств.

Рассматриваемое ПО преобразуется в граф на основании логики Хоара и детализируется до уровня команд микропроцессора. Для оценки времени в каждой из вершин вводится параметр t, который изменяется согласно времени выполнения программы и чаще всего измеряется в тактах. На каждом ребре отмечается два значения – пределы минимального и максимального времени для рассматриваемого доказательства корректности ПО (рисунок 1).

Рисунок 1 – Формализация оценки временных параметров ребра графа

Тогда предусловие может быть получено из постусловия посредством следующих соотношений (1, 2):

$$t_{min}(V) = t_{min}(P) - t_{max}(S); \quad (1)$$

$$t_{max}(V) = t_{max}(P) - t_{min}(S). \quad (2)$$

С помощью (1, 2) можно проводить доказательство корректности снизу вверх.

Описанное представление позволяет вывести постусловие из предусловия и на его основании можно проводить доказательство корректности ПО сверху вниз (3, 4):

$$t_{min}(P) = t_{min}(V) + t_{min}(S); \quad (3)$$

$$t_{max}(P) = t_{max}(V) + t_{max}(S). \quad (4)$$

Последовательность двух блоков операторов может быть объединена в одну, как показано на рисунке 2.

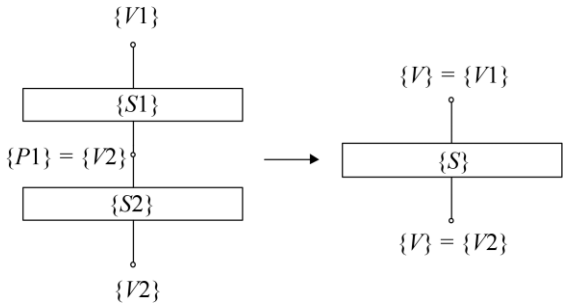

Рисунок 2 – Объединение двух последовательных блоков алгоритма

Здесь преобразование производится согласно объединению последовательности операторов (*правило вывода S*1 [5]), при котором параметры минимального и максимального времени выполнения вычисляются (5, 6):

$$t_{min}(S) = t_{min}(S1) + t_{min}(S2); \quad (5)$$

$$t_{max}(S) = t_{max}(S1) + t_{max}(S2). \quad (6)$$

Два параллельных блока алгоритма могут быть объединены так, как показано на рисунке 3.

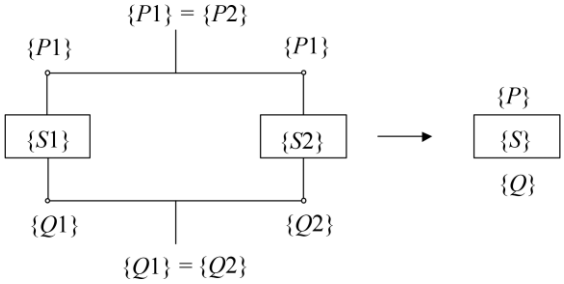

Рисунок 3 – Объединение параллельных блоков алгоритма

В данном случае параметры минимального и максимального времени выполнения будут связаны выражениями

$$t_{min}(S) = \min(t_{min}(S1), t_{min}(S2)); \quad (7)$$

$$t_{max}(S) = \max(t_{max}(S1), t_{max}(S2)), \quad (8)$$

где $\min(X, Y)$ – функция, возвращающая минимальное значение одного из двух аргументов X и Y; $\max(X, Y)$ – функция, возвращающая максимальное значение одного из двух аргументов X и Y.

Для иллюстрации особенностей подхода рассмотрим пример преобразования программы в граф и расчет временных характеристик с помощью данного метода. Программа написана для микроконтроллера P16C73 в соответствующем синтаксисе и представлена в таблице 1 [6].

Таблица 1 **– Пример программы P16C73**

Метка	Команда и аргументы	Время выполнения, в тактах
	BTFSS TCNHP,0	1 или 2[1]
	GOTO MIN1	2
	MOVLW 1	1
	ADDWF ST13,F	1
	BTFSS STATUS,C	1 или 2[1]
	GOTO MIN1	2
	DECF ST13,F	1
	INCF ST14,F	1
MIN1	BTFSC TCNHP,0	1 или 2[1]
	GOTO MIN2	2
	MOVLW 1	1
	ADDWF SST15,F	1
	BTFSS STATUS,C	1 или 2[1]
	GOTO MIN2	2
	DECF SST15,F	1
	INCF ST16,F	1
MIN2		
Примечание – 1 – один такт, если условие не выполняется; 2 – два такта если условие выполняется и пропускается последующая команда.		

На рисунке 4 показаны алгоритм программы, представленный в виде графа, и информация о вычислении временных характеристик его работы при доказательстве сверху вниз.

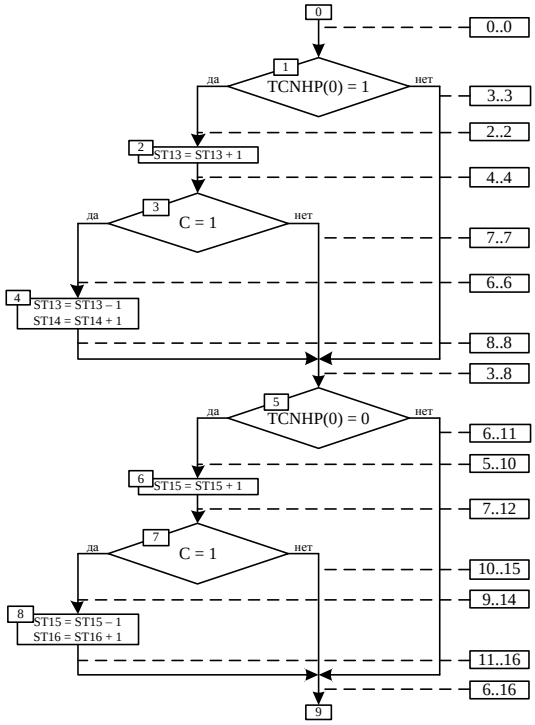

Рисунок 4 – Анализ временных характеристик алгоритма

Алгоритм имеет ряд точек-вершин, в которых может быть определено минимальное и максимальное время выполнения с помощью представленных в методе преобразований (3–8), а результаты вычисления показаны на рисунке справа. Этот же алгоритм в виде графа показан на рисунке 5.

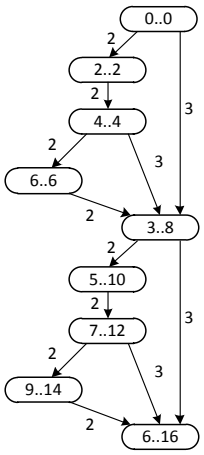

Рисунок 5 – Граф расчета временных параметров

В данном примере демонстрируется, что в ряде случаев анализ теряет возможные значения времени, находящиеся между максимальным и минимальным пределами. Таким образом, в предусловии блока 5 время выполнения может быть 3, 7 или 8 тактов. Значение 7 тактов теряется как несущественное. Аналогично, в конце выполнения программы её время, исходя из анализа, составляет от 6 до 16 тактов, а значения 10, 11, 14, 15 оказываются отброшенными как несущественные.

Приведенный пример показал, что анализ не затрагивает логику работы программы. Если значение *TCNHP*(0) может находиться только в состоянии 1 или 0, то количество вариантов работы программы уменьшается. Если алгоритм отрабатывает по переходу 1–2, то он должен пройти по переходу 5–9. Аналогично, если имеет место переход 1–5, то должен произойти переход 5–6. В таком случае при необходимости более точной оценки можно использовать предикаты {*TCNHP*(0) = 1} и {*TCNHP*(0) = 0}, тем самым упростив граф и улучшив результат, который в данном случае будет являться диапазоном от 10 до 11 тактов включительно.

Представленный метод имеет существенные ограничения в применении, но при этом обладает рядом преимуществ и, как показывает практика верификации реальных микропроцессорных СЖАТ, позволяет быстро и эффективно оценивать временные параметры работы АПК и доказывать их корректность.

К недостаткам описанного метода относятся:

– отсутствие анализа логики ПО;

– необходимость структурированности ПО;

– необходимость дополнительных действий в случае присутствия циклов.

Из-за отсутствия анализа логики работы программы (исходя из состояний внутренних переменных), можно получить строгую, но более слабую оценку времени. Практика показывает, что в абсолютном большинстве случаев того, что дает представленный метод, оказывается достаточно, и при этом более детальное рассмотрение возможно, но не является необходимым.

Анализ поведения АПК в конкретных ситуациях эффективно проводить с помощью предикатов абстракций (*Predicate Abstraction*) [7]. Ввод предиката производится до того, как запускается процесс оценки с помощью описываемого метода и при этом граф работы программы зачастую существенно упрощается. Таким образом, использование предикатов абстракций позволяет эффективно повысить точность оценки и уменьшить затраты на верификацию.

Предлагаемый в работе метод не позволяет рассчитывать временные параметры для условных неструктурированных GOTO-переходов и циклов. Разрешение данного вопроса предполагается вне описанных инструментов.

Условные неструктурированные GOTO-переходы в любом случае сложны для анализа, и их необходимо избегать на этапах проектирования и разработки [8]. Однако в случае их присутствия в готовом ПО, которое нельзя изменить, возможно для каждой точки ветвления определить условие, на основании которого сформировать предикат и последующий граф выполнения.

Практика показывает, что абсолютное большинство циклов ПО устройств низкого уровня СЖАТ имеют фиксированное количество повторений, число которых может быть определено заранее. Соответственно, для оценки времени работы алгоритма достаточно представить цикл в виде последовательности блоков и произвести расчёт. Параметризованные циклы рассматриваются вне описанных инструментов.

Необходимость использования парадигмы структурного программирования на этапе проектирования и разработки обосновывается как теоретической возможностью представления любого алгоритма в виде последовательностей, ветвлений и циклов, так и опытом верификации, когда неподготовленное ПО не поддается анализу и при этом содержит труднонаходимые ошибки [9, 10].

К достоинствам метода относятся:

– простота и эффективность использования;

– покрытие большинства задач верификации;

– гибкое применение с другими методами;

– легкость автоматизации.

Рассматриваемый метод успешно применялся для верификации ПО со сложностью в сотни и тысячи строк кода низкого уровня, даже с отсутствием автоматизации, при котором затраты на расчёт сопоставимы со временем формализации алгоритма (создание графа троек Хоара).

Основной задачей верификации безопасного поведения АПК микроэлектронных СЖАТ при оценке временных параметров является определение минимального и максимального периодов работы между различными событиями. С точки зрения безопасности устройства имеют временные пределы для переходов из одного состояния в другое, границы во времени на реакции изменения входных сигналов, гарантированные периоды обмена информацией между подсистемами и др. Для решения всех задач данного типа достаточно строгого определения рамок диапазона времени работы ПО.

Предлагаемый набор инструментов эффективно используется совместно с другими известными методами верификации, так как не имеет дополнительных требований. Возможно его применение для небольших локальных последовательностей операторов с дальнейшей компоновкой при необходимости более точного и глубокого анализа. Кроме того, в случае его ограниченности применения и требований большей точности отдельные части ПО могут быть изъяты, рассмотрены отдельно и легко интегрированы обратно в контекст верификации.

Опыт доказательства корректности ПО показывает, что в большинстве устройств СЖАТ вычислительные ресурсы микроконтроллеров используются не более чем на 1 %. Тем не менее, возможны рассинхронизации взаимодействующих подсистем, быстрые небезопасные переходы между состояниями, длительные внутренние циклы и др. Данные проблемы могут возникать редко и оставаться незамеченными как при тестировании, так и во время длительной эксплуатации. Использование доказательства корректности ПО позволяет уменьшить вероятность подобных проявлений и улучшить безопасность и надежность АПК.

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

3 **Butler, R. W.** «What is Formal Methods?» NASA LaRC Formal Methods Program, 2001.

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

5 **Бейбер, Р. Л.** Программное обеспечение без ошибок : пер. с англ. / Р. Л. Бейбер; / под ред. Д. И. Правикова. – М. : Джон Уайли энд Санз, Радио и связь, 1996. – 176 с.: ил. – Пер. изд.: Baber, Robert Laurence. Error-Free Software: Know – How and Know Why of Program Correctness. – John Wiley & Sons Ltd., Chichester, 1991.

6 **Verle, Milan.** PIC Microcontrollers / mikroElektronika. – 1st edition (2008) / ISBN-13: 978-86-84417-15-4.

7 **Graf, S.** Construction of abstract state graphs with PVS / Saidi H. / In Proc. of CAV'4997: Computer Aided Verification. Vol. 1254 of LNCS, p. 72–83. Springer, 1997.

8 **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.

9 **Bohm, Corrado; and Giuseppe Jacopini.** «Flow Diagrams, Turing Machines and Languages with Only Two Formation Rules» / Communications of the ACM 9 (5): 366–371. (May 1966) DOI: 10.1145/355592.365646.

10 **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.

Получено 07.12.2012

K. A. Bochkov, S. N. Kharlap, B. V. Sivko. Assessing time specifications of microprocessor interface rail automation and remote control devices.

The verification method of assessing time parameters of microprocessor interface railway automation units and remote control devices by proving correctness basis with weakening conditions without logic checking has been proposed. It is shown that the method is fruitful when it use for strictly proved features of the system and can be used with minimal costs during the general proof of correctness.

Results of the analysis of microelectronic railway devices using this approach has been published. It is shown that the verification of low-level software can assess the time specifications and can be quickly and efficiently carried out by using the provided method.