АВТОМАТИЗАЦИЯ ВЕРИФИКАЦИИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ СВЯЗАННЫХ С БЕЗОПАСНОСТЬЮ СИСТЕМ

Сивко Б. В.
Лаборатория «Безопасность и электромагнитная совместимость технических средств»,
Белорусский государственный университет транспорта
Гомель, Республика Беларусь
E-mail: bsivko@gmail.com

Рассматривается практика верификации программного обеспечения систем, связанных с безопасностью. В докладе излагается несколько задач верификации, решение которых проводится автоматизированными программными средствами. В качестве таких задач рассматривается определение временных параметров работы устройств, оценка степени диверситета аппаратно-программных комплексов, автоматизация выбора адресов по методу обнаружения отказов на основе доступности адресных данных. Решения основываются на формальной верификации исходного кода программ.

Введение

Разработка систем, связанных с безопасностью (*safety-critical systems*, ССБ), сопряжена с дополнительными мероприятиями и затратами на обеспечение предъявляемых к ним требований безопасности. Данная задача усложняется для микропроцессорных аппаратно-программных комплексов, так как они обладают высокой сложностью, присущей программному обеспечению (ПО). В то же время, к ССБ предъявляются повышенные требования по безопасности и надежности функционирования, что создаёт потребность в создании эффективных методов и средств для повышения качество решения ключевых проблем разработки и верификации [1].

Одним из способов решения описанной задачи является создание автоматизированных программных средств, позволяющих снизить влияние человеческого фактора и предоставить дополнительный способ полного охвата заданных спецификациями положений. Помимо этого, средства автоматизации могут уменьшить затраты посредством выявления ошибок проектирования на более ранних стадиях разработки — до имитационных испытаний. Для ССБ автоматизация прежде всего рассматривается как дополнительный способ повышения показателей отказоустойчивости и безопасности. Трудности данного направления обусловлены разнообразием решаемых задач и особенностями элементной базы, что осложняет необходимую для автоматизации формализацию.

I. Оценка временных параметров

Ряд ССБ относятся к системам реального времени, и поэтому для них характерно требование выполнения определённых временных параметров, например, гарантии перехода в безопасное состояние по заданному тайм-ауту, предоставление возможности задания периода обновления устройств индикации, определение частоты опроса внешних устройств и других [1].

Практика верификации ПО ССБ железнодорожной автоматики и телемеханики показывает, что к основным задачам такого типа относятся: вычисление времени выполнения между двумя произвольными точками, определение обстоятельств зацикливания программы, а также доказательство обязательного завершения алгоритма. Кроме того, на практике для циклических систем оказывается актуальным поиск точек программы, где выполнение обязательно произойдёт при каждом выполнении тела цикла. Для автоматического решения данных задач разработано ПО *Formal Time Verifier* [2], с помощью которого возможна верификация программ PIC-контроллеров модели 16F877A, а так как используется общая база команд исполнения PIC, то ПО может применяться для микроконтроллеров других модификаций. Функционально *Formal Time Verifier* проводит синтаксический разбор исходного кода и создаётся граф переходов, и разработанные алгоритмические решения представляют собой решения задач на графах.

II. Оценка степени диверситета

Диверситет является одним из основных способов повышения отказоустойчивости и безопасности ССБ, заключающийся в создании как можно более разных систем таким образом, чтобы в случае отказа они повели себя по-разному, что позволяет обнаружить отказ, провести диагностику, перейти безопасное состояние или в режим самовосстановления. Здесь одной из важных задач становится оценка степени полученного диверситета (то есть, различия), позволяющая определить эффективность применяемых методов и средств [1, 3].

Автоматизация решений проблем, связанных с диверситетом, является сложной и актуальной задачей, так как оценка степени достигнутого диверситета на современном этапе производится неформальными методами. Одним из

методов формализации является диверсификация аксиоматических базисов, когда разрабатываемая система опирается на заранее определённые формализованные утверждения [3]. Соответственно, возможна разработка средств автоматизации, позволяющих проверить выполнение данных утверждений для разработанной системы и их истинность во время проявления тех или иных отказов.

Для этих целей разработано ПО *Diverse Axiomatic Basis Checker* [4], позволяющее определить базисы на основе исходного кода программ и информации об микропроцессорной архитектуре. ПО позволяет верифицировать программы для PIC-контроллеров модели 16F877A. Для других моделей или систем необходима адаптация (задание конфигурации) определения базисов. *Diverse Axiomatic Basis Checker* проверяет микропроцессорную систему на константные отказы (*SA*, *stuck-at faults*) и отказы короткого замыкания произвольных информационных линий (*B*, *bridging faults*), в которые входят отказы ячеек памяти, дешифратора команд и выполнения инструкций микроконтроллера, отказы регистров и аккумулятора. Практика применения *Diverse Axiomatic Basis Checker* показала, что подход дополнительно позволяет проверить ряд утверждений, которые вручную проверять затруднительно – например, влияние отказов при рассмотрении больших объёмов памяти. В то же время, ограничениями подхода стали сложность реализации проверки утверждений, в частности, когда для анализа необходима дополнительная информация о работе алгоритма [4, 5].

III. Выбор адресов

Во время применения метода обнаружения отказов на основе доступности адресных данных происходит выбор определённого набора адресов, который зависит от того множества отказов, наличие которых требуется проверить. Идея метода состоит в том, что в случае отказа один из адресов становится недоступным, и на этом основании система может обнаружить проблему, перейти в безопасное состояние или запустить процедуры самовосстановления [6].

У произвольной системы некоторые области адресного пространства заняты исполняемым кодом или данными, и эти области формализуются в виде диапазонов, которые могут быть определены исходя из исходного кода. Во время практического применения требуется, чтобы пользователь мог задать важные для него критерии (множество проверяемых отказов, разрешенные адресные диапазоны) и найти оптимальные варианты. Для решения описанной задачи разработано ПО *Address Detection* [7, 8], реализующее цели автоматизации: уменьшение ошибок во время поиска адресов, нахождение оптимального из возможных наборов и уменьшение затрат. ПО выполняет расчёт в рамках базовых моделей *SA*- и *B*-отказов. Решение для SA-отказов является простой задачей, так как они задают конкретные биты адресного регистра. Для одного *B*-отказа задача эквивалентна раскраске графа в два цвета, которая решается за линейное время. Для большего числа *B*-отказов проблема становится *NP*-сложной. *Address Detection* предоставляет решение исходя из практических потребностей, позволяя вычислять пары адресов с минимальным расстоянием друг от друга и с наибольшим минимальным адресом с начала адресного пространства.

Заключение

В докладе рассматриваются задачи автоматизации, разработанные алгоритмы и особенности применения предложенных программных средств. ПО *Formal Time Verifier*, *Diverse Axiomatic Basis Checker* и *Address Detection* опробованы в лаборатории «БЭМС ТС» БелГУТа и зарегистрированы в 2017 году в Национальном центре интеллектуальной собственности, г. Минск [2, 4, 7].

1. Бочков, К. А. Микропроцессорные системы автоматики на железнодорожном транспорте : учеб. пособие / К. А. Бочков, А. Н. Коврига, С. Н. Харлап; М-во образования Респ. Беларусь, Белорусский государственный университет транспорта. – Гомель. – 2013.
2. Сивко, Б. В. Formal Time Verifier: свидетельство о регистрации компьютерной программы в Национальном центре интеллектуальной собственности Республики Беларусь № 995 / Б. В. Сивко. – Опубл. 13.12.2017.
3. Бочков К. А., Разработка отказоустойчивых систем на основе диверситетных аксиоматических базисов / К. А. Бочков, С. Н. Харлап, Б. В. Сивко // Автоматика на транспорте: ПГУПС. – 2016. – № 1 , т. 2. – С. 47–64.
4. Сивко, Б. В. Diverse Axiomatic Basis Checker: свидетельство о регистрации компьютерной программы в Национальном центре интеллектуальной собственности Республики Беларусь № 996 / Б. В. Сивко. – Опубл. 13.12.2017.
5. Сивко, Б. В. Автоматизация процесса оценки степени диверситета аппаратно-программных комплексов / Б. В. Сивко // Проблемы безопасности на транспорте : материалы VIII Междунар. науч.-практ. конф., посвящ. Году науки: в 2 ч. Ч. 1, Гомель, 23–24 ноября 2017 г. / БелГУТ; под общ. ред. Ю. И. Кулаженко [и др.]. – Гомель, 2017. – С. 199–200.
6. Сивко, Б. В. Обнаружение отказов на основе доступности адресных данных / Сивко Б.В. // Информационные технологии и системы: материалы междунар. науч. конф., Минск, 26 октября 2016 г. / БГУИР; гл. ред. Л. Ю. Шилин [и др.]. – Минск, 2016. – С. 66–67.
7. Сивко, Б. В. Address Detection: свидетельство о регистрации компьютерной программы в Национальном центре интеллектуальной собственности Республики Беларусь № 983 / Б. В. Сивко. – Опубл. 23.10.2017.
8. Бочков, К. А. Автоматизация метода обнаружения отказов на основе доступности адресных данных / К. А. Бочков, С. Н. Харлап, Б. В. Сивко // Проблемы безопасности на транспорте : материалы VIII Междунар. науч.-практ. конф., посвящ. Году науки: в 2 ч. Ч. 1, Гомель, 23–24 ноября 2017 г. / БелГУТ; под общ. ред. Ю. И. Кулаженко [и др.]. – Гомель, 2017. – С. 183–184.