

5 (96)' 2012

**ІНФОРМАЦІЙНО -КЕРУЮЧІ
СИСТЕМИ НА ЗАЛІЗНИЧНОМУ
ТРАНСПОРТІ**

Виходить 6 разів на рік
Видається з 23 квітня 1996 р.

**INFORMACIJO-KERUÛCI SISTEMI
NA ZALIZNICNOMU TRANSPORTI**

Зміст – Содержание – Contents

Видання

Державної адміністрації
залізниць України

Української державної
академії залізничного
транспорту

Міжнародна видавнича рада

Басов В. І. (Україна)
Бочков К.А. (Білорусь)
Данько М.І. (Україна)
Загарій Г.І. (Україна)
Зубко А.П. (Україна)
Jiang Xin Hua (China)
Кравцов Ю.О. (Росія)
Негрей В.Я. (Білорусь)
Остапчук В.М. (Україна)
Сапожніков Вал.В. (Росія)
Соболев Ю.В. (Україна)
Шепко Н.А. (Україна)

*В номере опубліковані матеріали виступлений
участников 25-й Международной конференции
"Перспективные компьютерные, управляющие и
телекоммуникационные системы для железных дорог
Украины"
(г. Алушта, Крым, Украина, 24-29 сентября 2012 г.)*

**Бочаров А. П., Науменко П. П., Карбивский Ф. А.,
Шиш В. А.**

Разработка и внедрение корпоративной
информационной системы: основные положения и
проблемы3

Скобцов Ю. А., Скобцов В. Ю., Нассер ИЯД К. М.

Генерація тестов для несправностей
«індуцирована задержка» со многими
агрессорами9

Буцько Т. В., Прохорченко А. В., Музикіна С. І.

Формування моделі оперативного управління
процесом просування вагонів з небезпечними
вантажами в підсистемі
“технічна станція – прилегла дільниця”
на базі нечіткої ситуаційної мережі13

Яцько С. І.

Методи зниження витрат електроенергії шляхом
узгодження режимів роботи споживачів
електроенергії моторвагонного рухомого складу17

Рибальченко Л. І. Підходи до реалізації системи підтримки прийняття рішення оперативного управління за допомогою сучасних технологій моделювання.....	20
Дербунович Л. В., Малиновский М. Л., Караман Д. Г., Мирошник М. А., Осипенко А. Н. Метод проектирования строго безопасных автоматов локомотивной сигнализации	25
Ващишин Л. В., Нічога В. О., Сторож І. В. Штучні нейронні мережі, як засіб для розпізнавання дефектів залізничних рейок	34
Казанский Н. А., Кашин Д. И. Оценка качества передачи в каналах современных волоконно-оптических систем связи	38
Шкиль А. С., Сыревич Е. Е., Альмадхоун С. Локализация ошибок проектирования в эффективно-декомпозированной HDL-модели	42
Алешин Г. В., Левтеров А. И., Ярута А. Н. Оценка эффективности многоэтапного метода маршрутизации	46
Блиндюк В. С. Обобщенная машинная модель вагона электропоезда.....	51
Жуковицкий И. В. Цифровая система управления торможением отцепы замедлителями тормозной позиции с дискретным измерителем скорости	62
Сафронов В. В. Метод решения нечетко заданных задач многокритериального ранжирования для управления железнодорожным транспортом и проектирования его подсистем.....	67
Твердохлебов В. А. Задача размещения маршрутов движения по сети железных дорог.....	71
Кустов В.Ф., Каменєв О.Ю. Удосконалення методів математичного моделювання випробувальних засобів систем МПЦ на основі топологічних матриць	75
Кочин А.Е. Оптическая система контроля геометрических параметров заготовок железнодорожных колес в технологическом потоке производства	84

УДК 681.326

ШКИЛЬ А. С., к.т.н., доцент,
СЫРЕВИЧ Е. Е., к.т.н., доцент,
АЛЬМАДХОУН С., аспирант (ХНУРЭ)

Локализация ошибок проектирования в эффективно-декомпозированной HDL-модели

Анализ предметной области и постановка задачи

Верификация цифровых проектов, то есть аппаратных или встроенных аппаратно-программных систем, выполненных на языке описания аппаратуры (Hardware Description Language – HDL), является важной задачей в процессе проектирования цифровых устройств (ЦУ). Первоначально HDL предназначался для моделирования (что и объясняет его большую универсальность), но позднее из него было выделено синтезируемое подмножество. Написание алгоритмической модели в синтезируемом подмножестве гарантирует автоматический синтез по этой модели некоторой схемной реализации. Главная особенность HDL – параллелизм и наличие абстракций типа «сигнал» в описании. В модели на языках программирования все операторы выполняются друг за другом (не учитывая переходы и вызовы функций). Таким образом, модели на языках описания аппаратуры, с одной стороны, выглядят и ведут себя как код на языках программирования (например, при использовании подпрограмм или переменных); с другой стороны, обладают рядом кардинальных отличий (сигналы, параллелизм, синтезируемость). Отсюда невозможно тестировать и диагностировать HDL-модели исключительно методами верификации ПО или методами диагностирования аппаратуры [1].

Тестовое диагностирование аппаратной реализации ЦУ подразумевает наличие списка контрольных точек (КТ), тестовых значений и эталонов в КТ. Под диагностированием HDL – модели понимают поиск (локализацию) ошибок проектирования. В этой фазе стоит задача выявить место и, по возможности, вид ошибки проектирования.

При диагностировании HDL-моделей выделяют два типа КТ, используемых при поиске места ошибки. КТ первого рода – сигнал (переменная) модели, эталонные значения которых известны из спецификации. КТ второго рода – сигнал (переменная) модели, значе-

ния которых наблюдаемы, но до начала диагностического эксперимента неизвестны [2]. В данной работе рассмотрено нахождение эталонов для КТ второго рода в HDL-моделях (т.е. КТ, в которых эталоны неизвестны, но легко предсказуемы).

Цель работы: повышение глубины локализации ошибки проектирования в HDL-моделях. Данная цель достигается путем решения двух задач, а именно:

- декомпозиции (явной – с выделением структурно и функционально законченного модуля на этапе реализации системы, неявной – с помощью средств анализа в специализированных САПР);
- доискивания в декомпозированных модулях (применение ad hoc процедур для каждого типа модуля).

Признаки эффективной декомпозиции HDL-кода

Типы описаний устройств можно условно разделить на простые и сложные. К простым описаниям относятся описания с интуитивно понятным (тригонометрические функции и т.д.), логичным (например, операции чтения/записи) или совпадающим с аппаратной реализацией законом функционирования (регистры, счетчики и т.д.). В большинстве случаев легко можно восстановить структуру устройства (например, автомат, описанный с применением стандартных шаблонов, или логическая функция). Сложные описания представляют собой комбинацию простых, но восстановить визуальную структуру и/или тип (функцию) не представляется возможным.

По аналогии с тестопригодным проектированием определим, что системой, пригодной к локализации ошибок проектирования, является та, части которой реализованы по простому типу описания.

Критерии качества программного обеспечения (имея ввиду дуальность HDL-модели) определяются несколькими параметрами, в том числе и логически обоснованной декомпозицией. С точки зрения бесклас-

сового кода (каким является HDL-модель), реализующего вычислительные задачи, наилучшей декомпозицией является разбиение модуля на логически и функционально законченные блоки-функции.

Г. Буч [3] приводит правило Клеменса и Вейса, которым активно пользуются разработчики программного обеспечения при выделении модулей программ: «Особенности системы, подверженные изменениям, следует скрывать в отдельных модулях; в качестве межмодульных можно использовать только те элементы, вероятность изменения которых мала». Также необходимо отметить, что поскольку модули служат элементарными и неделимыми блоками программы, которые могут использоваться повторно, это должно учитываться при распределении классов и объектов по модулям.

Введем понятие эффективно-декомпозированной системы. Учитывая дуальность системы на HDL, к ней можно предъявить те же требования, что и к программному коду. Эффективно-декомпозированной системой является та, у которой логически законченные операции над внутренними или входными словами вынесены в отдельные модули. Таким образом, каждый модуль будет являться законченной функцией. Если это невозможно организовать структурно, то такая декомпозиция должна быть организована с применением иных средств, например, с помощью детального комментирования.

Декомпозиция может быть явной и неявной. Явно декомпозированной является та система, которая была спроектирована и реализована с учетом требований к верификации и локализации. Неявно декомпозированной является та, которой можно применить специализированную САПР для поиска фрагментов, удовлетворяющих определению простых описаний.

Помимо работоспособности модуля, при явной декомпозиции его алгоритм должен быть понятен не только разработчику, а и «непосвященному» человеку, если не с первого, то со второго раза.

Критерии неявной декомпозиции зависят от выбранной САПР и сочетают общепринятые правила, сходные с правилами на синтез схемных реализаций по заданному HDL-описанию, с особенностями данной САПР. Так например, в качестве примера САПРа для декомпозиции можно привести системы Debussy и Verdi компании SpringSoft, которые выделяют схемные элементы без оптимизации, автоматы, «черные ящики» - функции без аппаратного эквивалента. И что немаловажно, есть возможность кросс-анализа (визуальное отображение фрагмента кода, эквивалентного схемной реализации). Следует отметить, что стандартные системы синтеза не пригодны для декомпозиции, т.к. в процессе логического вывода код подвергается оптимизации и кросс-анализ неинформативен или недоступен.

Декомпозиция, во-первых, сокращает число оши-

бок проектирования благодаря визуальному контролю распознанных блоков (при неявной декомпозиции) и более детальной проработке каждого модуля (при явной). Во-вторых, декомпозированный код на порядок повышает эффективность диагностических процедур (процедур локализации) за счет добавления КТ с предсказуемыми эталонами. Система из полностью «черного ящика» превращается в «белый» или «серый».

Таким образом, в общем виде задача локализации ошибок проектирования решается следующим образом:

- Необходимо выполнить неявную декомпозицию системы, если явная декомпозиция не была выполнена на этапе реализации.
- Сформировать три группы модулей (фрагментов): эквивалентных аппаратным примитивам (триггеры, мультиплексоры, цепи логических элементов), автоматные модули (преимущественно FSM или FSMD, но с незначительной вычислительной частью), «черные ящики» (часто это функции, преобразование которых в схемные элементы без оптимизации невозможно).
- Применить к каждой группе модулей отдельную процедуру локализации (доискивания).

Доискивание в простых описаниях

Декомпозиция на таком уровне (это может быть триггер, регистры, мультиплексоры, логические элементы) позволяет применить ранее разработанные процедуры на основе функционального, структурного методов или их производных.

Выделение шаблонов описания автоматов позволяет провести диагностический эксперимент над автоматами (проверка достижимости с использованием линий сброса, обход всех дуг, обход всех вершин) [4].

Т.к. в простых описаниях, совпадающих с аппаратной реализацией, легко восстановить соответствие между словесным описанием и схемным эквивалентом, то диагностирование неисправности в примитиве будет соответствовать локализации ошибки проектирования.

Локализация ошибок проектирования в описаниях функций

Декомпозиция (явная или неявная) дает возможность применить методы тестирования ПО и методы мат. анализа для модулей (или фрагментов кода), реализующих непрерывные функции. Такие декомпозированные модули могут иметь или не иметь разветвления, однако наличие вызовов функций внутри модуля потребует дополнительных действий при верификации.

Введем понятие области возможных значений (ОВЗ) как множества значений, которые может принимать переменная, и которые ограничены ее типом и

размерностью.

Утверждение. Если функция, реализуемая модулем, непрерывна на ОВЗ входных слов, то в пределах ОВЗ модуль ведет себя однозначно и в соответствии с вычисленной областью значений.

Доказательство. Приведем теорему Больцано-Коши и ее обобщение. Если функция f непрерывна на $[a, b]$ и число C лежит между $f(a)$ и $f(b)$, то существует такая точка $x \in [a, b]$, что $f(x) = C$. Всякая непрерывная функция f , определенная на множестве вещественных чисел и принимающая какие-либо два значения, принимает и любое лежащее между ними.

Большинству типов данных языков описания аппаратуры можно поставить в соответствие вещественные числа алгебры. Тогда перепишем обобщение теоремы Больцано-Коши для таких типов данных. Всякая непрерывная функция f , определенная на R - множестве значений, которые могут принимать входные слова, и принимающая какие-либо два значения, принимает и любое лежащее между ними. Таким образом, отсутствует необходимость проверять модуль, реализующий непрерывную алгебраическую функцию, на всей области возможных значений.

Следствие. Нет жестких требований к количеству точек при проверке описания непрерывной функции.

Другими словами, к таким модулям достаточно применить подход, взятый из теории тестирования ПО, а именно: выделение классов эквивалентных данных и областей граничных значений. При необходимости можно дополнить тест значениями из спецификации, пожеланий заказчика либо разработчика и т.д. Однако следует оговорить, что это справедливо только для алгоритмов, представленных непрерывной функцией.

Исходя из вышесказанного, можно сделать следующий вывод. В условиях эффективно-декомпозированного кода нахождение эталонов сводится к определению функции, реализуемой блоком, и ОВЗ входных слов. Эталоны могут быть получены для любого набора из класса эквивалентных данных.

Получение эталонов для декомпозированных модулей позволит сузить область поиска ошибок проектирования до одного модуля. Если такие модули объединены в сеть, то задача решается применением структурно-функционального подхода, описанного в [1]. При этом вместо графа связей сигналов (между операторами) достаточно восстановить структуру связей между модулями внутри системы. Однако функции, представленные «черным ящиком», не позволяют увеличить глубину локализации ошибки проектирования.

Выводы

Рассмотренная проблема диагностирования (локализации) ошибок проектирования при верификации описаний сложных цифровых устройств на HDL является актуальной, т.к. с повышением сложности ЦУ невозможно обеспечить максимальную глубину локализации без наложения определенных ограничений на верифицируемую систему. Эти ограничения предполагают применение явной декомпозиции на этапе создания системы таким образом, чтобы модули системы являлись логически законченными описаниями, имеющими аппаратные эквиваленты. При отсутствии явной декомпозиции необходимо использовать соответствующие САПР уже на этапе верификации. В зависимости от типа модулей к ним применяется один из известных способов локализации ошибок проектирования. При невозможности дальнейшей декомпозиции фрагмент считается «черным ящиком», и если он реализует непрерывную функцию, то для него предлагается адаптация и практическое применение теоремы Больцано-Коши. В целом предложенная стратегия позволит увеличить глубину локализации до оператора (в аппаратно-эквивалентных модулях) или до модуля (в модулях - «черных ящиках»).

Литература

1. Шкиль А.С., Сыревич Е.Е., Альмадхоун С. Модели описаний цифровых устройств для диагностирования // Вестник Херсонского государственного технического университета. – Херсон: ХГТУ – 2010. – №2 (38). – С. 258-265.
2. Альмадхоун С., Шкиль А.С., Сыревич Е.Е., Кучеренко Д.Е. Метод обратного прослеживания для поиска ошибок проектирования в HDL-коде // Радиоэлектроника. Информатика. Управление. – Запорожье ЗНТУ. – 2009. – №2. – С. 86-90
3. Слюсаренко И.М., Слюсаренко М.Ю. Системологический подход к декомпозиции в объектно-ориентированном анализе и проектировании программного обеспечения [Электронный ресурс] / CITForum – Режим доступа: [www/URL: http://citforum.ru/programming/case/ood_systemology/](http://citforum.ru/programming/case/ood_systemology/) – 04.07.2012. – Загл. с экрана.
4. Хаханов В.И., Шкиль А.С., Ковалев Е.В. Модели цифровых автоматов для проектирования тестов в среде Activt-HDL // Радиоэлектроника и информатика. – 2000. – № 2. – С. 86-92.

Резюме

В статье рассматривается проблема диагностирования ошибок проектирования при верификации описаний сложных цифровых устройств на HDL. Предлагается подход к диагностированию эффективно-декомпозированных моделей ЦУ на HDL. Предлагается адаптация и практическое применение теоремы Больцано-Коши для описаний, реализующих непрерывные функции

У статті розглянуто проблему діагностування помилок проектування під час верифікації описів складних цифрових систем на HDL. Запропоновано підхід до діагностування ефективно-декомпозованих моделей ЦП на HDL. Запропоновано адаптацію та практичне застосування теореми Больцано-Коші для описів, які реалізують неперервні функції

A problem of design fault localization during complex digital system HDL-model verification is considered in the article. An approach is proposed for diagnosing effectively-decomposed models in HDL. An adaptation and practical usage of Boltsano-Koshi theorem is proposed for descriptions that implement continuous functions

Ключові слова: языки описания аппаратуры, программное обеспечение, ошибки проектирования, верификация, диагностирование, декомпозиция

Рецензент д.т.н., профессор Дербунович Л. В. (НТУ "ХПИ")

Поступила 28.05.2012 г.