

МАТЕРИАЛЫ
VI Международной молодежной
научной конференции
«МАТЕМАТИЧЕСКОЕ
И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
ИНФОРМАЦИОННЫХ,
ТЕХНИЧЕСКИХ
И ЭКОНОМИЧЕСКИХ СИСТЕМ»

Томск, 24–26 мая 2018 г.

*Под общей редакцией
кандидата технических наук И.С. Шмырина*

Томск
Издательский Дом Томского государственного университета
2018

становятся недоступными для включения вредоносных подсхем, оставшиеся линии маскируются с помощью перепрограммируемых LUT. Разработана программа, реализующая описанные подходы к покрытию «подозрительных» линий. В дальнейшем предполагается проведение компьютерных экспериментов для контрольных примеров реальных логических схем.

ЛИТЕРАТУРА

1. *Матросова А.Ю., Останин С.А., Николаева Е.А.* Синтез частично программируемых схем, ориентированных на маскирование вредоносных подсхем (Trojan Circuits) // Труды Института системного программирования РАН. – 2017. – Т. 29. – № 5. – С. 61–74.
2. *Матросова А.Ю., Томков В.В.* Использование точных оценок управляемости и наблюдаемости для выявления вредоносных подсхем (Trojan Circuits) в последовательностных схемах // Известия вузов. Физика. – 2016. – Т. 59. – № 8/2. – С. 68–71.
3. *Yamashita S., Yoshida H., Fujita M.* Increasing yield using partially-programmable circuits // Workshop on Synthesis And System Integration of Mixed Information technologies (SASIMI). – 2010. – P. 237–242.
4. *Jo S., Matsumoto T., Fujita M.* SAT-based automatic rectification and debugging of combinational circuits with LUT insertions // Proc. Of IEEE Asian Test Symposium. – 2012. – P. 19–24.

ПОСТРОЕНИЕ ВСЕХ ТЕСТОВЫХ НАБОРОВ, ОБНАРУЖИВАЮЩИХ КОНСТАНТНЫЕ НЕИСПРАВНОСТИ ЛИТЕРАТУРЫ ЭНФ

М.И. Броч

Томский государственный университет
mishabroch@gmail.com

Введение

Для увеличения быстродействия схем высокого уровня интеграции необходимо обнаружение их ложных путей. Ложными путями в схеме являются пути, для которых изменение значения сигнала на входе пути не приводит к изменению значения сигнала на выходе схемы (комбинационной составляющей схемы с памятью). Ложные пути могут оказаться среди путей, характеризующихся самыми большими задержками. Эти задержки вычисляются по характеристикам элементов, составляющих путь. Самые большие задержки путей в схеме определяют её быстродействие: время между синхросигналами в схеме с памятью немного больше самой большой из задержек путей. Исключение ложных путей может привести к сокращению времени между синхросигналами и, следовательно, к повышению быстродействия схемы. В [1] проблема поиска ложных путей в схеме с памятью сведена к нахождению всех тестовых наборов, обнаруживающих константные неисправности литер эквивалентной нормальной формы (ЭНФ), сопоставляемых рассматриваемым путям. ЭНФ строится по комбинационной составляющей схемы с памятью. Далее находится последовательность, обеспечивающая доставку одного из тестовых наборов для константных неисправностей исследуемого пути. В данной работе ЭНФ компактно представляется И-ИЛИ деревом. По нему строится множество конъюнкций, порождающих тестовые наборы, которое затем представляется Reduced Ordered Binary Decision Diagram (ROBDD-графом). Затем выполняется перемножение полученного графа с ROBDD-графом схемы. Разработаны программы, реализующие алгоритм нахождения всех тестовых наборов для константной неисправности литеры ЭНФ.

1. Представление схемы размеченным И-ИЛИ деревом

Ранее было показано [2], что все пути схемы представляются размеченным И-ИЛИ деревом, причём каждой концевой вершине дерева сопоставляется единственный путь в схеме. Задана комбинационная схема (рис. 1):

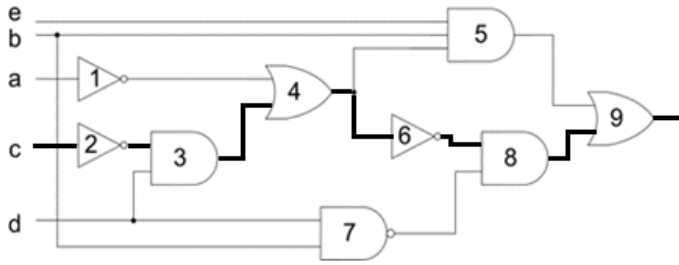


Рис. 1. Комбинационная схема

Представим её в виде BLIF (Berkeley Logic Interchange Format) – представление данных, целью которого является описание иерархической (комбинационной) схемы логического уровня в текстовой форме. Схема может рассматриваться как ориентированный граф комбинационных логических узлов и последовательных логических элементов. Автором написана программа, которая, используя такое представление данных, строит размеченное И-ИЛИ дерево. Далее в размеченном И-ИЛИ дереве находится конечная вершина, соответствующая заданному пути в схеме, с использованием алгоритма, представленного в [2].

На рис. 2 комбинационная схема рис. 1 представлена размеченным деревом с выделенной вершиной, сопоставляемой одному из путей схемы.

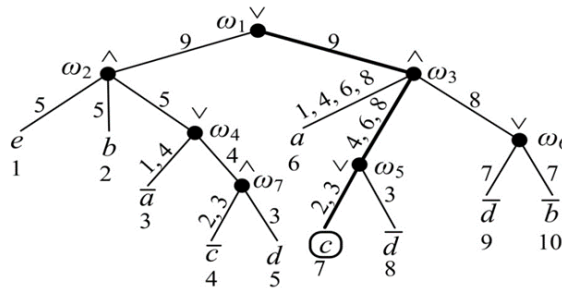


Рис. 2. Размеченное дерево для схемы рис. 1

2. Коррекция И,ИЛИ дерева относительно выделенной конечной вершины

После того, как в размеченном дереве найдена интересующая нас конечная вершина, переходим к построению всех тестовых наборов для b_p неисправности литеры, сопоставляемой рассматриваемому пути. Под b_p неисправностью понимается замена константой 1 каждого вхождения литеры ЭНФ, сопоставляемой рассматриваемому пути. Теперь нам нет необходимости использовать размеченное дерево, можно убрать разметки его рёбер и использовать неразмеченное И-ИЛИ дерево для дальнейших вычислений.

Далее неразмеченное (размеченное) дерево преобразуем таким образом, что выделенная вершина рассматриваемого пути оказывается крайней левой. Поясним необходимую последовательность действий на примере дерева рис. 2.

Вершина c является крайней левой относительно ближайшей к ней неконцевой вершине ω_5 . Переходим к соседней, более близкой к корню, неконцевой вершине ω_3 . Ребро, связывающее обе вершины, сделаем крайним левым вместе с инцидентным ему поддеревом. Наконец, ребро, связывающее вершины ω_3 , ω_1 вместе с инцидентным ему поддеревом сделаем крайним левым. В результате получаем дерево рис. 3.

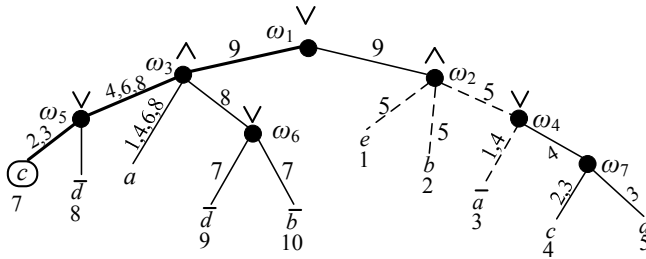


Рис. 3. Скорректированное размеченное дерево схемы рис. 1

Выделим существенное поддерево в И-ИЛИ дереве. Двигаемся от корня к конечным вершинам. Если очередная проходима вершина помечена символом \wedge , то включаем в существенное поддерево все исходящие из вершины рёбра, если символом ∇ – включаем одно исходящее ребро. На рис. 3 существенное поддерево выделено пунктиром. Жирной линией отмечен путь с вершиной c . Существенные поддеревья можно строить как для неразмеченных, так и размеченных И-ИЛИ деревьев. Нас интересуют конъюнкции, содержащие литеру, сопоставляемую рассматриваемому пути, и принадлежащие множеству конъюнкций, порождающих все тестовые наборы для b_p неисправности литеры.

3. Построение всех тестовых наборов

Тестовый набор, как известно, обеспечивает распространение смены значения сигнала в присутствии неисправности от места её возникновения до выхода схемы. В нашем случае – от входа схемы вдоль заданного пути.

Пусть K – найденная по дереву конъюнкция; она может быть пустой. Будем говорить, что K расширяема по литере x_i^α (здесь i – индекс входной переменной, α – путь, представляемый последовательностью номеров составляющих его элементов), если выбрасывание из неё этой литеры приводит к образованию непустой конъюнкции K^* , являющейся импликантой функции. Иначе она не расширяема по этой литере.

Будем рассматривать K^* как результат склеивания по литере x_i^α конъюнкций K и \underline{K} , в т.ч. и для пустой конъюнкции K . Здесь \underline{K} получается из K заменой литеры x_i^α на инверсную литеру. Будем называть \underline{K} дополнением K по литере x_i^α или просто дополнением K .

Построение всех тестовых наборов для b_p неисправности

Из неразмеченного И-ИЛИ дерева сокращённым перебором существенных поддеревьев, предложенным в [3], извлекаем все конъюнкции, порождающие тестовые наборы для b_p неисправности литеры. С этой целью корректируем процедуру отбрасывания частично построенных конъюнкций.

Не достраиваем частично построенные конъюнкции, содержащие повторяющиеся переменные, одна из которых отмечает начало рассматриваемого пути. Повторяющиеся считаются переменные, знаки инверсий которых совпадают.

Не достраиваем частично построенные конъюнкции, содержащие взаимно инверсные переменные, отличающиеся от переменной, отмечающей начало рассматриваемого пути.

Из множества конъюнкций, порождающих все тестовые наборы для b_p неисправности литеры формируем множество конъюнкций типа \underline{K} , в том числе и из подходящих пустых конъюнкций.

Каждая из конъюнкций типа \underline{K} приводится к нормальному виду исключением повторяющихся литер. По полученному множеству элементарных конъюнкций строится ROBDD-граф $R(\underline{K})$.

По И-ИЛИ дереву строится ROBDD-граф $R(C)$ исследуемой схемы.

Инверсная функция схемы C представляется графом $\overline{R(C)}$, который получается из $R(C)$ перестановкой терминальных вершин. Перемножив $R(\underline{K})$ и $\overline{R(C)}$, получаем ROBDD-граф $R(b_p)$, компактно представляющий все тестовые наборы для b_p неисправности.

Построение всех тестовых наборов для a_p неисправности

Построение всех тестовых наборов для a_p неисправности сводится к построению всех тестовых наборов для b_p неисправности инверсного И-ИЛИ дерева. Инверсное И-ИЛИ дерево получается из описанного ранее дерева заменой разметок внутренних вершин и знаков инверсий концевых вершин. Символ \vee заменяется символом \wedge , символ \wedge заменяется символом \vee .

Перемножив соответствующие $R(\underline{K})$ и $R(C)$ (вместо $\overline{R(C)}$, здесь используем $R(C)$), получаем ROBDD-граф $R(a_p)$, компактно представляющий все тестовые наборы для a_p неисправности.

Заключение

Реализованы программы, позволяющие представить комбинационную схему (Benchmarks), описанную в BLIF формате, в виде размеченного И-ИЛИ дерева, находить нужную концевую вершину в дереве, соответствующую заданному пути, отыскивать все конъюнкции, которые могут порождать b_p неисправности литеры. Далее, используя CUDD (пакет для осуществления операций над BDD-графами), строим ROBDD-графы и при их перемножении получаем множество всех тестовых наборов для константных неисправностей литеры.

ЛИТЕРАТУРА

1. Матросова А.Ю., Андреева В.В., Чернышов С.В., Рожкова С.В., Кудин Д.В. Обнаружение ложных путей в последовательностных схемах // Известия вузов. Физика. – 2017. – Т. 60. – № 10. – С. 170–178.
2. Матросова А.Ю., Останин С.А., Сингх В. Обнаружение несущественных путей логических схем на основе совместного анализа и-или деревьев и SSBDD-графов // Автоматика и телемеханика. – 2013. – № 9. – С. 126–142.
3. Матросова А.Ю. Алгоритмические методы синтеза тестов. – Томск, Издательство Томского университета, 1990. – 206 стр.

ИССЛЕДОВАНИЕ И РАЗРАБОТКА АЛГОРИТМОВ УПРОЩЕНИЯ КНФ ПРИ РЕШЕНИИ ПРОБЛЕМЫ ВЫПОЛНИМОСТИ

Н.Н. Пашкова, В.В. Андреева

Томский государственный университет
neldnelaps@gmail.com, avv.21@mail.ru

Введение

Сложность проектирования сверхбольших интегральных схем (СБИС) значительно возросла за последние годы. Большая часть времени теперь тратится на разработку тестов, которые обеспечивают быстрое обнаружение неисправностей, возникших в процессе производства схемы.

Методы тестирования разделяются на встроенные и внешние. Наиболее перспективным представителем встроенного метода является Built-In Self-Test (BIST) или