

Д.В. Тельпухов¹, В.В. Надоленко², С.И. Гуров³

ВЫЧИСЛЕНИЕ НАБЛЮДАЕМОСТИ ЭЛЕМЕНТОВ КОМБИНАЦИОННЫХ ЛОГИЧЕСКИХ СХЕМ С ИСПОЛЬЗОВАНИЕМ БИТ-ПАРАЛЛЕЛЬНОГО МОДЕЛИРОВАНИЯ

Введение

Наблюдаемость элементов комбинационной схемы характеризует наиболее сильный [1] и сложный для оценки [2] механизм маскирования ошибок – логический. Данный механизм срабатывает тогда, когда сбой происходит на ненаблюдаемом элементе G , т.е. сигналы на первичных выходах схемы не зависят от сигнала на выходе элемента G . В общем случае логическое маскирование зависит от состояния схемы (значений сигналов на всех ее узлах). В свою очередь, состояние комбинационной схемы при условии ее корректного функционирования однозначно определяется набором входных сигналов. Поэтому наблюдаемость элемента G (т.е. вероятность несрабатывания логического механизма маскирования при инверсной ошибке на данном элементе) определяется формулой:

$$P_{O_G} = \sum_i P_{X_i} * O_G^i, \quad (1)$$

где X_i – i -й входной вектор, O_G^i – наблюдаемость элемента G при данном входном векторе (0 или 1), суммирование ведется по всем входным векторам.

Рассчитав наблюдаемости всех элементов по формуле (1), можно затем оценить сбоеустойчивость всей схемы [3].

Бит-параллельное моделирование

В данной работе используется бит-параллельное моделирование – метод Монте-Карло, позволяющий хранить и обрабатывать одновременно множество состояний схемы [4]. Вместо одного логического значения каждому узлу цепи присваивается сигнатура:

$$signature_G = [f_G(X_1)f_G(X_2) \dots f_G(X_N)], \quad (2)$$

где X_1, \dots, X_N – входные наборы (вектора значений сигналов на входах схемы), f_G – логическая функция узла.

¹ Институт проблем проектирования в микроэлектронике РАН, рук. отдела, к.т.н.

² Институт проблем проектирования в микроэлектронике РАН, инженер-проектировщик.

³ Факультет ВМК МГУ им. М. В. Ломоносова, доцент, к.ф.-м.н.

Сначала в соответствии с входными векторами задаются сигнатуры первичных входов схемы. Для этого составляется матрица из векторов-столбцов X_1, \dots, X_N , строками которой являются сигнатуры соответствующих входов:

$$\begin{array}{cccccc}
 & X_1 & X_2 & \cdots & X_N & \\
 \text{signature}_{in1} & x_{11} & x_{21} & \cdots & x_{N1} & \\
 \text{signature}_{in2} & x_{12} & x_{22} & \cdots & x_{N2} & \\
 \vdots & \vdots & \vdots & \ddots & \vdots & \\
 \text{signature}_{inK} & x_{1K} & x_{2K} & \cdots & x_{NK} &
 \end{array} \quad (3)$$

Затем выполняется обход элементов схемы в топологическом порядке. Сигнатура выхода элемента вычисляется побитовым применением его логической функции к сигнатурам входов. Таким образом, для каждого входного вектора производится полное логическое моделирование схемы.

Далее для каждого элемента G при каждом входном векторе X_i определяются O_G^i (см. (1) и формируется вектор наблюдаемостей – маска ODC (observability don't care) [5]:

$$ODC_G = [O_G^1 O_G^2 \dots O_G^N] \quad (4)$$

При равномерном распределении вероятностей появления входных векторов X_1, \dots, X_N из (1) следует:

$$P_{O_G} = \frac{w(ODC_G)}{|ODC_G|}, \quad (5)$$

где $w(ODC_G)$ – вес Хемминга булева вектора ODC_G , т.е. количество единиц в нем, $|ODC_G|$ – длина данного вектора.

Вычисление масок ODC

В данной работе рассматриваются 5 методов вычисления масок ODC: моделирование инверсной ошибки [6], обратное распространение [5], ускоренная симуляция ошибки [7], модульный подход [8, 9] и частичное моделирование.

Первый метод является эталонным, поскольку в нем используется само определение наблюдаемости для ее вычисления. Вторым методом отличается малыми временными затратами, однако его погрешность сильно зависит от структуры схемы и в присутствии реконвергентных путей может достигать 3-5%, а при наличии в схеме функционально избыточных частей – 10% и более. Остальные методы по-разному комбинируют моделирование сбоя и обратное распространение. Ускоренная симуляция дает точный результат, однако требует меньше времени, чем полное моделирование. Модульный подход и частичное моделирование направлены на достижение компромисса между скоростью и точностью.

Далее подробно рассматривается каждый метод, его преимущества и недостатки.

Моделирование инверсной ошибки

Данный метод заключается в повторном логическом моделировании схемы для каждого элемента G . Сигнатура элемента G при этом полностью инвертируется, и в его выходном конусе распространяется ошибка. Таким образом, на каждой итерации на выходах схемы появляется дополнительный набор ошибочных сигнатур. Сравнивая его с основным набором, полученным при эталонном моделировании схемы, можно определить маску ODC на выходе элемента G :

$$ODC_G = \cup_j [sign_{out_j} \wedge sign_{out_j}^{E_G}], \quad (6)$$

где $sign_{out_j}$ – эталонная сигнатура на выходе схемы out_j , $sign_{out_j}^{E_G}$ – сигнатура на том же выходе, полученная при моделировании с инверсной ошибкой на элементе G , знак \wedge обозначает побитовое исключающее или).

Существенным недостатком данного метода является квадратичная временная сложность относительно количества элементов. Несмотря на то, что при моделировании инверсной ошибки на элементе достаточно пересчитывать сигнатуры только в его выходном конусе (для схем ISCAS'85 объем вычислений при этом сокращается в среднем в 5 раз), время выполнения алгоритма на схемах с количеством элементов больше 1000 на порядки превышает приемлемые значения [7].

Обратное распространение

Метод обратного распространения заключается в последовательной передаче масок ODC от выходов схемы к ее входам. В его основе лежат два действия: передача маски ODC через узел (от приемников сигнала к источнику) и через элемент (от выхода к входам).

Передача ODC от приемников сигнала к источнику выполняется с помощью побитовой дизъюнкции:

$$ODC_S = \cup_i ODC_{R_i}, \quad (7)$$

где S – источник сигнала, R_i – i -й приемник.

Для передачи маски ODC от выхода Y некоторого логического элемента к одному из его входов X_k необходимо сначала вычислить локальную маску ODC данного входа – последовательность значений относительной наблюдаемости $O_{X_k \rightarrow Y}$:

$$LODC_{X_k} = [O_{X_k \rightarrow Y}^1 \ O_{X_k \rightarrow Y}^2 \ \dots \ O_{X_k \rightarrow Y}^N], \quad (8)$$

$O_{X_k \rightarrow Y}$ принимает значение 1, если переключение входа X_k в данном состоянии схемы приведет к переключению выхода Y , и 0 в противном случае.

Глобальная маска ODC_{X_k} затем может быть вычислена по формуле:

$$ODC_{X_k} = LODC_{X_k} \cap ODC_Y, \quad (9)$$

Стоит отметить, что относительная наблюдаемость $O_{X_k \rightarrow Y}$ является функцией остальных входов элемента, и $LODC_{X_k}$ можно получить побитовым применением данной функции к соответствующим сигнатурам. Расчет масок ODC всех узлов в схеме сводится к $(N_{pin} - N_{PO})$ побитовым вычислениям соответствующих булевых функций, где N_{pin} – общее количество контактов у всех элементов схемы, N_{PO} – количество первичных выходов. Количество логических операций в формуле (7) напрямую зависит от разветвления по выходу соответствующего элемента и, как правило, не превышает 10. Сложность формулы (9) в общем случае зависит от количества входов элемента, его логической функции и оптимальности составленных функций LODC. В данном исследовании использовалась библиотека стандартных ячеек NanGate 45nm Open Cell Library. Функции LODC для всех элементов были составлены таким образом, что для их вычисления необходимо произвести не более 15 логических операций.

Как видно из представленных рассуждений, временную сложность алгоритма обратного распространения можно считать линейной относительно количества элементов в схеме, что дает ему значительное преимущество перед моделированием сбоев. Однако при наличии в схеме реконвергентных путей данный метод не дает точного результата.

Реконвергентными называются различные пути сигнала от одного источника к одному приемнику. На рис. 1 приведен пример схемы с реконвергентными путями (от первичного входа X2 к элементу N3) и результат применения к ней алгоритма обратного распространения.

При входном векторе [1 1 1] (последний бит сигнатур и масок ODC) инверсия на первичном входе X2 приводит к переключению обоих входов элемента N3 и ошибке на выходе Y1. Однако переключения каждого входа N3 по отдельности в данном состоянии схемы не наблюдаемы, и алгоритм обратного распространения обрабатывает их независимо, что приводит к ложному нулю в маске ODC первичного входа X2.

Для большинства схем погрешность вычисления коэффициента чувствительности с помощью метода обратного распространения масок ODC лежит в пределах 1-2%, что вполне приемлемо для грубой оценки сбоеустойчивости на логическом уровне проектирования. Однако наличие в схеме функционально избыточных участков может значительно снизить точность вычислений. Например, после тройного резервирования участка схемы наблюдаемость всех элементов внутри этого участка падает до ну-

ля. В то же время наблюдаемость остальных элементов не меняется, поскольку участок заменяется его полным функциональным эквивалентом (подробное объяснение можно найти в [10]). Однако метод обратного распространения в худшем случае присвоит нулевые маски ODC всем узлам во входном конусе мажоритарного элемента.

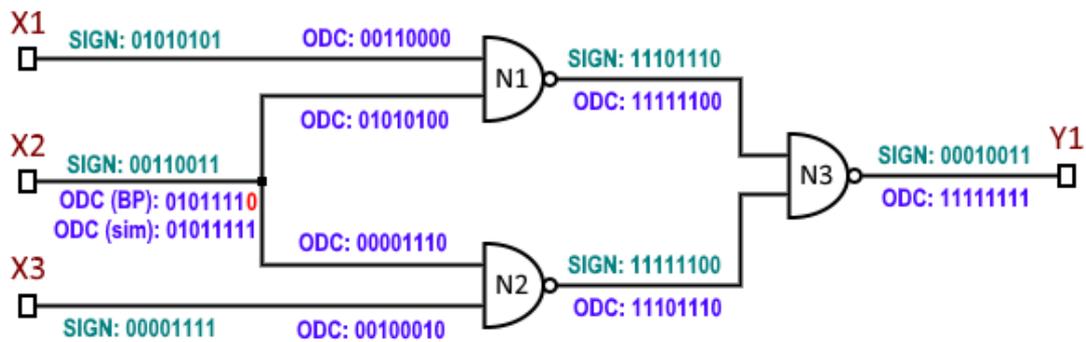


Рис. 1. Пример работы алгоритма обратного распространения при наличии реконвергентных путей. Верхняя маска ODC на входе схемы X2 получена алгоритмом обратного распространения, нижняя - симуляцией ошибки

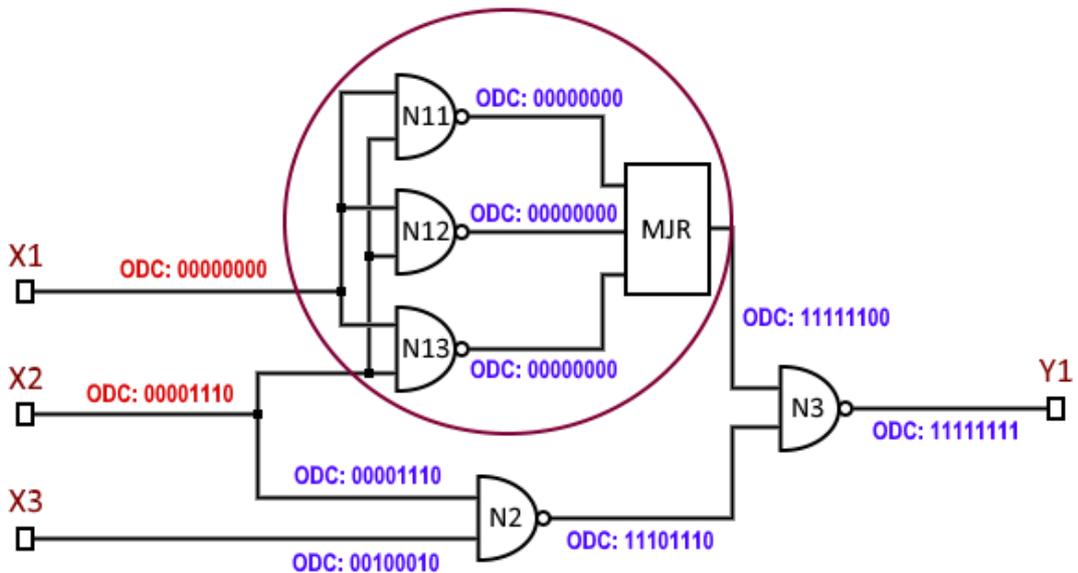


Рис. 2. Пример работы алгоритма обратного распространения масок ODC на схеме с функциональной избыточностью, MJR – мажоритарный элемент

В качестве примера приведем рассмотренную выше схему и применим тройное резервирование к элементу N1. На рис. 2 изображена результирующая схема и ее маски ODC, рассчитанные с помощью метода обратного распространения. Выделенный участок схемы (элементы N11, N12, N13, MJR) выполняет функцию элемента `pand2`, т.е. эквивалентен

элементу N11 на рис. 1. Однако маски ODC входов X1 и X2 существенно отличаются от соответствующих масок на рис. 1. Достаточно подключить к ним выходы каких-либо других модулей, чтобы коэффициент чувствительности всей схемы определялся с большой погрешностью. Поэтому не рекомендуется применять метод обратного распространения для оценки логического маскирования в схемах с большим количеством функционально избыточных участков.

Ускоренная симуляция ошибки

Как упоминалось ранее, данный метод является более быстрым аналогом полного моделирования сбоев. Ускорение достигается за счет вычисления относительных наблюдаемостей на нереконвергентных путях сигнала с помощью метода обратного распространения и дальнейшего их использования для передачи ошибки на несколько логических уровней вперед. Подробное описание алгоритма можно найти в [7]. В данной статье ограничимся примером его работы.

Рассмотрим вычисление масок ODC в схеме из трех элементов `pand2`, представленной на рис. 1. Элемент N3 в данной схеме является приемником реконвергентных сигналов, т.е. одиночный сбой в его входном конусе может привести к инверсии сигналов на обоих его входах. Поэтому сбои, проходящие через данный элемент, необходимо моделировать.

Применим сначала метод обратного распространения ко всей схеме, однако для входного конуса элемента N3 будем рассчитывать относительные наблюдаемости на его входах (см. рис. 3).

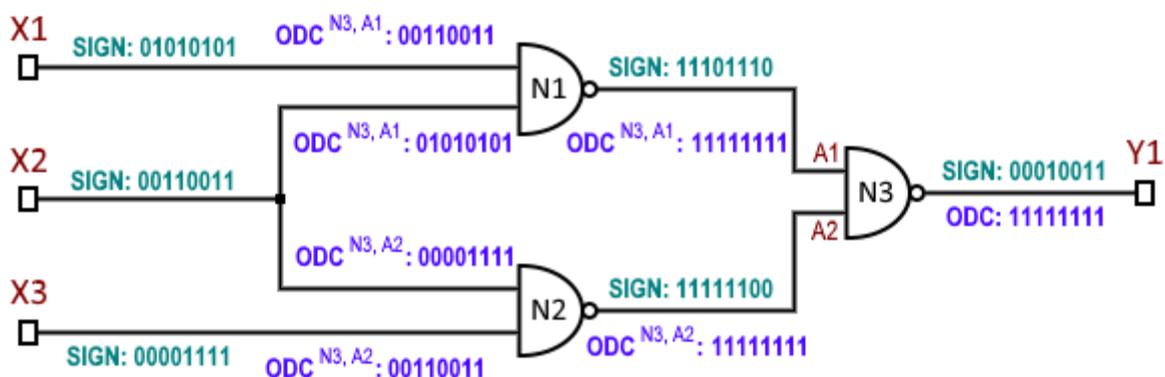


Рис. 3. Пример расчета относительных масок ODC

Теперь для любого источника инверсной ошибки во входном конусе элемента N3 можно вычислить соответствующие вектора ошибок на его входах:

$$error_R^E = error_S^E \cap ODC_S^R, \quad (10)$$

где R – приемник сигнала (в данном случае один из входов элемента N3), S – источник сигнала, $error_G^E$ – такой булев вектор для внесенной ошибки E , что:

$$sign_G^E = sign_G \wedge error_G^E, \quad (11)$$

т.е. единицы в данном векторе сигнализируют о наличии инверсии на соответствующем узле, а нули показывают ее отсутствие. Таким образом, реализуется четырехзначная логика, описанная в [11].

При симуляции сбоя на очередном узле ему присваивается вектор ошибки, заполненный единицами, а затем реализуется прямое распространение ошибок по схеме. Использование относительных масок ODC позволяет пропускать моделирование промежуточных элементов на неконвергентных путях.

Рассмотрим в качестве примера симуляцию сбоя на первичном входе X2 в представленной на рис. 3 схеме. В первую очередь вектор ошибки присваивается ее источнику: $error_{X2} = [11111111]$. Далее с использованием известных масок по формуле (10) вычисляются ошибки на входах элемента N3: $error_{N3,A1} = error_{X2} \cap ODC_{X2}^{N3,A1}$, $error_{N3,A2} = error_{X2} \cap ODC_{X2}^{N3,A2}$. По полученным векторам ошибок восстанавливаются соответствующие сигнатуры в присутствии сбоя, которые затем используются для вычисления сигнатуры выхода элемента N3. Завершает цикл определение вектора ошибок на выходе элемента N3. На рис. 4 проиллюстрировано выполнение описанного алгоритма.

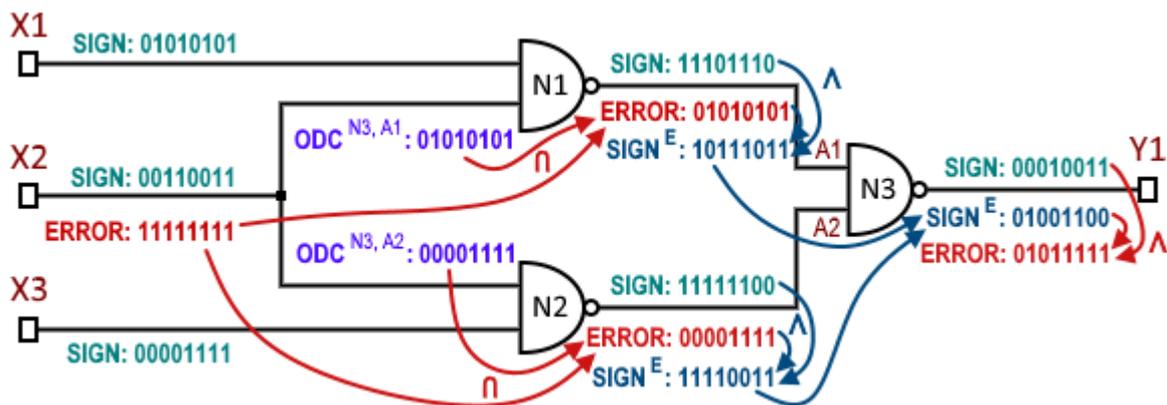


Рис. 4. Распространение вектора ошибки с использованием относительных ODC

Может показаться, что алгоритм ускоренной симуляции требует большего количества вычислений, чем простое моделирование, однако это справедливо только для простых примеров. В общем случае временные затраты алгоритма ускоренной симуляции зависят от структуры схемы, при отсутствии реконвергентных путей приближаясь к затратам алгоритма обратного распространения. Основным препятствием для его ис-

пользования является тот факт, что количество реконвергентных путей в схемах, как правило, достаточно велико. В таблице 1 представлено количество источников и приемников реконвергентных сигналов в схемах из набора ISCAS'85, синтезированных в Synopsys Design Compiler с оптимизацией по задержке на критическом пути. Ранее те же схемы были использованы при тестировании алгоритма ускоренной симуляции и сравнении его с полным моделированием. На большинстве из них первый метод оказался быстрее в 1.5-3 раза [7]. Из таблицы 1 видно, что число элементов, участвующих в ускоренной симуляции, и общее число элементов для большинства представленных схем, независимо от их размера, имеют один порядок. Поэтому на практике временные затраты алгоритма ускоренной симуляции растут квадратично с увеличением размеров схемы. Таким образом, область применения точных методов ограничивается небольшими схемами (до 1000 элементов).

| Схема | Количество элементов | Количество источников реконвергентных сигналов | Количество приемников реконвергентных сигналов |
|-------|----------------------|--|--|
| C17 | 6 | 0 | 0 |
| C432 | 228 | 94 | 79 |
| C1355 | 385 | 66 | 95 |
| C499 | 474 | 110 | 140 |
| C1908 | 503 | 159 | 176 |
| C2670 | 508 | 116 | 154 |
| C5315 | 1039 | 203 | 367 |
| C3540 | 1187 | 424 | 549 |
| C7552 | 1536 | 514 | 808 |
| C6288 | 3149 | 1707 | 1760 |

Таблица 1. Количество источников и приемников реконвергентных сигналов в схемах из набора ISCAS'85

Модульный подход

Низкая масштабируемость точных методов и недостаточная надежность метода обратного распространения обусловили необходимость разработки новых алгоритмов вычисления наблюдаемости.

В [8] предлагается при оценке сбоеустойчивости использовать существующую иерархическую структуру схемы, применяя точный метод для модулей и допуская погрешность на верхнем уровне. В [9] предлагаются небольшие участки схемы с реконвергентными путями рассматривать как единые блоки, т.е. создавать иерархическую структуру автоматически. Стоит отметить, что второй подход более универсален. Авторы [8] опираются на принцип декомпозиции в проектировании схем, ожидая

четкого разделения их на функциональные блоки. Однако при синтезе схемы и ее оптимизации с использованием какой-либо САПР создается плоская структура, и поиск в ней эквивалентных участков – отдельная трудоемкая задача. Поэтому будем считать, что исходная схема не содержит иерархических блоков (далее – ИБ). В этом случае алгоритму оценки должен предшествовать этап выделения подсхем с реконвергентными путями (далее – этап кластеризации).

На этапе кластеризации все участки схемы, удовлетворяющие заданным условиям, представляются в виде ИБ. В данном исследовании были установлены следующие ограничения:

- каждая подсхема содержит ровно один приемник реконвергентных сигналов, источником которых является один из входов подсхемы;
- количество логических уровней в подсхеме (длина наибольшего реконвергентного пути) не больше L ;
- общее количество входов подсхемы не больше K .

Величины L и K являются параметрами, регулирующими время работы и точность алгоритма. На рис. 5 приведен пример подсхемы с соответствующими обозначениями.

Процедуру выделения подсхем можно выполнять несколько раз, создавая многоуровневую структуру. Так, если повторно применить алгоритм кластеризации с параметрами $L \geq 3$, $K \geq 5$ к схеме на рис. 5 (б), элементы N10, N11, N22 и S1 образуют новую подсхему. Следует понимать, что данный пример лишь демонстрирует возможности алгоритма и на практике не является оптимальным решением.

За этапом кластеризации следует этап вычислений. В контексте бит-параллельного моделирования вычисляются маски ODC всех узлов. На верхнем уровне иерархии применяется метод обратного распространения. Для передачи масок от выходов ИБ к его входам используется сокращенная симуляция.

Данный метод позволяет частично учесть короткие реконвергентные пути, где корреляция сигналов может внести значительную погрешность. Однако у него есть один существенный недостаток. Создание иерархической структуры приводит к ограничению области поиска, поскольку ИБ не должны пересекаться. В рассмотренном выше примере элемент N16 лежит в двух разных реконвергентных путях: N11 – N23 и N3 – N22. Учесть влияние обеих реконвергенций в этом случае можно только добавлением еще одного уровня иерархии, что усложняет структуру ИБ. Кроме того, необходимо отслеживать и предотвращать появление ложных обратных связей на верхнем уровне, чтобы алгоритм обратного распространения работал корректно. Пример образования ложной обратной связи представлен на рис. 6: один из входов подсхемы S1 принимает

сигнал с выхода подсхемы S2, и один из входов S2 подключен к выходу S1, хотя исходная схема является комбинационной.

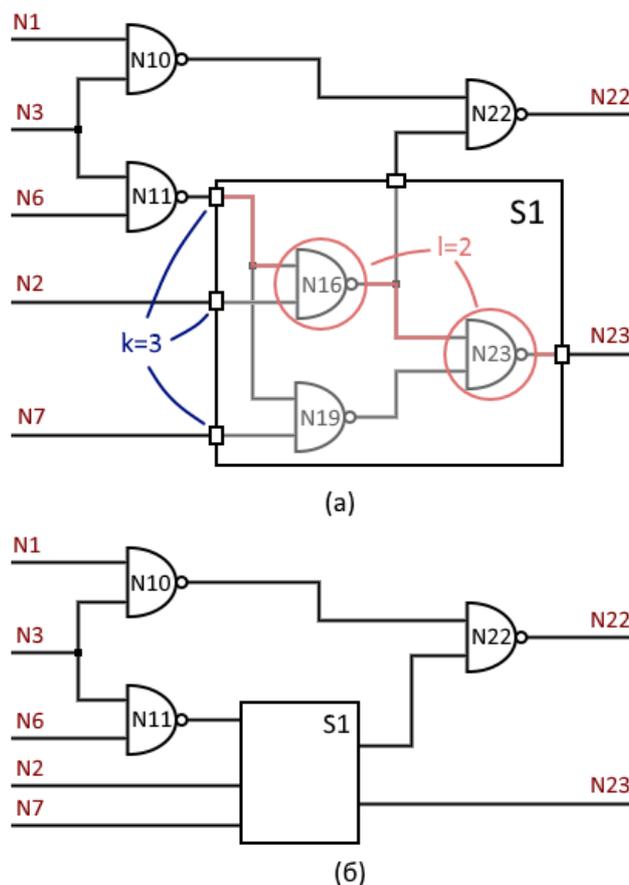


Рис. 5. Схема с17 с подсхемой, содержащей реконвергентные пути (а) и ее представление на верхнем уровне иерархии (б), l – количество логических уровней в подсхеме, k – количество входов подсхемы

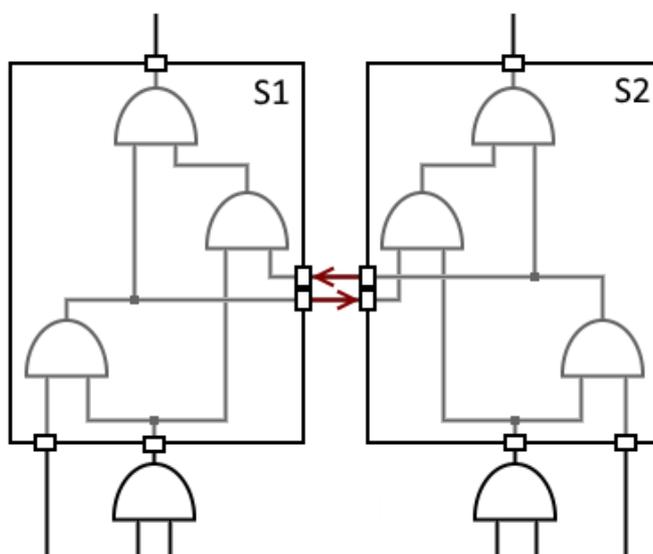


Рис. 6. Пример образования ложной обратной связи при выделении под-схем

Частичное моделирование

В [8] предлагается также применять моделирование ошибки, ограничивая область ее распространения эвристически (например, определенным количеством логических уровней). Результатом такого моделирования является множество вторичных источников и соответствующих векторов ошибок. Считая их независимыми, можно с помощью формулы (10) вычислить вектора ошибок на первичных выходах схемы, а затем применить к последним побитовую дизъюнкцию.

Данный метод, как и предыдущий, позволяет учесть короткие реконвергентные пути, не влияя при этом на структуру схемы. Дальнейшее повышение его эффективности в настоящем исследовании получено включением в область распространения ошибки только реконвергентных путей.

Моделирование с ограничением по количеству логических уровней, как и модульный подход, позволяют в некоторой степени управлять скоростью и качеством работы алгоритма, смещая баланс в ту или иную сторону в соответствии с текущими требованиями. Однако оценить точность полученного результата можно только посредством сравнения с эталоном, поскольку погрешность зависит как от параметров алгоритма, так и от структуры схемы.

В рамках данного исследования также был разработан и протестирован алгоритм, позволяющий оценить погрешность «сверху» при частичном моделировании. Суть его заключается в том, что на каждом шаге алгоритма распространения ошибки анализируется взаимное влияние вторичных источников. Реконвергенция сигналов может привести к неточности вычисления наблюдаемости только в том случае, если выходные конусы источников ошибки пересекаются. Поэтому для каждого состояния схемы (набора значений сигналов на всех узлах) верно следующее утверждение:

Если существует хотя бы один независимый источник ошибки (такой, что его выходной конус не пересекается с выходными конусами других источников), и его наблюдаемость равна 1, то ошибка будет распространена до первичных выходов схемы.

Таким образом, существуют два условия, при которых возможна передача вектора ошибки со вторичного источника на выходы схемы (формула 10) без потери точности: независимость данного источника и существование другого источника, одновременно независимого и наблю-

даемого. Если ни одно из условий не выполняется, в вычислениях может быть допущена ошибка.

В контексте бит-параллельного моделирования была введена маска погрешности – булев вектор с единицами в позициях, где сбой на элементе мог быть неверно передан на выходы схемы. Очередной вторичный источник ошибки рассматривался как независимый в том случае, если это не приводило к превышению допустимого количества единиц в маске погрешности. Такой подход позволил обеспечить заданную точность вычисления масок ОДС, однако оказался непрактичным. Описанный алгоритм, как правило, дает сильно завышенную оценку погрешности, поэтому при допусках не более 10% область моделирования ошибки для большинства источников совпадает с их выходными конусами.

Модель эксперимента и результаты

Для сравнения представленных методов были использованы 10 схем из набора ISCAS'85, оптимизированных по задержке на критическом пути. Также был сформирован дополнительный набор из 5 схем, содержащих множество функционально избыточных (мажорированных) участков. Далее будем использовать обозначения «обычные схемы» и «схемы с избыточностью», соответственно.

Для каждой схемы проводилось логическое моделирование со случайными входными сигнатурами длины 10 000, а затем по очереди применялись алгоритмы вычисления масок ОДС. В качестве характеристики быстродействия алгоритма была выбрана относительная разность времени его работы с временем, затраченным на обратное распространение. Точность полученных масок ОДС характеризуется двумя величинами: погрешностью значения сбоеустойчивости, вычисленного с их использованием, и долей ошибочных бит в самих масках. В качестве метрики сбоеустойчивости использовалась «чувствительная площадь»:

$$SA = \sum_G P_{O_G} * A_G, \quad (12)$$

где P_{O_G} – вероятность логического маскирования сбоя на элементе G (формула (1)), A_G – площадь элемента G .

Эталоном точности выбран метод сокращенной симуляции, поскольку ранее было показано, что полученные с его помощью маски ОДС не содержат ошибок. Стоит отметить, однако, что входные воздействия для схемы генерируются случайно, поэтому значения вероятностей маскирования по формуле (1) вычисляются с некоторой погрешностью, присущей всем Монте-Карло методам. Величина этой погрешности имеет нормальное распределение, параметры которого зависят от длины сигнатур и структуры схемы, но не от используемого метода вычисления масок

ODC. В проведенных экспериментах погрешность, связанная с неполным перебором состояний схемы, составила не более 1% с вероятностью 99%.

Метод частичного моделирования применялся с различными значениями параметра L – максимального количества логических уровней в области распространения ошибки. Основным параметром метода кластеризации в модульном подходе было количество итераций I . Число входов подсхемы было ограничено согласно эмпирической формуле: $K = 4 + 3 * I$. Ограничение на количество логических уровней L на первой итерации принималось равным 4, а на каждой последующей увеличивалось на 2. Для метода моделирования с заданной точностью параметром являлась допустимая погрешность E в процентах.

| Метод | | $\langle \Delta t_{BP} \rangle, \%$ | $\langle \Delta SA \rangle, \%$ | $\langle \Delta ODC \rangle, \%$ |
|---|------------|-------------------------------------|---------------------------------|----------------------------------|
| Сокращенная симуляция | | 2521,5 | 0,000 | 0,000 |
| Обратное распространение | | 0,0 | 1,414 | 0,770 |
| Модульный подход | $I = 1$ | 37,6 | 0,899 | 0,555 |
| | $I = 2$ | 62,3 | 0,975 | 0,561 |
| | $I = 3$ | 79,1 | 0,732 | 0,474 |
| | $I = 4$ | 156,4 | 0,702 | 0,453 |
| Частичное моделирование (ограничение по логическим уровням) | $L = 3$ | 40,5 | 0,589 | 0,443 |
| | $L = 4$ | 110,6 | 0,435 | 0,371 |
| | $L = 5$ | 227,0 | 0,391 | 0,298 |
| | $L = 6$ | 354,4 | 0,399 | 0,260 |
| Частичное моделирование (ограничение по допусти- мой погрешности) | $E = 2\%$ | 2624,9 | 0,004 | 0,002 |
| | $E = 5\%$ | 2586,9 | 0,009 | 0,006 |
| | $E = 10\%$ | 2558,4 | 0,012 | 0,007 |
| | $E = 20\%$ | 2531,6 | 0,028 | 0,017 |

Таблица 2. Эффективность алгоритмов вычисления масок ODC, средние значения для 10 схем из набора ISCAS'85

| Метод | | $\langle \Delta t_{BP} \rangle, \%$ | $\langle \Delta SA \rangle, \%$ | $\langle \Delta ODC \rangle, \%$ |
|---|---------|-------------------------------------|---------------------------------|----------------------------------|
| Сокращенная симуляция | | 1992,1 | 0,000 | 0,000 |
| Обратное распространение | | 0,0 | 29,002 | 4,266 |
| Модульный подход | $I = 1$ | 36,2 | 28,814 | 4,227 |
| | $I = 2$ | 70,1 | 19,853 | 3,140 |
| | $I = 3$ | 93,8 | 13,989 | 2,322 |
| | $I = 4$ | 126,6 | 8,433 | 1,630 |
| | $I = 5$ | 194,4 | 5,476 | 1,212 |
| | $I = 6$ | 318,4 | 4,095 | 1,089 |
| | $I = 7$ | 485,3 | 2,184 | 0,739 |
| Частичное моделирование (ограничение по логиче- ским уровням) | $L = 2$ | 7,5 | 28,548 | 4,171 |
| | $L = 3$ | 56,9 | 23,916 | 3,539 |
| | $L = 4$ | 142,0 | 16,714 | 2,539 |
| | $L = 5$ | 257,0 | 10,276 | 1,644 |

| | | | | |
|--|---------|--------|-------|-------|
| | L = 6 | 401,8 | 6,504 | 1,169 |
| | L = 7 | 560,7 | 3,831 | 0,799 |
| | L = 8 | 722,2 | 2,740 | 0,608 |
| Частичное моделирование (ограничение по допустимой погрешности) | E = 2% | 2091,1 | 0,625 | 0,084 |
| | E = 5% | 2056,4 | 0,877 | 0,130 |
| | E = 10% | 2045,4 | 2,036 | 0,299 |
| | E = 20% | 2033,5 | 3,068 | 0,500 |
| | E = 50% | 1894,1 | 6,159 | 1,020 |

Таблица 3. Эффективность алгоритмов вычисления масок ODC, средние значения для 5 схем с избыточностью

Таблица 2 содержит результаты тестирования представленных алгоритмов на обычных схемах, таблица 3 – на схемах с избыточностью. $\langle \Delta t_{BP} \rangle$ означает среднюю разницу в затраченном времени с алгоритмом обратного распространения, $\langle \Delta SA \rangle$ – среднюю погрешность «чувствительной площади», $\langle \Delta ODC \rangle$ – среднюю долю ошибочных бит в масках ODC.

Как упоминалось ранее, оценка возможной погрешности «сверху» при частичном моделировании не дала практически значимых результатов: время работы алгоритма слабо зависит от параметра и в большинстве случаев превышает даже время сокращенной симуляции. Ограничение на количество логических уровней и модульный подход, напротив, позволяют управлять скоростью и точностью алгоритма. Чтобы сравнить их эффективность, представим данные таблиц 2 и 3 в виде точек на координатной плоскости «время-погрешность» (рис. 7, 8). Координата X соответствует $\langle \Delta t_{BP} \rangle$ в таблице, Y соответствует $\langle \Delta ODC \rangle$. На рис. 7 представлены данные тестирования алгоритмов на обычных схемах. Модульный подход для них менее эффективен и нестабилен: увеличение количества итераций не всегда приводит к повышению точности. Однако на схемах с большим количеством функционально избыточных участков (рис. 8) картина меняется. За несколько итераций алгоритм кластеризации захватывает большую часть мажоритарных структур, устраняя основной источник погрешности. Временные затраты при этом растут медленно, что обеспечивает высокую эффективность модульного подхода.

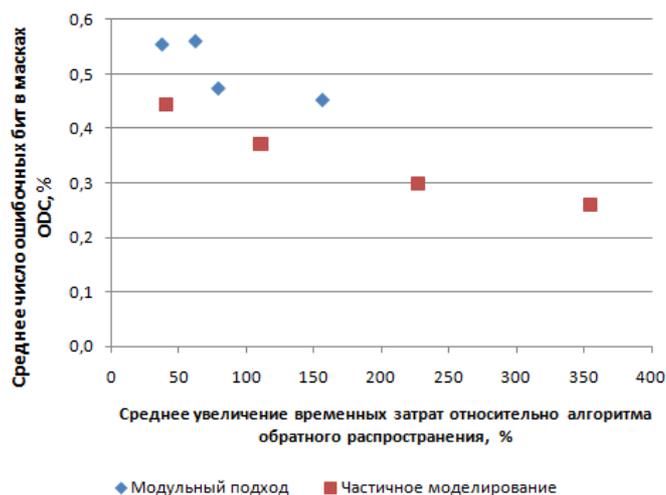


Рис. 7. Эффективность модульного подхода и частичного моделирования на обычных схемах.

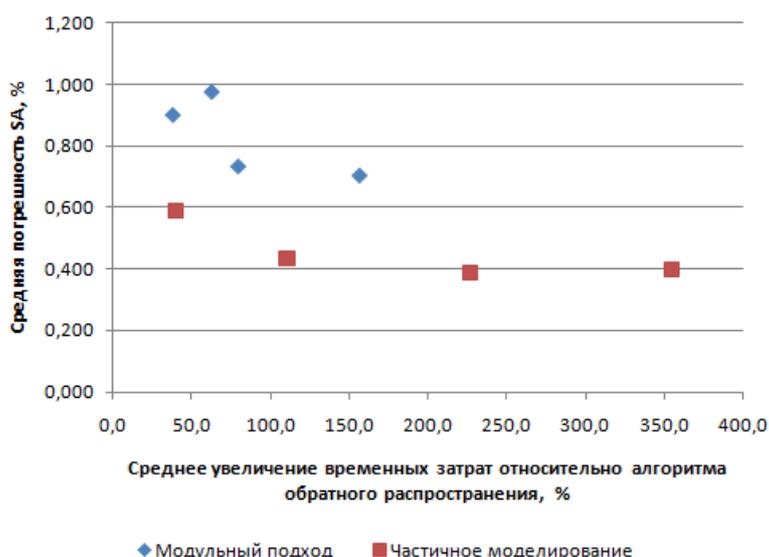


Рис. 8. Эффективность модульного подхода и частичного моделирования на схемах с избыточностью

Таким образом, наиболее универсальным и надежным является метод частичного моделирования с ограничением на количество логических уровней. Но если схема содержит локальные структуры с многочисленными реконвергентными путями, оказывающими значительное влияние на точность обратного распространения, модульный подход при правильном подборе параметров кластеризации может оказаться более эффективным.

Заключение

В данной статье рассмотрены различные методы вычисления наблюдаемостей элементов комбинационной логической схемы в контексте бит-параллельного моделирования. Для каждого метода описаны его дос-

тоинства и недостатки, а также возможная область применения. Приведенные рассуждения и рекомендации по использованию методов подкреплены результатами тестов с использованием схем из набора ISCAS'85, оптимизированных по задержке на критическом пути, а также сгенерированных на их основе схем с большим количеством функционально избыточных участков. Дальнейшие исследования в этой области могут быть направлены на разработку эвристических методов подбора параметров частичного моделирования и кластеризации (при модульном подходе) в соответствии со структурой схемы. Кроме того, нерешенной остается проблема обеспечения заданной точности вычислений.

Литература

1. *Asadi, H., Tahoori, M. B., Fazeli, M., Miremadi, S.G.* Efficient algorithms to accurately compute derating factors of digital circuits // *Microelectron Reliab.* 2012. No. 52(6). P. 1215-1226.

2. *George, N., Lach, J.* Characterization of logical masking and error propagation in combinational circuits and effects on system vulnerability // *Dependable Systems Networks (DSN), 2011 IEEE/IFIP 41st Intern. Conf.* 2011. P. 323-334.

3. *Стемковский А. Л., Тельпухов Д. В., Соловьев Р. А., Мячиков М. В., Тельпухова Н. В.* Разработка технологически-независимых метрик для оценки маскирующих свойств логических схем // *Вычислительные технологии.* 2016. Т. 21. № 2. С. 53-62.

4. *Стемковский А. Л., Тельпухов Д. В., Соловьев Р. А., Мячиков М. В.* Методы повышения производительности вычислений при расчёте метрик надёжности комбинационных логических схем // *Вычислительные технологии.* 2016. Т. 21. № 6. С. 104-112.

5. *Smita Krishnaswamy, Stephen M. Plaza, Igor L. Markov, and John P. Hayes.* Enhancing Design Robustness with Reliability-aware Resynthesis and Logic Simulation // *IWLS, 2007.*

6. *Стемковский А. Л., Соловьев Р. А., Тельпухов Д. В., Соловьев А. Н., Мячиков М. В.* Моделирование возникновения неисправностей для оценки надёжностных характеристик логических схем // *Информационные технологии.* 2014. № 11. С. 30-36.

7. *Stempkovskiy A. L., Telpukhov D. V., Nadolenko V. V.* Fast and Accurate Back Propagation Method for Reliability Evaluation of Logic Circuits // *2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus).*

8. *J. Han, H. Chen, E. Boykin, and J. Fortes.* Reliability evaluation of logic circuits using probabilistic gate models // *Microelectronics Reliability.* 2011. No. 51(2). P. 468-476.

9. *Mohyuddin, N., Pakbaznia, E., Pedram, M.* Probabilistic error propagation in logic circuits using the boolean difference calculus // In: Proceedings of 26th International Conference on Computer Design (ICCD). 2008. P. 7-13.

10. *Надоленко В. В., Тельпухов Д. В., Битков Ю. В.* Разработка маршрута ресинтеза комбинационных логических схем с целью повышения мас-кирующих свойств // VIII Всероссийская научно-техническая конференция «Проблемы разработки перспективных микро- и нано-электронных систем – 2018»: сб. научн. тр. - М.: ИППМ РАН, 2018

11. *Fazeli, M., Ahmadian, S. N., Miremadi, S. G., Asadi, H., Tahoori, M. B.* Soft error rate estimation of digital circuits in the presence of Multiple Event Transients (METs) // 2011 Design, Automation & Test in Europe. P. 1-6.