

УДК 681.32

Ю.А. СКОБЦОВ¹, В.Ю. СКОБЦОВ², ИЯД К.М. НАССЕР¹¹Донецкий национальный технический университет, Украина²Институт прикладной математики и механики НАН Украины, Украина

ГЕНЕРАЦИЯ ПРОВЕРЯЮЩИХ ТЕСТОВ ДЛЯ ИНДУЦИРОВАННЫХ ЗАДЕРЖЕК

Рассматривается проблема построения проверяющих тестов для «перекрестных» неисправностей типа задержка, характерных для глубокого субмикронного проектирования элементной базы современных компьютерных систем. При решении этой задачи используется многозначное логическое моделирование и генетический алгоритм генерации проверяющих тестов для этих неисправностей. Показано, что применение генетических алгоритмов в сочетании с логическим моделированием в многозначном алфавите позволяет эффективно решать задачу построения тестов для неисправностей индуцированных задержек.

Ключевые слова: перекрестные неисправности, многозначный алфавит, проверяющие тесты, генетический алгоритм.

Введение

Диагностика современных компьютерных систем требует анализа не только классических константных неисправностей, но и более сложных моделей, которые учитывают временные характеристики схемы. Прежде всего это относится к глубокому субмикронному проектированию.

Увеличение числа транзисторов на кристалле ведет к тому, что большее количество элементов переключаются одновременно, что может уменьшить для них уровень напряжения и увеличить задержки распространения сигналов. При этом некоторые пересекающиеся линии, которые предполагались электрически изолированными, могут взаимодействовать друг с другом. Одно из подобных взаимодействий, вызванное паразитической емкостной связью между проводниками, называется “crosstalk” (перекрестная помеха), может привести к функциональным проблемам и ухудшить временные характеристики.

1. Перекрестные (crosstalk) неисправности

Обычно рассматриваются два основных типа перекрестных помех: 1) “crosstalk” индуцированные импульсы; 2) “crosstalk” индуцированные задержки. В первом случае быстро переключающаяся линия - «агрессор» (aggressor) может индуцировать короткий импульс на статической линии - «жертве» (victim). В зависимости от амплитуды и ширины эти импульсы могут оказывать существенное влияние на характеристики схемы. Неисправности первого типа “crosstalk glitch” - индуцированные импульсы

вызываются наводкой между близко расположенными проводниками, которые имеют несбалансированные значения драйвера и нагрузки [1,2].

Второй случай (индуцированные задержки) имеет место тогда, когда на линиях «агрессоре» и «жертве» происходят (почти) одновременные переходы сигналов. Если на обеих линиях переходы в одном направлении, то время перехода сокращается и, следовательно, уменьшается время задержки распространения сигналов. Этот эффект называется «перекрестным ускорением» (“crosstalk speedup”) [3]. Сильный «агрессор» может вызвать задержку распространения сигнала на линии-«жертве», которая имеет противоположное значение сигнала. Если на линиях «агрессоре» и «жертве» происходят переходы сигналов в противоположных направлениях, то время перехода увеличивается и имеет место эффект «перекрестного замедления» (“crosstalk slowdown”). Если вызванный шум на линии-«жертве» больше порогового напряжения или индуцированная задержка больше допустимой, то это может привести к логическим отказам или функциональным проблемам на соседних триггерах или выходах.

2. Многозначный алфавит S_9

В процессе построения проверяющего теста для повышения эффективности целесообразно использовать многозначные алфавиты, которые позволяют описывать ситуации, возникающие в процессе моделирования и генерации тестов [4]. В таблице 1 представлены символы 9-значного алфавита S_9 и их физическая интерпретация для перекрестных неисправностей.

Отметим, что переменные внешних входов при тестировании рассматриваемой неисправности могут принимать только четыре значения “статический 0” S0, “статический 1” S1, передний фронт R и задний фронт F. Внутренние линии схемы кроме приведенных выше четырех значений могут принимать следующие: PG (положительный импульс), NG (отрицательный импульс), DR (увеличение задержки); DF (уменьшение задержки), U – неопределенное значение.

Таблица 1

Многозначный алфавит C_9

Символ C_9	Интерпретация	Двоичные значения		
		t-1	t	t+1
S0	Статический 0	0	0	0
S1	Статическая 1	1	1	1
RT	Передний фронт	0	1	1
FT	Задний фронт	1	0	0
PG	Положительный импульс	0	1	0
NG	Отрицательный импульс	1	0	1
DR	Увеличение задержки	0	0	1
DF	Уменьшение задержки	1	1	0
U	Неопределенность			

Из табл. 1 видно, что по сути исходные сигналы в каждом такте являются двоичными. Но с вычислительной точки зрения удобнее и эффективнее работать с символами многозначного алфавита, которые позволяют обрабатывать одновременно несколько тактов. Для этого необходимо определить функционирование логических элементов в данном многозначном алфавите.

Таблица 2

Табличное задание вентиля И в алфавите C_9

И	S0	S1	R	F	PG	NG	DR	DF	U
S0									
S1	S0	S1	R	F	PG	NG	DR	DF	U
R	S0	R	R	S0	PG	DR	DR	PG	U
F	S0	F	S0	F	S0	F	S0	F	U
PG	S0	PG	PG	S0	PG	U	S0	S0	U
NG	S0	NG	DR	F	S0	NG	DR	F	U
DR	S0	DR	DR	S0	S0	DR	DR	S0	U
DF	S0	DF	PG	F	PG	F	S0	S0	U
U	S0	u	F0	G0	u	u	G0	F0	U

Логические функции в многозначном алфавите обычно описываются табличным способом. Например, для вентиля И табличная модель в алфавите C_9 представлены в табл. 2 и получена на основании физического смысла символов алфавита C_9 и логики функционирования элементов.

Из табл. 1 видно, что исходные сигналы в каждом такте являются двоичными, поэтому девятизначный алфавит C_9 (как и многие другие [4]) принадлежит декартовому произведению двоичных алфавитов $C_9 \subseteq B_2 \times B_2 \times B_2$

3. Постановка задачи

При построении проверяющего теста для такой неисправности необходимо:

1) найти входные наборы, которые вызывают необходимый переход сигналов на линии-агрессоре:

2) найти входные наборы, обеспечивающие необходимый переход в противоположном направлении на линии-«жертве» и распространение возникшей индуцированной задержки от жертвы до одного из внешних входов.

Пусть состояния линии-жертвы g и линии-агрессора h описываются двумя булевыми функциями от переменных, связанных с внешними входами, $g(x_1, x_2, \dots, x_n)$ и $h(x_1, x_2, \dots, x_n)$.

Аналогично на каждом внешнем выходе реализуется булева функция $f_j(x_1, x_2, \dots, x_n, g, h)$, которая зависит как от внешних переменных (x_1, x_2, \dots, x_n) , так и от внутренних переменных g, h .

Для проверки перекрестной неисправности индуцированной задержки необходимы такие значения внешних входов, которые обеспечивают на линии-агрессоре передний фронт R, а на линии-жертве задний фронт F либо наоборот, на линии-агрессоре задний фронт F, а на линии-жертве передний фронт R.

Пусть состояния линии-жертвы g и линии-агрессора h описываются двумя булевыми функциями от переменных, связанных с внешними входами, $g(x_1, x_2, \dots, x_n)$ и $h(x_1, x_2, \dots, x_n)$.

Аналогично на каждом внешнем выходе реализуется булева функция $f_j(x_1, x_2, \dots, x_n, g, h)$, которая зависит как от внешних переменных (x_1, x_2, \dots, x_n) , так и от внутренних переменных g, h , где $j=1, 2, \dots, m$ и x_i может принимать значения s0 “статический 0”, s1 “статическая 1”, R и F.

Тогда проблема построения теста для перекрестной неисправности индуцированной задержки (заднего фронта) сводится к решению следующей системы логических уравнений:

$$G(x_1, x_1, \dots, x_n) = F, \quad (1)$$

$$H(x_1, x_1, \dots, x_n) = R, \quad (2)$$

$$F_j(x_1, x_1, \dots, x_n, g, h) = DR, \quad (3)$$

где решение (1) обеспечивает значение F на линии-жертве, (2) – значение R на линии-агрессоре и (3) обеспечивает распространение влияния неисправности от линии-жертвы до одного из внешних входов. Отметим, что из этих условий следует, проверяющим тестом может быть не один входной набор, а, по крайней мере, пара наборов, которая обеспечивает фронт изменение сигнала на линии-агрессоре. Если система не имеет решения, то данная неисправность является не обнаружимой (по крайней мере, в данной постановке).

Очевидно, проблема построения проверяющего теста перекрестной неисправности задержки переднего фронта также можно свести к решению системы аналогичных логических уравнений.

Для решения этой системы можно использовать различные методы, из которых мы отдаем предпочтение генетическому алгоритму (ГА).

4. Моделирование перекрестных неисправностей типа задержка

Моделирование перекрестных неисправностей выполняется на основе событийного логического моделирования в многозначном алфавите S_9 [5]. Его целью является проверка обнаружимости перекрестных неисправностей на заданной входной последовательности. Укрупненный алгоритм моделирования неисправностей состоит из следующих шагов.

1. Моделирование исправной схемы на случайной входной последовательности.
2. Выбор неисправности индуцированной задержки из множества перекрестных неисправностей.
3. Моделирование исправной и неисправной схемы для каждого входного набора.
4. Если линии жертва и агрессор имеют различные фронты, то активизация неисправности.
5. Неисправность вносится в схему путем задержки сигнала на линии жертве на одну единицу модельного времени. Остальные события в неисправном варианте схемы такие же, как в исправной схеме.
6. Сравнение полученных в процессе моделирования значений сигналов исправной и неисправной схемы на внешних выходах схемы.
7. Если есть выход со значениями сигналов DR, то неисправность проверяется данной входной последовательностью и удаляется из множества неисправностей.
8. Пока не достигнут конец, чтение следующего входного набора и переход на п.2.

5. Генетический алгоритм генерации теста

Генетические алгоритмы широко применяются при построении проверяющих тестов для константных неисправностей [4] в силу их простоты и универсальности. На наш взгляд их применение при построении тестов для более сложных моделей неисправностей еще более оправдано, поскольку в этом случае трудно использовать детерминированные методы.

Для того, что использовать ГА при решении конкретной задачи прежде всего необходимо определить понятие особи, популяции, генетических операторов кроссинговера и мутации, фитнес-функцию, которая позволяет оценивать качество потенциального решения [5].

Ясно, что проверяющий тест для перекрестных неисправностей типа индуцированные задержки должен состоять из пар входных наборов, обеспечивающих приведенные выше условия. Здесь каждая особь P_i имеет длину $2n$ битов, где n – число внешних входов схемы. При этом каждому внешнему входу соответствует два связанных двоичных бита, которые для данного входа представляют значения сигналов на двух последовательных наборах (00), (11), (10) и (11), что в многозначном алфавите моделируется значениями S_0, S_1, R и F соответственно.

Множество пар входных наборов составляет популяцию. При поиске таких пар можно использовать ГА в сочетании с многозначным алфавитом [4]. При этом генетический алгоритм обеспечивает механизм направленного случайного поиска пар тестовых наборов, удовлетворяющих указанным условиям. Генерацию тестов для индуцированных задержек можно рассматривать как поиск в n -мерном двоичном 0-1 пространстве значений входных наборов.

При генерации новых особей мы используем турнирный метод отбора родительских особей. В этом случае все особи популяции разбиваются на подгруппы размера m с последующим выбором в каждой из них особи с лучшим значением фитнес-функции. Параметром этой процедуры является размер тура t , который принимает значения из диапазона $2 < t < N$. Используются два способа выбора: детерминированный и случайный.

При детерминированном способе выбор выполняется с вероятностью, равной 1; детерминированном способе выбор выполняется с вероятностью, равной 1.

При случайном методе выбор осуществляется с вероятностью P_t меньше 1. Чаще всего популяция разбивается на подгруппы по 2-3 особи в каждой ($m=2,3$).

Поскольку пара наборов представляется двоичной строкой, то можно использовать стандартные операторы кроссинговера и мутации [2] с некоторой модификацией, которая учитывает структуру хромосомы, где каждому входу схемы в паре входных наборов соответствует 2 связанных бита. Среди различных операторов кроссинговера мы отдаем предпочтение однородному кроссинговеру.

Напомним, что стандартный однородный кроссинговер выполняется следующим образом [5]. Здесь каждый ген потомка создается путем копирования соответствующего гена из первого или второго родителя, то есть каждая позиция потенциально является точкой кроссинговера. Для этого случайным образом генерируется двоичная маска кроссинговера той же длины (с тем числом бит), что у хромосом родителей. Четность бита маски показывает родителя, из которого копируется ген потомка. Структура хромосомы, используемая в данной задаче, требует модификации данного оператора кроссинговера. Поскольку в данном случае каждому входу схемы соответствует 2 бита двоичных значений, то логично, чтобы значения 2 бита потомков, соответствующие определенному входу, формировались из аналогичных битов этого же входа одного из родителей.

Кроме однородного, предлагается также использовать структурный кроссинговер, в котором обмен между родителями также производится столбцами, соответствующими одной древовидной подсхеме. Здесь предварительно схема должна быть разбита на древовидные подсхемы. Входы, которые «питают» одну древовидную подсхему, относятся к одной и той же группе. Здесь обмен производится группами столбцов, соответствующих одной и той же древовидной подсхеме. Отметим, что при таком подходе в одну группу попадают входы, определяющие значения внутренних «узловых» точек схемы. Прежде всего, это касается линии-жертвы, значения сигналов которой определяется только значениями внешних входов соответствующей древовидной подсхемы. В этом случае обмен производится более направленно для соответствующих внутренних линий схемы, что в некоторых случаях повышает эффективность поиска тестовых последовательностей.

Далее, как обычно, к полученным потомкам применяется оператор мутации. Здесь, как и в операторе кроссинговера, структура хромосомы пары входных наборов, используемая в данной задаче, требует модификации

классического оператора мутации. При этом случайно выбирается вход схемы (не 1 бит хромосомы!) и с небольшой вероятностью инвертируются значения двух битов, которые соответствуют выбранному входу.

Исходя из приведенных требований, мы используем в генетическом алгоритме генерации тестов фитнес-функцию для оценки каждой особи (пары входных наборов) в виде линейной комбинации трех функций

$$f = w_e \cdot f_e + w_p \cdot f_p + w_c \cdot f_c, \quad (4)$$

где: f_e оценивает способность особи возбуждать необходимое значение сигнала на линии-жертве (например, S0); f_p оценивает способность особи распространять влияние неисправности от линии-жертвы до внешних выходов; f_c оценивает способность особи учитывать влияние линий-агрессоров. Коэффициенты w_e , w_p , w_c определяются экспериментально.

Выводы

Показано, что применение ГА в сочетании с логическим моделированием в многозначном алфавите позволяет эффективно решать задачу построения тестов для неисправностей индуцированных задержек.

Литература

1. Itazaki, N. Rubio. *An approach to the analysis and detection of crosstalk faults in digital VLSI circuits [Text]* / Rubio, N. Itazaki, X. Xu, K. Kinoshita // *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*. – March 1994. – Vol.13, No.3. – P. 387 – 394.
2. Скобцов, Ю.А. *Генерация тестов для неисправностей типа индуцированные импульсы [Текст]* / Ю.А. Скобцов, В.Ю. Скобцов, Яд К.М. Нассер // *Інформаційно-керуючі системи на залізничному транспорті*. – №4. – 2010. – С. 27 – 29.
3. *Delay Testing Considering Cross-Induced Effects [Text]* / A. Krstic, J.J. Liou, Y.M. Jiang, K.T. Cheng // *Proceedings of International Conference, 2001*. – P. 233 – 238.
4. Скобцов, Ю.А. *Логическое моделирование и тестирование цифровых устройств [Текст]* / Ю.А. Скобцов, В.Ю. Скобцов. – Донецк: ИПММ НАНУ, ДонНТУ, 2005. – 436 с.
5. Скобцов, Ю.А. *Основы эволюционных вычислений [Текст]* / Ю.А. Скобцов. – Донецк: ДонНТУ, 2008. – 326 с.

Поступила в редакцию 2.03.2012

Рецензент: д-р техн. наук, проф. М.Ф. Каравай, Институт проблем управления, Москва, Россия.

**ПОБУДОВА ПЕРЕВІРЯЛЬНИХ ТЕСТІВ
ДЛЯ ІНДУКОВАНИХ ЗАТРИМОК***Ю.О. Скобцов, В.Ю. Скобцов, Іяд К.М. Насер*

Розглядається проблема побудови перевіряльнич тестів для «перехресних» несправностей типу затримка, що характерні для глибокого субмікронного проектування елементної бази сучасних комп'ютерних систем. При розв'язку поставленого завдання використовується багатозначне логічне моделювання й генетичний алгоритм генерації перевіряльнич тестів для цих несправностей. Показано, що застосування генетичних алгоритмів в поєднанні з логічним моделюванням в багатозначному алфавіті дозволяє ефективно вирішувати задачу побудови тестів для несправностей індукованих затримок.

Ключові слова: перехресні несправності, багатозначний алфавіт, перевіряльні тести, генетичний алгоритм.

**CHECK TEST GENERATION
FOR DELAY CROSSTALK FAULTS***Y.A. Skobtsov, V.Y. Skobtsov, Iyad K.M. Nasser*

The problem of checking tests generation is considered for delay crosstalk faults, which are characteristic for the deep submicron design of element base for the modern computer systems. Under a decision of this problem the multiple-valued logical simulation and genetic test generation algorithm for these faults are used. It is shown that application of genetic algorithms in a combination to logic modeling in the multiple-valued alphabet allows to solve effectively a problem of creation of tests for malfunctions of the induced delays.

Key words: crosstalk faults, multivalued alphabet, checking tests, genetic algorithm.

Скобцов Юрій Александрович – д-р техн. наук, проф., зав. кафедри АСУ Донецького національного технічного університета, Донецьк, Україна.

Скобцов Вадим Юрьевич – канд. техн. наук, доцент, зав. лабораторією дискретної математики і прикладної алгебри Інститута прикладної математики і механіки НАН України, Донецьк, Україна.

Насер Іяд К.М. – аспірант кафедри АСУ Донецького національного технічного університета, Донецьк, Україна.