

УДК 681.32

СКОБЦОВ Ю. А., д.т.н., професор,
СКОБЦОВ В. Ю., к.т.н., доцент,
НАССЕР ІЯД К. М., аспірант (ДонНТУ)

Генерація тестов для несправностей «індуцирована задержка» со многими агрессорами

Увеличение плотности микросхем на кристалле и рост числа пересечений проводящих слоев приводит к необходимости применения новых методов диагностирования для определения физических дефектов, связанных с временными характеристиками цифровых схем. К таким дефектам относятся "перекрестные несправности" (cross talk faults) [1,2].

Несправности индуцированные задержки

В настоящей работе рассматриваются перекрестные несправности типа индуцированные задержки, которые имеет место тогда, когда на линиях «агрессоре» и «жертве» происходят (почти) одновременные переходы сигналов [1]. В этом случае 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. Перекрестные несправности вызываются паразитическими наводками между соседними проводящими линиями, которые в общем случае имеют индуктивные и емкостные составляющие. Для информационных линий в схемах доминируют емкостные связи.

Генерація тестов для индуциованих задержек со многими агрессорами

Рассмотрим задачу построения теста для кратных несправностей задержек, индуцированных различными линиями агрессорами. В этом случае линии-жертвы входят в некоторый путь, связывающий внешний вход

с внешним выходом схемы [2]. Множество линий-агрессоров образуют те линии схемы, которые могут воздействовать на линии-жертвы и тем самым вызвать задержку распространения сигналов на указанном пути. При решении этой задачи необходимо решить как минимум три подзадачи [2]:

- выбор множества критических путей, формирующих линии-жертвы;
- выбор множества линий-агрессоров для заданного критического пути;
- построение пары входных тестовых наборов, проверяющих индуцированные задержки для заданного пути и множества линий-агрессоров.

Рассмотрим несправность «индуцированная» задержка, которая вызывается вследствие влияния нескольких линий-агрессоров. Пример такой несправности показан на рис.1, где линия-жертва V управляется выходом вентиля G0, а возможные линии агрессоры A1, A2, A3, A4 управляются вентилями G1, G2, G2, G4 соответственно. Здесь линии-агрессоры отобраны на основе близости к линии-жертве на подложке, числа в квадратных скобках показывают степень влияния на линию-жертву каждого потенциального агрессора [3] (она характеризуется прежде всего емкостной связью между этими линиями, которая определяется их близостью на подложке). Чем выше степень влияния, тем большую задержку вызывает агрессор на жертве. Суммарная индуцированная задержка на линии-жертве в этом случае пропорциональна сумме задержек агрессоров, имеющих противоположное направление переключения сигналов.

© Ю. А. Скобцов, В. Ю. Скобцов, К. М. Нассер Іяд, 2012

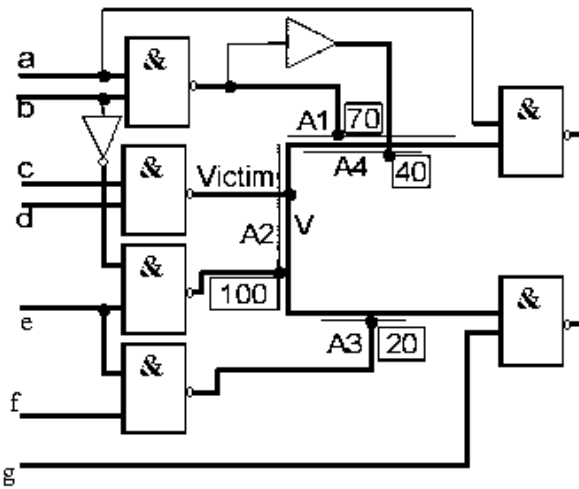


Рисунок 1 – Индуцированная задержка со многими линиями-агрессорами

Покажем, как влияет выбор тестовой пары на тестирование индуцированной задержки со многими агрессорами. Для начала возьмем тестовую пару $a=0, b=0, c=1, d=R, e=R, f=1, g=1$, где входы d, e имеют задний фронт. Эта тестовая пара обеспечивает задний фронт на линии-жертве $V=R$ и передний фронт (противоположный) на следующих линиях-агрессорах: $A2=F$ с максимальным значением $C_{A2}=100$ и $A3=F$ с $C_{A3}=20$, что дает суммарную задержку $100+20=120$. Это приводит к замедлению переднего фронта на линии-жертве V , которое при значении входа $g=1$ (что имеет место на данной тестовой паре) распространяется на внешний выход вентиля $G6$.

Постановка задачи

Выбор множества линий агрессоров для данной линии-жертвы можно формализовать следующим образом. Пусть $A=\{a_1, a_2, \dots, a_m\}$ обозначает множество возможных линий агрессоров. Для каждой линии-агрессора a_i определим c_i - степень его влияния на жертву, которая определяется, прежде всего, величиной емкостной связи между этими линиями.

Далее определим множество булевых переменных $S=\{s_1, s_2, \dots, s_m\}$, где каждая переменная s_i соответствует своему агрессору a_i . При этом

$$s_i = \begin{cases} 1, & \text{если } a_i \text{ принадлежит текущему множеству агрессоров} \\ 0, & \text{иначе} \end{cases}$$

То есть каждая переменная s_i является признаком принадлежности линии a_i к рассматриваемому в текущий момент множеству агрессоров. Тогда степень влияния текущего множества агрессоров на линию-жертву можно определить следующим образом

$$\sum_{i=1}^m s_i \cdot c_i$$

Для максимального увеличения кросс-индуцированной задержки текущее множество агрессоров должно быть возбуждено таким образом, чтобы эффект задержки на линии-жертве был максимальным. То есть необходимо найти такой булев вектор S , при котором имеет место

$$\max \sum_{i=1}^m s_i \cdot c_i$$

и выполняются необходимые логические условия для построения тестовой пары [2]. Очевидно, что данная задача носит переборный характер и является NP-трудной. В худшем случае придется перебрать 2^m всевозможных подмножеств линий-агрессоров.

Алгоритмы выбора множества линий-агрессоров

В этом случае можно использовать различные подходы к поиску множества-линий агрессоров, которые дают максимальное увеличение индуцированной задержки. Одной из самых простых является «жадная» эвристика. При этом все линии-агрессоры сортируются по убыванию значений степени влияния на линию-жертву c_i и заносятся в \max -стек T , где верхушка стека содержит агрессор с наибольшим значением c_{\max} . Далее линии-агрессоры итеративно извлекаются из стека (с невозрастающими значениями c_i) и включаются в допустимое решение, если выполняются соответствующие ограничения. Ниже представлен "жадный" алгоритм выбора линий-агрессоров в виде псевдокода.

Алгоритм A1 (жертва, множество агрессоров, значения степени влияния c_i)

```

{
  Ввод значений степени влияния агрессоров на
  линию-жертву;
  Сортировка по убыванию значений  $c_i$ ;
  Занесение значений  $c_i$  в  $\max$ -стек;
  While T!=пусто
  {
     $a_k=POP(T)$ ;
     $s_k=1$ ;
    Проверка ограничений на допустимость реше-
    ния;
    если решение  $S=\{s_1, s_2, \dots, s_m\}$  является недопусти-
    мым,
    то  $s_k=0$ ;
  }
  Вывод решения  $S=\{s_1, s_2, \dots, s_m\}$ ;
}
    
```

Этот алгоритм имеет полиномиальную сложность и прост в реализации, но не гарантирует получения лучшего решения. Здесь некоторые подмножества агрессоров могут быть пропущены, в том числе те, которые дают максимальную индуцированную задержку на линии-жертве.

Отметим, что задача выбора множества линий-агрессоров, которое дает максимальную задержку, может быть также решена с помощью генетического алгоритма (ГА), поскольку здесь потенциальное решение представляется булевым вектором $S = \{s_1, s_2, \dots, s_m\}$, где каждая переменная s_i соответствует своему агрессору a_i . При этом можно применять стандартные генетические операторы кроссинговера, мутации и репродукции [4,5]. В качестве фитнес-функции в этом случае естественно использовать степень влияния данного множества линий-агрессоров на линию-

$$\text{жертву } F = \sum_{i=1}^m s_i \cdot c_i .$$

Генетический алгоритм построения теста для индуцированной задержки

На наш взгляд применение генетического алгоритма (ГА) при построении тестов для более сложных моделей неисправностей более оправдано, чем для константных неисправностей, поскольку в этом случае трудно использовать детерминированные методы. Ясно, что проверяющий тест для перекрестных неисправностей типа индуцированные задержки должен состоять из пар входных наборов, обеспечивающих приведенные выше условия. Отметим, что для данного критического пути время распространения сигнала может быть различным для разных пар входных наборов. Очевидно, что желательно найти пары наборов с максимальной задержкой, которые позволяют проверить целевые неисправности. В качестве особи ГА здесь целесообразно использовать пару входных наборов $(x_1^t, x_2^t, \dots, x_n^t; x_1^{t+1}, x_2^{t+1}, \dots, x_n^{t+1})$ с общей длиной $2n$ битов, где каждому i -ому внешнему входу схемы соответствует два связанных двоичных бита $(x_i^t; x_i^{t+1})$. Множество таких пар входных наборов составляет популяцию, которая эволюционирует с помощью генетических операторов репродукции, кроссинговера и мутации. При отборе родительских особей предлагается использовать турнирный метод.

Далее, как обычно, к родительским особям применяется генетический оператор кроссинговера. Здесь мы используем модификацию однородного кроссинговера, которая учитывает структуру хромосомы, где каждому входу схемы в паре входных наборов соответствует 2 связанных бита $(x_i^t; x_i^{t+1})$. Поскольку в этом случае каждому входу схемы соответствует 2 бита двоичных значений, то естественно, что значения 2 битов по-

томков, соответствующие определенному входу, формируются из аналогичных битов этого же входа одного из родителей. При этом случайно генерируется двоичная маска кроссинговера той же длины (с тем числом бит), что у хромосом родителей, где значение бита маски (0 или 1) показывает родителя, из которого копируется ген потомка.

Кроме однородного кроссинговера, можно использовать также и структурный кроссинговер [4,5], где обмен между родителями производится столбцами, соответствующими одной древовидной подсхеме. При этом на этапе препроцессорной обработки схема должна быть разбита на древовидные подсхемы. Тогда внешние входы схемы, которые «питают» одну древовидную подсхему, относятся к одной и той же группе. В этом случае обмен производится группами столбцов, соответствующих одной и той же древовидной подсхеме. Отметим, что при этом в одну группу попадают входы, определяющие значения внутренних «узловых» точек схемы. Особенно, это касается линии-жертвы, значения сигналов которой определяется только значениями внешних входов соответствующей древовидной подсхемы. В этом случае обмен генетическим материалом выполняется более направленно для соответствующих внутренних линий схемы, что в некоторых случаях повышает эффективность поиска превосходящих тестовых последовательностей.

После кроссинговера к полученным потомкам применяется генетический оператор мутации, где также учитывается структура хромосомы пары входных наборов, используемая в данной задаче, которая требует модификации классического оператора мутации. В качестве фитнес-функции проще всего использовать число проверяемых неисправностей, которое можно получить с помощью программы моделирования неисправностей.

Выводы

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

Литература

1. Скобцов Ю.А., Скобцов В.Ю., Нассер Яд К.М. Генерация тестов для неисправностей типа индуцированные импульсы // Информационно-керуючі системи на залізничному транспорті. – 2010. – №4. – С. 27-29.
2. Скобцов Ю.А., Скобцов В.Ю., Нассер Яд К.М. Проверяющие тесты для перекрестных неисправностей типа задержка // Информационно-керуючі системи на залізничному транспорті. – 2011. – №4. – С. 56-59.

3. Kunal P., Ganeshpure, Sandip Kundu. Automatic test generation for maximal circuits noise in multiple aggressor crosstalk faults // 2007.-Proceedings DATE07.
4. Скобцов Ю.А., Скобцов В.Ю. Логическое моделирование и тестирование цифровых устройств. – Донецк: ИПММ НАНУ, ДонНТУ, 2005. – 436 с.
5. Скобцов Ю.А., Сперанский Д.В., Скобцов В.Ю. Моделирование, тестирование и диагностика цифровых устройств: Учебное пособие. – М.: Национальный открытый университет «ИНТУ-ИТ», 2012. – 439 с.

Резюме

Рассмотрено применение генетических алгоритмов и логического моделирования для построения тестов поиска неисправностей в виде индуцированных задержек

Розглянуто використання генетичних алгоритмів та логічного моделювання для побудови тестів пошуку відмов у вигляді індукованих затримок

The application of genetic algorithms and logic simulation to test construction troubleshooting as induced delays is described

Ключові слова: неисправностей в виде индуцированных задержек, тесты, генетические алгоритмы

Рецензент д.т.н., профессор Кривуля Г. Ф. (ХНУРЭ)

Поступила 23.06.2012 г.