

ГЕНЕРАЦИЯ ТЕСТОВ ДЛЯ ПРОВЕРКИ НЕИСПРАВНОСТЕЙ ЦИФРОВЫХ СИСТЕМ

ХАХАНОВ В.И., МАСУД М.Д. МЕХЕДИ

Предлагается метод генерации тестов для константных неисправностей, использующий кубические покрытия списков неисправностей для построения одномерных путей активизации. Проблема синтеза тестов цифровых устройств формализуется в виде линейного уравнения диагноза.

1. Введение

Достижения в микроэлектронике привели к созданию кристаллов сверхбольшой степени интеграции, на которых реализуются: микропроцессоры, сигнальные процессоры, заказные БИС, ПЛИС. В свою очередь для разработки цифровых систем (ЦС) на СБИС необходимы новые технологии автоматизированного проектирования, включающего: описание, верификацию, синтез, трассировку, размещение и имплементацию на кристалле [1-6]. Существующие САПР ведущих фирм мира: Xilinx, Aldec, Altera, Actel, Asset, Cadence, Mentor Graphics, Synopsys достаточно успешно решают все задачи, относящиеся к автоматическому проектированию [7]. При этом существует проблема верификации проектов ЦС, связанная с их размерностью, которая достигает нескольких миллионов эквивалентных вентилях на кристалле. Ее решение сводится к формальной или тестовой проверке соответствия проекта, представленного на языках высокого уровня описания аппаратуры (VHDL, Verilog, System C, Abel, EDIF) заданной спецификации. Тестовая верификация заключается в сравнительном анализе результатов моделирования спецификации и формальном описании на представительной и валидной выборке входных последовательностей, называемых тестом. Для реализации упомянутой процедуры необходимо решить задачу генерации теста проверки дефектов заданного класса [3-6], к которому предъявляются противоречивые требования: он должен быть минимальной длины и проверять все существенные неисправности [1].

Рассмотрим одиночные константные неисправности (ОКН) эквивалентных линий проектов цифровых систем. Их число определяется удвоенным количеством таких линий и достигает в реальных проектах десятков и сотен тысяч. Качество теста есть отношение проверенных неисправностей к их общему числу, выраженное в процентах. В качестве объекта выступает ЦС, реализуемая в кристаллах ПЛИС вентиляжного, функционального и алгоритмического уровней, представленная на языке описания аппаратуры VHDL. Существующие методы генерации тестов можно классифицировать по трем группам: алгоритмические, псевдослучайные и детерминированные [1]. Последние ориентированы на построение тестов для конкрет-

ных неисправностей или для проверки исправного поведения цифровых систем. Учитывая, что по вычислительной сложности данная задача относится к NP-полным [1], то многообразие существующих методов построения тестов укладывается в схему снижения размерности вычислительных затрат. В связи с этим далее предлагается метод активизации неисправностей, связанный с построением одномерных путей активизации ОКН [4] путем использования линейной зависимости между тестом, неисправностью и моделью ЦС или ее функционального элемента, заданных кубическими покрытиями (КП) [1].

2. Модель представления цифровой системы

Автоматная модель примитива представляется в виде:

$$M = \langle X, Y, Z, f, g \rangle,$$

где $X = (X_1, X_2, \dots, X_i, \dots, X_m)$, $Y = (Y_1, Y_2, \dots, Y_i, \dots, Y_h)$,

$Z = (Z_1, Z_2, \dots, Z_i, \dots, Z_k)$ – множества входных, внутренних и выходных автоматных переменных, отношения между которыми описываются уравнениями [2]:

$$Y(t) = f[X(t-1), X(t), Y(t-1), Z(t-1)];$$

$$Z(t) = g[X(t-1), X(t), Y(t-1), Y(t), Z(t-1)]. \quad (1)$$

Функциональный последовательностный примитив задается компонентами: $F^2 = \langle (t-1, t), (X, Z, Y), \{A^2\} \rangle$, где $(t-1, t)$ – два автоматных соседних такта в описании функции; (X, Z, Y) – векторы входных, внутренних и выходных переменных; $\{A^2\}$ – двухтактный алфавит описания состояний (переходов) автоматных переменных [1,2]: $A^2 = \{Q=00, E=01, H=10, J=11, O=\{Q, H\}, I=\{E, J\}, A=\{Q, E\}, B=\{H, J\}, S=\{Q, J\}, P=\{E, H\}, C=\{E, H, J\}, F=\{Q, H, J\}, L=\{Q, E, J\}, V=\{Q, E, H\}, Y=\{Q, E, H, J\}, A^1=\{0, 1, X=\{0, 1\}\}, \emptyset(U)\}$.

Формой описания примитива выступает КП: $C = (C_1, C_2, \dots, C_i, \dots, C_n)$, где $C_i = (C_{i1}, C_{i2}, \dots, C_{ij}, \dots, C_{iq})$ – куб, включающий входные, внутренние, выходные координаты $C_i = (C_i^X, C_i^Y, C_i^Z)$, $q = m + h + k$. Для комбинационного автомата формат описания КП: $F^1 = \langle (t), (X, Z), \{A^1\} \rangle$ определяется отношениями на $(q = m + k)$ -мерном векторе переменных $C_i = (C_i^X, C_i^Z)$. Формат задает многовыходовой комбинационный примитивный элемент с m входами и k выходами. Двоичная булева функция от m переменных $Z = f(X_1, X_2, \dots, X_m)$ определяется при $k=1$.

Проблемы тестирования цифрового устройства (примитива) формулируются в условиях неопределенности одного из компонентов модели

$$W = (M, L, T),$$

где M – модель, представленная кубическим покрытием C ; L – кубическое покрытие списков неисправностей (КПЧ) (Fault Lists Cubic Covering – FLCC); T – тест проверки неисправностей.

Покрытие неисправностей для примитива или ЦС задается в виде:

$$L = (L_1, L_2, \dots, L_i, \dots, L_n),$$

здесь $L_i = (L_{i1}, L_{i2}, \dots, L_{ij}, \dots, L_{iq})$ – куб, включающий входные, внутренние, выходные координаты: $L_i = (L_i^X, L_i^Y, L_i^Z)$, $q=m+h+k$; $(L_{ij}^Y, L_{ij}^Z) = \{0, 1, X\}$; 0 – определяет вычитание (дополнение) списка L_{ij} ; 1 – задает пересечение L_{ij} ; $X=\{0,1\}$ – идентифицирует несущественность списка неисправностей L_{ij} . Если L_{ir}^Z – выходная наблюдаемая переменная, то (0) 1 – есть идентификатор (не-) проверки неисправностей куба L_i на выходе r , $X=\{0,1\}$ – задает неопределенное состояние выходной координаты L_{ir}^Z , которое можно интерпретировать, как 0, и как 1.

3. Проблема моделирования неисправностей

Кубическое покрытие списков неисправностей L для вектора T и покрытия примитива C вычисляется по линейному уравнению

$$T \oplus C = L, \quad (2)$$

где \oplus – бинарная координатная операция XOR, определяющая взаимодействие компонентов $T, C (L)$ в троичном алфавите:

$$T_j \oplus C_{ij} = \begin{array}{|c|c|c|c|} \hline \oplus & 0 & 1 & X \\ \hline 0 & 0 & 1 & X \\ \hline 1 & 1 & 0 & X \\ \hline X & X & X & X \\ \hline \end{array}. \quad (3)$$

Универсальная формула анализа КПСН, полученного в результате применения (3) к тест-вектору T и покрытию многовыходового примитива C для определения по выходу r списка проверяемых неисправностей L_r , имеет следующий вид:

$$L_r = \bigcup_{\forall i(T_r \oplus C_{ir}=1)} \bigcap_{j=1}^k L_{ij}^{T_j \oplus C_{ij}}, \quad (4)$$

где $L_{ij}^{T_j \oplus C_{ij}} = \begin{cases} L_{ij} \leftarrow T_j \oplus C_{ij} = 1; \\ \bar{L}_{ij} \leftarrow T_j \oplus C_{ij} = 0, \end{cases}$

\bar{L}_{ij} – рассматривается как список дефектов, относящийся к линии j , который следует вычитать из неисправностей, проверяемых по невыходным линиям примитива; L_{ij} – неисправности, которые необходимо пересекать с невыходными списками. Если тест-вектор определен в троичном алфавите $T_j = \{0,1,X\}$, то после получения покрытия списков неисправностей каждый куб L_i следует верифицировать по правилу

$$L = L - L_i \Leftarrow [f(C_i^{(X)}) = X], i = \overline{1, n}. \quad (5)$$

Здесь знак “–” эквивалентен теоретико-множественному вычитанию; $C_{ij}^{(X)} \in C_i^{(X)}$ – входные координаты куба покрытия ($j = \overline{1, m}$), модифицированные для выполнения прямой импликации $f[C_i^{(X)}]$ в целях определения существенности координат вектора $C_{ij}^{(X)} = X$ на состояние выхода r примитива,

$$\text{по правилу } C_{ij}^{(X)} = \begin{cases} C_{ij} \Leftarrow T_j \neq X; \\ X \Leftarrow T_j = X. \end{cases}$$

Пример 1. Для мультиплексора, описанного функцией $f(a, b, c) = a\bar{c} \vee bc$, КПСН для тестового набора $T = (1010)$ определяется по формуле (2):

$$T(1010) \oplus C_{MUX} \begin{array}{|c|c|c|c|} \hline a & b & c & d \\ \hline 0 & X & 0 & 0 \\ \hline 1 & X & 0 & 1 \\ \hline X & 0 & 1 & 0 \\ \hline X & 1 & 1 & 1 \\ \hline \end{array} = L \begin{array}{|c|c|c|c|} \hline a & b & c & d \\ \hline 1 & X & 1 & 0 \\ \hline 0 & X & 1 & 1 \\ \hline X & 0 & 0 & 0 \\ \hline X & 1 & 0 & 1 \\ \hline \end{array}. \quad (6)$$

Здесь L – кубическое покрытие списков неисправностей, по которому можно записать дизъюнктивную нормальную форму или теоретико-множественное уравнение для вычисления выходного списка дефектов на векторе (1010):

$$L = \bar{a}c \vee b\bar{c} = (c - a) \cup (b - c).$$

Из данного выражения следует, что собственный список проверяемых входных ОКН мультиплексора представлен множеством $L = \{b^1, c^0\}$.

В общем случае для вычисления собственных неисправностей примитива применяется **F-процедура**:

- 1) терм дизъюнктивной нормальной формы, имеющий более одной переменной без инверсии или все переменные с инверсией, вычеркивается;
- 2) оставшиеся термы должны иметь одну переменную без инверсии, которая формирует собственную проверяемую неисправность $j^{\bar{T}_j}$, инверсную состоянию координаты тест-вектора T_j .

Пример 2. Задано КП функции

$$C(\bar{X}_2 \bar{X}_3 \vee X_1 X_2 X_3) = \begin{array}{|c|c|c|c|} \hline X_1 & X_2 & X_3 & Y \\ \hline X & 0 & 0 & 1 \\ \hline 1 & 1 & 1 & 1 \\ \hline X & 0 & 1 & 0 \\ \hline X & 1 & 0 & 0 \\ \hline 0 & 1 & X & 0 \\ \hline \end{array}. \quad (7)$$

Списки идентификаторов неисправностей для трех входов определены подмножествами:

$$L_1 = \{1,2,4,5\}; L_2 = \{1,2,3,6\}; L_3 = \{1,3,4,7\}. \quad (8)$$

Определить выходной список неисправностей для тест-вектора 1111.

Решение сводится к выполнению \oplus -операции между вектором T и покрытием C :

$$T(1111) \oplus C \begin{array}{c|ccc|c} X_1 & X_2 & X_3 & Y \\ \hline X & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ X & 0 & 1 & 0 \\ X & 1 & 0 & 0 \\ 0 & 1 & X & 0 \end{array} = L \begin{array}{c|ccc|c} X_1 & X_2 & X_3 & Y \\ \hline X & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ X & 1 & 0 & 1 \\ X & 0 & 1 & 1 \\ 1 & 0 & X & 1 \end{array}.$$

Далее по единичным значениям выхода Y в покрытии L записывается дизъюнктивная нормальная форма (теоретико-множественное уравнение) проверяемых входных списков:

$$L = X_2 \bar{X}_3 \vee \bar{X}_2 X_3 \vee X_1 \bar{X}_2 = (X_2 - X_3) \cup (X_3 - X_2) \cup (X_1 - X_2). \quad (9)$$

Подстановкой вместо переменных X_i соответствующих списков L_i получается следующий результат:

$$L = (\{1,2,3,6\} - \{1,3,4,7\}) \cup (\{1,3,4,7\} - \{1,2,3,6\}) \cup (\{1,2,4,5\} - \{1,2,3,6\}) = \{2,4,5,6,7\}. \quad (10)$$

4. Проблема генерации тестов

Тест проверки дефектов примитива, задаваемых L_i^1 -кубом покрытия списка неисправностей $L = (L_i^0, L_i^1)$, определяется по уравнению:

$$L_i^1 \oplus C = T^k, \quad (11)$$

где L_i^0, L_i^1 — кубы, имеющие нулевые или единичные значения на выходной координате g . При этом определяются векторы-кандидаты в тест $T_t^k \in T^k$. Из множества T^k формируется тест T , где в качестве элемента рассматривается набор $T_t \in T$. Каждый $T_t \in T$ должен удовлетворять условиям:

$$T_t = C_i \cap T_t^k \Leftarrow \exists i(C_i \cap T_t^k \neq \emptyset). \quad (12)$$

Пример 3. Для функции AND-NOT построить тест, проверяющий списки неисправностей, заданные кубом $L(011)$.

Приведем процедуру получения теста на основе последовательного применения формул (11) и (12):

$$L(011) \oplus_{AND} C \begin{array}{c|ccc|c} a & b & c \\ \hline 0 & X & 1 \\ X & 0 & 1 \\ 1 & 1 & 0 \end{array} = T^k \begin{array}{c|ccc|c} a & b & c \\ \hline 0 & X & 0 \\ X & 1 & 0 \\ 1 & 0 & 1 \end{array} = T \begin{array}{c|ccc|c} a & b & c \\ \hline 1 & 1 & 0 \\ 1 & 0 & 1 \end{array}. \quad (13)$$

В результате получены два набора, которые одновременно активизируют входную переменную b . Следовательно, тест является полным относительно проверки одиночных константных дефектов на линиях b , c .

Тест проверки одиночных константных неисправностей одновыходового примитива, задаваемых L^1 -покрытием списка дефектов

$$L^1 = \begin{array}{c|ccc|c|c} X_1 & X_2 & \dots X_j \dots & X_m & Z \\ \hline 1 & 0 & \dots 0 \dots & 0 & 1 \\ 0 & 1 & \dots 0 \dots & 0 & 1 \\ \dots & \dots & \dots 1 \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots 0 \dots & 1 & 1 \end{array}, \quad (14)$$

определяется **T-процедурой**:

1. Формирование кандидатов в тест $T_t^k \in T^k$ путем выполнения операций над векторами из L^1, C :

$$L^1 \oplus C = T^k. \quad (15)$$

2. Получение теста T ($T_t \in T$) из множества кандидатов T^k на основе применения выражения (12).

3. Минимизация множества тест-векторов T путем выполнения операции поглощения:

$$T = T - T_r \Leftarrow T_i \cap T_r = T_r.$$

Пример 4. Построить тест проверки всех ОКН для функции, заданной покрытием (7).

Генерируется L^1 -покрытие списка дефектов по аналогии с (14) и для строки L_i^1 выполняется \oplus -операция по (11):

$$L^1 \begin{array}{c|ccc|c} X_1 & X_2 & X_3 & Y \\ \hline 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{array} \oplus C \begin{array}{c|ccc|c} X_1 & X_2 & X_3 & Y \\ \hline X & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ X & 0 & 1 & 0 \\ X & 1 & 0 & 0 \\ 0 & 1 & X & 0 \end{array} = T^k(L_i^1), i=1,3. \quad (16)$$

К полученным подмножествам кандидатов в тест:

$$T^k(L_1^1) = \begin{array}{c|ccc|c} X_1 & X_2 & X_3 & Y \\ \hline X & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ X & 0 & 1 & 1 \\ X & 1 & 0 & 1 \\ 1 & 1 & X & 1 \end{array};$$

$$T^k(L_2^1) = \begin{array}{c|ccc|c} X_1 & X_2 & X_3 & Y \\ \hline X & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ X & 1 & 1 & 1 \\ X & 0 & 0 & 1 \\ 0 & 0 & X & 1 \end{array};$$

$$T^k(L_3^1) = \begin{array}{c|ccc|c} X_1 & X_2 & X_3 & Y \\ \hline X & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ X & 0 & 0 & 1 \\ X & 1 & 1 & 1 \\ 0 & 1 & X & 1 \end{array}.$$

применяется процедура (12), которая определяет в данном случае уже минимальный тест, гарантированно проверяющий схемную структуру, которая соответствует двухуровневой реализации булевой функции (7):

$$T = \begin{array}{c|cccc} & X_1 & X_2 & X_3 & Y \\ \hline & 0 & 1 & 1 & 0 \\ & 1 & 1 & X & 1 \\ \hline X & 1 & 0 & 0 & 0 \\ & 1 & 0 & 1 & 0 \\ X & 1 & 1 & 1 & 1 \\ X & 0 & 0 & 0 & 1 \\ & 0 & 0 & X & 1 \\ \hline X & 0 & 1 & 1 & 0 \\ & 1 & 1 & 0 & 0 \\ X & 0 & 0 & 1 & 1 \\ X & 1 & 1 & 1 & 1 \end{array} \quad (17)$$

5. Проблема верификации тестов

Она определяет возможность верификации результатов, полученных при решении первых двух проблем

$$T \oplus L = C. \quad (18)$$

Необходимость такой процедуры связана с многочисленными ошибками, возникающими в процессе ручного или автоматизированного проектирования тестов. Кроме того, решение проблемы 3 обязательно для верификации покрытия неисправностей при наличии в тесте символов X. Для полученных наборов (17) верификация путем выполнения операций $L_i^1 \oplus T$ ($L_i^1, i = \overline{1,3}$), дает положительный результат:

$$\begin{array}{c} L^1 \\ \oplus T \\ = C \end{array} \begin{array}{c|cccc} X_1 & X_2 & X_3 & Y \\ \hline 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ \hline X_1 & X_2 & X_3 & Y \\ \hline 0 & 1 & 1 & 0 \\ 1 & 1 & X & 1 \\ \hline X & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ X & 1 & 1 & 1 \\ X & 0 & 0 & 1 \\ X & 0 & 0 & 1 \\ 0 & 0 & X & 1 \\ \hline X & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ X & 0 & 0 & 1 \\ X & 1 & 1 & 1 \end{array} \begin{array}{c} \oplus \\ = \\ \cdot \end{array}$$

Это означает, что любой вектор $L_i^1 \oplus T$ ($L_i^1 \in C$) является кубом исходного покрытия функции. Следовательно, тест проверки неисправностей не имеет ошибок проектирования.

6. Заключение

Предложенный метод также эффективно обрабатывает последовательностные примитивы цифровых автоматов, описанные двухтактными КП [2]. Последние формализуют описания алгоритмов в виде примитивов, соответствующих графам переходов, граф-схемам, таблицам переходов цифровых автоматов.

Предложенная технология тестирования по уравнению $T \oplus C = L$ дает возможность: моделировать неисправности на основе анализа КП; получать дедуктивные формулы для любых типовых функциональных элементов; проектировать компилятивные симуляторы для обработки цифровых устройств произвольного уровня описания; генерировать тесты для цифровых систем на основе использования КПСН; верифицировать результаты моделирования неисправностей и генерации тестов, проектировать аппаратурные быстродействующие симуляторы.

Модели и методы реализованы в виде программных приложений. Последние используются при генерации тестов для проектов на основе FPGA и CPLD. Класс обрабатываемых структур: конечные автоматы, описанные в виде графов переходов, а также булевы уравнения с триггерными схемами. Входной язык описания цифровых систем – VHDL. Программы генерации тестов и моделирования неисправностей ориентированы на интеграцию в систему проектирования фирмы Aldec [7].

Время проектирования теста функционально зависит от квадрата суммарного объема кубических покрытий:

$$W(T) = \left[\sum_{i=1}^M (n_i \times q_i) \right]^2.$$

Литература: 1. *Abramovici M., Breuer M.A. and Friedman A.D.* Digital System Testing and Testable Design, Computer Science Press, 1998. 652 p. 2. *Хаханов В.И.* Техническая диагностика элементов и узлов персональных компьютеров. К.: ИЗМН. 1997. 308 с. 3. *Ермилов В.А.* Метод отбора существенных неисправностей для диагностики цифровых схем. Общие выражения для неисправностей, возможных при эксперименте // Автоматика и телемеханика, 1971. № 1. С. 159-167. 4. *Armstrong D.B.* A deductive method of simulating faults in logic circuits. IEEE Trans. on Computers. Vol. C-21. No. 5. 1972. P. 464-471. 5. *Бургер А.Г.* Многозначное дедуктивное моделирование цифровых устройств // Автоматика и вычислительная техника. 1982. №4. С.77-82. 6. *Levendel Y.H. and Menon P.R.* Comparison of fault simulation methods – Treatment of unknown signal values. Journal of digital system. 1980. Vol.4. P.443-459. 7. *Active-VHDL Series.* Book №1-№4. Reference Guide. ALDEC Inc. 1998. 206 p.

Поступила в редколлегию 21.11.2000

Рецензент: д-р техн. наук, проф. Кривуля Г.Ф.

Хаханов Владимир Иванович, д-р техн. наук, профессор кафедры автоматизации проектирования вычислительной техники ХТУРЭ. Научные интересы: техническая диагностика вычислительных систем. Увлечения: баскетбол, футбол, горные лыжи. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 40-93-26. E-mail: Hahanov@Kture.Kharkov.ua

Масуд М.Д. Мехеди, аспирант кафедры автоматизации проектирования вычислительной техники ХТУРЭ. Научные интересы: генерация тестов для цифровых систем.. Увлечения: иностранные языки. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 40-93-26.