

КУБИЧЕСКОЕ МОДЕЛИРОВАНИЕ НЕИСПРАВНОСТЕЙ ЦИФРОВЫХ СИСТЕМ НА ОСНОВЕ FPGA, CPLD

ХАХАНОВ В.И., КОВАЛЕВ Е.В.,
МАСУД М.Д. МЕХЕДИ, ХАК ХМ. ДЖАХИРУЛ

Предлагаются модели и методы анализа цифровых проектов для генерации тестов и моделирования неисправностей. Используется двухтактная кубическая алгебра для проектирования компактных кубических покрытий последовательностных примитивов, генерации тестов, моделирования неисправностей и исправного поведения цифровых схем. Проблемы тестирования цифровых устройств формализуются в виде линейных уравнений. Описывается кубический метод моделирования неисправностей примитива, который позволяет: транспортировать входные списки неисправностей на выходы примитива; генерировать аналитические уравнения для дедуктивного моделирования цифровых схем вентильного, функционального, алгоритмического уровней описания; строить компилятивные и интерпретативные симуляторы неисправностей цифровых схем. Предлагается метод генерации тестов для константных неисправностей, использующий кубические покрытия списков неисправностей для построения одномерного пути активизации.

1. Введение

Программируемые логические интегральные схемы (ПЛИС) – Field Programable Gate Array (FPGA), Complex Programable Logic Device (CPLD) – достойно конкурируют с базовыми матричными кристаллами, сигнальными процессорами. Такой успех определяется:

- использованием субмикронных технологий изготовления кристаллов, применением Hardware-Software Cooperation-Design;
- минимальным временем проектирования цифровой системы (4-5 месяцев);
- высоким быстродействием выполнения операций (до 500 МГц);
- большой степенью интеграции элементов на кристалле (до 3,5 млн.).

Вместе с преимуществами ПЛИС существуют и проблемы их тестирования, стимулирующие развитие таких методов, алгоритмов и программ, которые должны обеспечивать:

- 1) тестирование цифровых проектов большой размерности, вентильного, функционального алгоритмического уровней описания, заданных в форме графов переходов конечных автоматов, булевых уравнений, многоуровневых иерархических структур;
- 2) проектирование тестов в виде покрытия одномерных путей активизации для проверки одиночных константных неисправностей, с полнотой, близкой к 100%;

3) приемлемое быстродействие алгоритмов моделирования неисправностей для оценки качества тестов; построения алгоритмов поиска дефектов.

4) верификацию и диагностирование синтезированных цифровых устройств на основе FPGA, CPLD;

5) возможность параллельного выполнения векторных операций логического анализа для генерации тестов и оценки их качества;

6) поддержку стандарта VHDL для описания цифрового устройства и полученного теста;

7) возможность интегрирования в существующие системы автоматизированного проектирования ведущих фирм мира.

Теоретические источники работы: многозначная алгебра [1], двухтактная кубическая алгебра [2], дедуктивный метод моделирования неисправностей [3-6], методы генерации тестов [7].

Предпочтительным по быстродействию является дедуктивный метод моделирования неисправностей. Он позволяет за одну итерацию обработки схемы определить все константные дефекты, проверяемые на входном тестовом векторе. Но данный метод ориентирован на вентильный уровень описания цифровых схем. Это связано со сложностью решения проблемы генерирования выходных списков неисправностей (output fault list generation) для невентильных примитивов.

Предлагаемый метод кубического моделирования неисправностей позволяет обрабатывать цифровые схемы, описанные на вентильном, функциональном и алгоритмическом уровнях. Обратной стороной решения упомянутой проблемы является метод генерации тестов для константных неисправностей, использующий кубические покрытия примитивов для построения одномерного пути активизации.

2. Математический аппарат анализа примитива

Автоматная модель последовательностного 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:

$$\begin{aligned} Y(t) &= f[X(t-1), X(t), Y(t-1), Z(t-1)]; \\ Z(t) &= g[X(t-1), X(t), Y(t-1), Y(t), Z(t-1)]. \end{aligned} \quad (1)$$

Переменные $Z(t)$ отличаются от $Y(t)$ тем, что первые наблюдаемы по выходным линиям, а $Y(t)$ в этом смысле есть внутренние. Формат автоматных переменных для записи кубического покрытия, соответствующий (1), имеет следующий вид:

$$\begin{array}{|c|c|c|} \hline X(t-1) & Y(t-1) & Z(t-1) \\ \hline X(t) & Y(t) & Z(t) \\ \hline \end{array},$$

которому отвечает автоматная модель, изображенная на рис. 1.

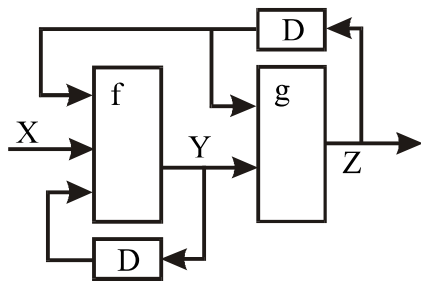


Рис.1. Автоматная модель примитива

Функциональный последовательный 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]:

$$A^2 = \{Q=00, E=01, H=10, J=11, O=\{Q, H\}, I=\{E, J\}, A=\{Q, E\}, B=\{H, J\}, S=\{Q, J\}, P=\{E, H\}, C=\{E, H, J\}, F=\{Q, H, J\}, L=\{Q, E, J\}, V=\{Q, E, H\}, Y=\{Q, E, H, J\}, A^1=\{0, 1, X=\{0, 1\}\}, \emptyset (U)\}.$$

Примитив описывается кубическим покрытием

$$C = (C_1, C_2, \dots, C_i, \dots, C_n),$$

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 кубического покрытия

$$F^1 = \langle (t), (X, Z), \{A^1\} \rangle$$

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 $k=1$.

Пример 1. Преимущества двухтактного алфавита A^2 демонстрируются на примере проектирования кубического покрытия CD-триггера. Исходное описание определено в виде таблицы переходов:

$$C_{CD} = \begin{array}{c|cccc|c} C_{t-1} & C_t & D_{t-1} & D_t & Q_{t-1} & Q_t \\ \hline 1 & 0 & 1 & 1 & X & 1 \\ 0 & 0 & X & X & 1 & 1 \\ 0 & 1 & X & X & 1 & 1 \\ 1 & 1 & X & X & 1 & 1 \\ 1 & 0 & 0 & 0 & X & 0 \\ 0 & 0 & X & X & 0 & 0 \\ 0 & 1 & X & X & 0 & 0 \\ 1 & 1 & X & X & 0 & 0 \end{array}$$

Далее применяется **C-процедура** получения двухтактного кубического покрытия:

1. Выполнение координатной операции конкатенации (#):

$$C_{t-1} \# C_t = \begin{array}{c|cccc} \# & 0 & 1 & X \\ \hline 0 & Q & E & A \\ 1 & H & J & B \\ X & O & I & Y \end{array}$$

для преобразования пары соседних во времени однотактных символов в один двухтактный.

2. Итеративная минимизация полученного покрытия по правилу – два куба объединяются в один, если они отличаются по одной переменной.

3. Дополнение полученной модели примитива кубами описания возможных состязаний.

Применение данной процедуры к покрытию CD-триггера дает следующий результат:

$$C_{CD} = \begin{array}{c|ccc} C & D & Q \\ \hline H & J & I \\ Q & Y & J \\ E & Y & J \\ J & Y & J \\ H & Q & O \\ Q & Y & Q \\ E & Y & Q \\ J & Y & Q \end{array} = \begin{array}{c|ccc} C & D & Q \\ \hline H & J & I \\ L & Y & J \\ H & Q & O \\ L & Y & Q \end{array} = \begin{array}{c|ccc} C & D & Q \\ \hline H & J & I \\ L & Y & S \\ H & Q & O \\ H & P & X \end{array}$$

В правое покрытие добавлен куб (HPX), который идентифицирует состязания при одновременном изменении сигналов на входах C и D. Полученное покрытие имеет только 12 символов, в то время как исходное содержит 48 координат.

Применение C-процедуры к таблице переходов двухразрядного счетчика еще более эффективно, чем для триггера. Процесс получения двухтактного покрытия счетчика представлен следующим выражением:

$$\begin{array}{c|cccc|cc} V & C & A_{t-1} & B_{t-1} & A_t & B_t \\ \hline 0 & X & X & X & 0 & 0 \\ 1 & E & 0 & 0 & 0 & 1 \\ 1 & E & 0 & 1 & 0 & 0 \\ 1 & E & 1 & 0 & 1 & 1 \\ 1 & E & 1 & 1 & 1 & 0 \\ 1 & F & 0 & 0 & 0 & 0 \\ 1 & F & 0 & 1 & 0 & 1 \\ 1 & F & 1 & 0 & 1 & 0 \\ 1 & F & 1 & 1 & 1 & 1 \end{array} = \begin{array}{c|ccc} V & C & A & B \\ \hline 0 & X & 0 & 0 \\ 1 & E & Q & E \\ 1 & E & E & H \\ 1 & E & J & E \\ 1 & E & H & H \\ 1 & F & Q & Q \\ 1 & F & Q & J \\ 1 & F & J & Q \\ 1 & F & J & J \end{array} = \begin{array}{c|ccc} V & C & A & B \\ \hline 0 & X & 0 & 0 \\ 1 & E & S & E \\ 1 & E & P & H \\ 1 & E & S & S \end{array}$$

Минимальное покрытие счетчика, записанное в двухтактном алфавите, имеет всего 16 координат.

В общем случае размерность таблиц описания функции счета для n разрядов в одно- и двухтактном

покрытии определяется отношением $h = \frac{2^n}{n}$.

3. Формулировка проблем тестирования

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

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

где M – модель, представленная кубическим покрытием $C=(C_1, C_2, \dots, C_i, \dots, C_n)$; L – кубическое покрытие списков неисправностей (КПШ) (Fault Lists Cubic Covering – FLCC); T – тест проверки неисправностей.

Покрытие неисправностей для примитива или цифровой схемы задается в виде:

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

где $L_i = (L_{i1}, L_{i2}, \dots, L_{ij}, \dots, L_{iq})$ – 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_j ; 1 – задает пересечение L_j ; $X=\{0,1\}$ – идентифицирует несущественность списка неисправностей L_j . 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, так и 1.

Проблема 1. 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:

$$L = \bigcup_{\forall i(T_i \oplus C_{ir} = 1)} \bigcap_{j=1}^k L_j^{T_i \oplus C_{ij}}, \quad (4)$$

где $L_j^{T_i \oplus C_{ij}} = \begin{cases} L_j \leftarrow T_i \oplus C_{ij} = 1; \\ \overline{L_j} \leftarrow T_i \oplus C_{ij} = 0, \end{cases}$

$\overline{L_j}$ – рассматривается как список дефектов, относящийся к линии j , который следует вычитать из неисправностей, проверяемых по невыходным линиям примитива; L_j – неисправности, которые необходимо пересекать с невыходными списками.

Замечание. 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

$$L = L - L_i \leftarrow (L_i \oplus T \neq C_i), i = \overline{1, n}. \quad (5)$$

Здесь и далее знак “–” эквивалентен теоретико-множественному вычитанию.

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

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

Здесь L – кубическое покрытие списков неисправностей, по которому можно записать дизъюнктивную нормальную форму (ДНФ) 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 (6) следует, что собственный список проверяемых входных одиночных константных неисправностей мультиплексора представлен множеством $L = \{b^1, c^0\}$.

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

1) терм ДНФ, 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 3. Задано кубическое покрытие 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 :

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

Далее по единичным значениям выхода Y в покрытии L записывается ДНФ (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:

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

Проблема 2. 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 4. Для функции AND-NOT построить тест, проверяющий списки неисправностей, заданные кубом $L(011)$.

Ниже приведена процедура получения теста на основе последовательного применения формул (11) и (12):

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

В результате получены 2 набора, которые одномерно активизируют входную переменную b . Следовательно, тест является полным относительно проверки одиночных константных дефектов на линиях b, c .

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

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

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 5. Построить тест проверки всех одиночных константных неисправностей 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):

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

К полученным подмножествам кандидатов в тест:

$$\begin{array}{c} T^k(L_1^1) \\ \begin{array}{c|ccc|c} X_1 & X_2 & X_3 & Y \\ \hline X & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ X & 0 & 1 & 1 \\ X & 1 & 0 & 1 \\ 1 & 1 & X & 1 \end{array} \end{array}, \begin{array}{c} T^k(L_2^1) \\ \begin{array}{c|ccc|c} X_1 & X_2 & X_3 & Y \\ \hline X & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ X & 1 & 1 & 1 \\ X & 0 & 0 & 1 \\ 0 & 0 & X & 1 \end{array} \end{array}, \begin{array}{c} T^k(L_3^1) \\ \begin{array}{c|ccc|c} X_1 & X_2 & X_3 & Y \\ \hline X & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ X & 0 & 0 & 1 \\ X & 1 & 1 & 1 \\ 0 & 1 & X & 1 \end{array} \end{array}$$

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

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

Проблема 3. Определяет возможность верификации результатов, полученных при решении первых двух проблем

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

Необходимость данной процедуры связана с многочисленными ошибками, возникающими в процессе ручного или автоматизированного проектирования

тестов. Кроме того, согласно замечанию, 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:

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

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.

4. Алгоритм кубического моделирования неисправностей комбинационных схем

На основе применения формулы (4) определяется алгоритм кубического дедуктивного моделирования неисправностей цифрового устройства:

1. Моделирование исправного поведения очередного примитива $P_i (i = \overline{1, M})$ на тест-векторе $T_t (t = \overline{1, N})$. Если $t = N$ — формируется список $L(T)$ проверенных неисправностей на тесте T. Конец моделирования. Иначе, $t < N$ — переход к п.2.

2. Если все элементы схемы обработаны ($i = M$), выполняется сравнение векторов исправных состояний линий в двух соседних итерациях. Если векторы идентичны $T_t^r = T_t^{r-1}$, конец моделирования T_t и переход к п. 3. Иначе — переход к п. 1.

3. Определение списков неисправностей внешних входов в виде дополнения к их исправному состоянию $L_j = \{\bar{j}^T\}$.

4. Моделирование неисправностей примитива $P_i (i = \overline{1, M})$ по процедуре (4). Дополнение полученного списка неисправностью выходной линии примитива, идентифицируемой в виде $\{\bar{j}^T\}$.

5. Если ($i = M$), выполняется формирование списка проверяемых дефектов $L(T_i)$, и переход к п.1. Иначе, $i < M$ — переход к п. 4.

Быстродействие описанного выше алгоритма сквозного интерпретативного синхронного моделирования неисправностей для одной итерации имеет оценку

$$W = \sum_{i=1}^M \{q_i \times n_i \times [(0,01 \times Q)^2 + 3] + 2\},$$

где n_i, q_i — число кубов и переменных в покрытии C_i ; $(0,01 \times Q)$ — средняя длина списка неисправностей для каждой линии схемы; Q — общее число линий в цифровом устройстве.

Пример 6. Выполнить моделирование одиночных константных неисправностей схемы, представленной на рис.2.

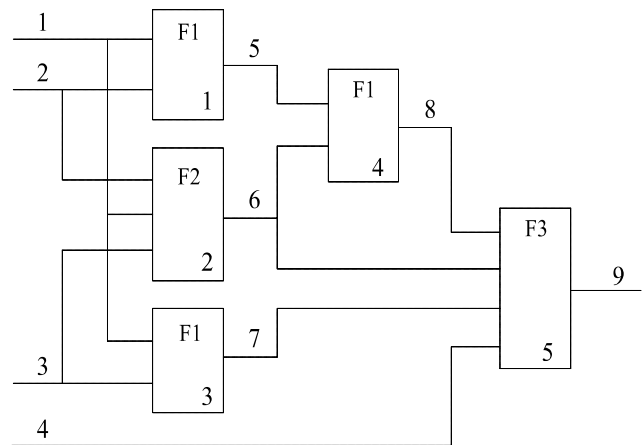


Рис.2. Цифровой функциональный модуль

Поведение примитивов задается кубическими покрытиями:

$$\begin{array}{c|ccc} C(F1) & & & \\ \hline 1 & 2 & 5 \\ 1 & 3 & 7 \\ 5 & 6 & 8 \\ \hline \overline{1} & X & \overline{0} \\ \overline{0} & 1 & 1 \\ \overline{0} & 0 & \overline{0} \end{array}; \begin{array}{c|ccc} C(F2) & & & \\ \hline 2 & 1 & 3 & 6 \\ X & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ X & 0 & 1 & 0 \\ X & 1 & 0 & 0 \\ \overline{0} & 1 & X & \overline{0} \end{array}; \begin{array}{c|ccccc} C(F3) & & & & & \\ \hline 8 & 6 & 7 & 4 & 9 \\ X & 0 & 0 & X & 1 \\ X & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & X & 1 \\ X & 0 & 1 & X & \overline{0} \\ 1 & X & 1 & X & \overline{0} \\ X & 1 & 0 & 0 & \overline{0} \end{array}$$

Рассматриваются одиночные константные неисправности линий, идентифицируемые как j^{σ} , где $\sigma = \{0,1\}$ — знак константной неисправности, j — номер линии схемы.

Для входного набора $T = 1111$ моделируется исправное поведение объекта, которое дает вектор (111101011). Затем генерируется список входных дефектов $L_j = \{\bar{j}^T\}$ цифрового устройства:

$$L_1 = \{1^0\}; L_2 = \{2^0\}; L_3 = \{3^0\}; L_4 = \{4^0\}.$$

Выполняется последовательная обработка всех элементов схемы для вычисления выходных списков неисправностей с учетом (4). Для первого примитива результат моделирования имеет вид

$$L_5 = T(110) \oplus C_{F1} \begin{array}{c|ccc} 1 & 2 & 5 \\ \hline 1 & X & \overline{0} \\ 0 & 1 & 1 \\ 0 & 0 & \overline{0} \end{array} = L \begin{array}{c|ccc} 1 & 2 & 5 \\ \hline 0 & X & \overline{0} \\ 1 & 0 & 1 \\ 1 & 1 & \overline{0} \end{array} = L_1 - L_2 = \{1^0\}.$$

Таблица 1

⊕	0	1	X	Z	G	T	K	N	Q	E	H	J	O	I	A	B	S	P	C	F	L	V	Y
0	0	1	X	Z	G	T	K	N	Q	E	H	J	O	I	A	B	S	P	C	F	L	V	Y
0	1	0	X	Z	G	T	K	N	E	Q	J	H	I	O	A	B	P	S	F	C	V	L	Y
1	0	1	X	Z	G	T	K	N	H	J	Q	E	O	I	B	A	P	S	L	V	C	F	Y
1	1	0	X	Z	G	T	K	N	J	H	E	Q	I	O	B	A	S	P	V	L	F	C	Y
0	X	X	X	X	G	T	K	N	A	A	B	B	Y	Y	A	B	Y	Y	B	B	A	A	Y
1	X	X	X	X	T	G	K	N	B	B	A	A	Y	Y	B	A	Y	Y	A	A	B	B	Y
X	0	1	X	Z	X	X	X	O	I	O	I	O	I	Y	Y	Y	Y	I	O	I	O	Y	
X	1	0	X	Z	X	X	X	I	O	I	O	I	O	Y	Y	Y	O	I	O	I	Y		
X	X	X	X	X	X	X	X	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Затем к полученному списку следует прибавить неисправность выходной линии, которая инверсна по отношению к значению координаты тест-вектора.

В данном случае получается список $L_5 = \{1^0, 5^0\}$.

Последовательная обработка оставшихся примитивов в схеме дает следующие результаты:

$$L_6 = \{1^0, 2^0, 3^0, 6^0\}; \quad L_7 = \{1^0, 7^1\};$$

$$L_8 = \{1^0, 2^0, 3^0, 5^1, 6^0, 8^0\}; \quad L_9 = \{1^0, 4^0, 7^1, 9^0\}.$$

Количество проверяемых дефектов F по q линиям формирует оценку качества теста (вектора):

$$Q = [F/(2 \times q)] \times 100\%.$$

Для тест-вектора 1111 упомянутая оценка равна:

$$Q = [4/(2 \times 9)] \times 100\% = 22 \%.$$

5. Моделирование неисправностей в последовательностных примитивных элементах

Функциональная зависимость списка дефектов по наблюдаемому выходу Z_t согласно (1) определяется уравнением

$$L_t^t = f[(T), (C), (L_X^{t-1}, L_X^t, L_Y^{t-1}, L_Z^{t-1})], \quad (19)$$

здесь $T_t \in T = (T_1, \dots, T_t, \dots, T_p)$ – пара соседних входных воздействий, где каждая координата определена в следующих сочетаниях:

$$T_t = \left[\frac{0}{0}, \frac{0}{1}, \frac{1}{0}, \frac{1}{1}, \frac{0}{X}, \frac{1}{X}, \frac{X}{0}, \frac{X}{1}, \frac{X}{X} \right]. \quad (20)$$

Двухфреймовый формат входного набора ориентирован на анализ последовательностного примитива, поскольку его покрытие в общем случае задано в двухтактном алфавите A^2 . Для технологичного выполнения операции XOR между координатами тест-вектора и кубического покрытия $T_{ij} \oplus C_{ij}$ используется табл. 1.

Каждая координата табл. 1 есть сокращенная форма записи списков неисправностей

$$L_j = T_{ij} \oplus C_{ij}.$$

Например, если $T_{ij} = (01), C_{ij} = P$, то по входной координате j в соответствии с табл. 1 получается:

$$T_{ij}(01) \oplus P\{(01), (10)\} = (\bar{L}_j^{t-1} \bar{L}_j^t) \vee (L_j^{t-1} L_j^t).$$

Табл. 2. предназначена для доопределения выходных координат куба покрытия списков неисправностей в моменты (t-1, t). Например, для координаты $L_{ij} = V$ ее значение в такте t-1 равно 0, а в момент t равно 1, что в табл.2 определяется буквой E. Исключение составляют преобразования символов S, P. Здесь

Таблица 2

0	1	X	Z	G	T	K	N	Q	E	H	J	O	I	A	B	S	P	C	F	L	V	Y	
Z	1	1	Z	G	T	Z	Z	G	E	T	J	Z	I	E	J	J/X	E/X	1	J	1	E	1	

существует различие интерпретации в зависимости от того, является ли выходная переменная функцией или аргументом к выходу, для которого строится список проверяемых неисправностей. В первом случае доопределение упомянутых символов дает (J, E), во втором – (X, X), что свидетельствует об отсутствии списков неисправностей для данного выхода в момент t-1.

Пример 7. Выполнить моделирование неисправностей на наборах 01, 11 $\bar{S}\bar{R}$ -триггера (Q, N – прямой и инверсный выходы), представленного покрытием в двухфреймовом формате

$$\begin{bmatrix} S & R & Q_{t-1} & N_{t-1} \\ S & R & Q_t & N_t \end{bmatrix}.$$

Процедура получения КПСН представлена следующим выражением:

$$T_1 \begin{bmatrix} X & X \\ 0 & 1 & 1 & 0 \end{bmatrix} \oplus C_{SR} \begin{bmatrix} S & R & Q & N \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & Q & J \\ 1 & 1 & J & Q \\ 1 & 1 & B & B \end{bmatrix} = L \begin{bmatrix} S & R & Q & N \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & Y & Y \end{bmatrix}.$$

Последующее применение правила верификации (5) к покрытию L, вследствие наличия во входном наборе символов X, определяет результат в виде КПСН:

$$L \begin{bmatrix} S & R & Q & N \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & Y & Y \end{bmatrix} \oplus T_1 \begin{bmatrix} X & X \\ 0 & 1 & 1 & 0 \end{bmatrix} = C_{SR}^k \begin{bmatrix} S & R & Q & N \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & Y & Y \end{bmatrix}.$$

Три последних куба в полученном C_{SR}^k не принадлежат к исходному покрытию C_{SR} , поэтому КПСН для триггера согласно (5) преобразуется к виду:

$$L \left[\begin{array}{cc|cc} S & R & Q & N \\ \hline 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{array} \right] = \begin{cases} L_Q = SR \vee Q_t = Q_t; \\ L_N = \overline{SR} \vee SR \vee N_t = R \vee N_t, \end{cases}$$

где инверсия переменной означает дополнение (вычитание) списка неисправностей.

С учетом выполнения F-процедуры получается, что в триггере на тест-векторе $T=(0110)$ проверяются собственные неисправности: $L = \{R^0, Q_t^0, N_t^1\}$.

Для входного набора $T_2 \left[\begin{array}{cc} & 1 & 0 \\ 1 & 1 & 1 & 0 \end{array} \right]$

моделирование неисправностей дает результат

$$T_2 \left[\begin{array}{cc} & 1 & 0 \\ 1 & 1 & 1 & 0 \end{array} \right] \oplus C_{\overline{SR}} \left[\begin{array}{cc|cc} S & R & Q & N \\ \hline 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & Q & J \\ 1 & 1 & J & Q \\ 1 & 1 & B & B \end{array} \right] = L \left[\begin{array}{cc|cc} S & R & Q & N \\ \hline 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & J & J \\ 0 & 0 & Q & Q \\ 0 & 0 & A & B \end{array} \right].$$

Поскольку входной набор – двоичный, верификация по правилу (5) отсутствует. ДНФ списков неисправностей имеет следующий вид:

$$\begin{aligned} L_Q &= \overline{SR} \vee \overline{SR}Q_{t-1}N_{t-1} \vee Q_t; \\ L_N &= SR \vee \overline{SR} \vee \overline{SR}Q_{t-1}N_{t-1} \vee N_t, \end{aligned}$$

что означает проверку собственных неисправностей триггера: $L = \{R^0, Q_t^0, N_t^1\}$.

Пример 8. Определить списки проверяемых неисправностей CD-триггера на двух тест-векторах. Для первого набора получается результат:

$$T_1 \left[\begin{array}{ccc} 1 & 0 & X \\ 0 & 0 & 0 \end{array} \right] \oplus C_{CD} \left[\begin{array}{cc|c} C & D & Q \\ \hline H & J & I \\ L & Y & S \\ H & Q & O \\ H & P & X \end{array} \right] = L \left[\begin{array}{cc|c} C & D & Q \\ \hline Q & J & I \\ C & Y & Y \\ 0 & Q & O \\ Q & P & X \end{array} \right].;$$

$$L(T_1) = \overline{C}_{t-1}\overline{C}_t D_{t-1} D_t \vee Q_t = D_t \vee Q_t.$$

Для второго вектора процедура моделирования формирует следующие кубы и списки:

$$T_2 \left[\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 1 & 0 \end{array} \right] \oplus C_{CD} \left[\begin{array}{cc|c} C & D & Q \\ \hline H & J & I \\ L & Y & S \\ H & Q & O \\ H & P & X \end{array} \right] = L \left[\begin{array}{cc|c} C & D & Q \\ \hline H & H & I \\ L & Y & S \\ H & E & O \\ H & S & X \end{array} \right].;$$

$$L(T_2) = C_{t-1}\overline{C}_t D_{t-1}\overline{D}_t \vee (\overline{C}_{t-1} \vee C_t)Q_{t-1} \vee Q_t = Q_{t-1} \vee Q_t.$$

Пример 9. Определить списки проверяемых неисправностей счетчика на двух тест-векторах. Моделирование первого набора дает результат:

$$T_1 \left[\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \end{array} \right] \oplus C_{\text{count}} \left[\begin{array}{cc|cc} V & C & A & B \\ \hline 0 & X & 0 & 0 \\ 1 & E & S & E \\ 1 & E & P & H \\ 1 & F & S & S \end{array} \right] = L \left[\begin{array}{cc|cc} V & C & A & B \\ \hline 1 & X & 0 & 1 \\ 0 & Q & S & Q \\ 0 & Q & P & J \\ 0 & C & S & P \end{array} \right].;$$

$$L_A(T_1) = \overline{V}_t \overline{C}_{t-1} \overline{C}_t A_{t-1} \overline{B}_{t-1} \vee \overline{V}_t \overline{C}_{t-1} \overline{C}_t A_{t-1} B_{t-1} \vee \overline{V}_t (C_{t-1} \vee C_t) A_{t-1} \vee A_t = A_{t-1} \vee B_{t-1} \vee A_t;$$

$$L_B(T_1) = V_t \vee \overline{V}_t \overline{C}_{t-1} \overline{C}_t B_{t-1} \vee \overline{V}_t (C_{t-1} \vee C_t) \overline{B}_{t-1} \vee B_t = V_t \vee B_{t-1} \vee C_{t-1} \vee C_t \vee B_t;$$

$$L(T_1) = V_t^0 \vee C_{t-1}^1 \vee C_t^0 \vee A_{t-1}^1 \vee B_{t-1}^1 \vee A_t^1 \vee B_t^0.$$

Результат моделирования второго набора имеет вид:

$$T_2 \left[\begin{array}{ccc} 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{array} \right] \oplus C_{\text{count}} \left[\begin{array}{cc|cc} V & C & A & B \\ \hline 0 & X & 0 & 0 \\ 1 & E & S & E \\ 1 & E & P & H \\ 1 & F & S & S \end{array} \right] = L \left[\begin{array}{cc|cc} V & C & A & B \\ \hline 1 & X & 0 & 1 \\ 0 & J & S & H \\ 0 & J & P & E \\ 0 & V & S & S \end{array} \right].;$$

$$L_A(T_2) = \overline{V}_t C_{t-1} C_t A_{t-1} B_{t-1} \vee \overline{V}_t C_{t-1} C_t \overline{A}_{t-1} \overline{B}_{t-1} \vee \overline{V}_t (\overline{C}_{t-1} \vee \overline{C}_t) A_{t-1} \vee A_t = A_{t-1} \vee A_t;$$

$$L_B(T_2) = V_t \vee \overline{V}_t C_{t-1} C_t \overline{B}_{t-1} \vee \overline{V}_t (\overline{C}_{t-1} \vee \overline{C}_t) B_{t-1} \vee B_t = V_t \vee B_t;$$

$$L(T_2) = V_t^0 \vee A_{t-1}^1 \vee A_t^1 \vee B_t^0.$$

Таким образом, проверяемые на выходах счетчика неисправности составляют множества $L(T_1)$ и $L(T_2)$, где знак константной неисправности отмечен верхним индексом.

6. Заключение

Метод кубического моделирования неисправностей является новой технологией обработки цифровых схем табличного (вентильного, функционального, алгоритмического) уровня представления. Он позволяет за одну итерацию моделировать все одиночные константные неисправности цифрового устройства, проверяемые тест-вектором. Условие применения заключается в наличии табличной формы описания примитивов цифрового устройства. Предложенный метод также эффективно обрабатывает и последовательные примитивы цифровых автоматов, описанные двухтактными кубическими покрытиями [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 кубического покрытия; 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 системы проектирования фирм: Aldec, Xilinx.

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. *Hayes J.P.* A systematic approach to multivalued digital simulation// ICCD-84: Proc. IEEE Int. Conf. Comput. 1984. No. 4. P. 177-182. 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 неис-

правностей, 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. *Burges A.G.* 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. *Abramovici M., Breuer M.A. and Friedman A.D.*, Digital System Testing and Testable Design, Computer Science Press, 1998. 652 p.

Поступила в редколлегию 25.12.99

Рецензент: д-р техн. наук 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

Ковалев Евгений Викторович, аспирант кафедры автоматизации проектирования вычислительной техники ХТУРЭ. Научные интересы: техническая диагностика цифровых автоматов. Увлечения: иностранные языки. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 40-93-26. E-mail: e_k@mail.ru

Масуд Мд. Мехеди, аспирант кафедры автоматизации проектирования вычислительной техники ХТУРЭ. Научные интересы: техническая диагностика цифровых устройств. Увлечения: поэзия. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 40-93-26.

Хак Х.М. Джахирул, аспирант кафедры автоматизации проектирования вычислительной техники ХТУРЭ. Научные интересы: техническая диагностика цифровых устройств. Увлечения: художественное фото. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 40-93-26.