

Для разрешения последнего конфликта применяется также операция сдвига точек излома трасс. В этом случае точки линии излома для первой трассы сдвигаются вверх, а для второй трассы — вниз. В результате таких действий устанавливаем истинность соотношения: $L_{\alpha 5}(\tilde{N}_1 > \tilde{N}_2)$.

Следует заметить, что предложенная топологическая модель печатной платы в виде совокупности четырех подмножеств, отображающих взаимное расположение трасс на границах дискрета КДРП, и двух массивов для хранения информации об изломах трасс в вертикальном и горизонтальном направлениях в совокупности с условиями возникновения конфликтных ситуаций позволяет организовать параллельный процесс макро- и микротрассировки.

Литература: 1. Алипов Н.В. Трассировка соединений в многослойных печатных платах (МПП) // АСУ и приборы автоматики. 1982. Вып.63. С. 24-31. 2. Алипов Н.В., Шумейко Н.А. Об одной модели печатной платы

УДК 519.713:681.326

АНАЛИЗ ГРАФОВЫХ СТРУКТУР ДЛЯ МОДЕЛИРОВАНИЯ ЦИФРОВЫХ СИСТЕМ

*ХАХАНОВ В.И., ЧУМАЧЕНКО С.В.,
КОЛЕСНИКОВ К.В., ХАХАНОВА А.В.*

Предлагаются быстродействующие алгоритмы структурного анализа сложных цифровых проектов, насчитывающих миллионы эквивалентных вентилях, которые используются на стадии предварительной обработки в целях существенного повышения быстродействия моделирования неисправностей и синтеза проверяющих тестов. Описываются структуры данных и программно-ориентированные процедуры для реализации алгоритмов в составе автоматической системы верификации тестов.

1. Введение

Актуальность данной темы определяется необходимостью значительного повышения быстродействия средств моделирования неисправностей и генерации тестов для структурно — и функционально — сложных цифровых систем, имплементированных в кристаллы программируемой логики. Автоматические системы тестирования известных фирм: Cadence, Mentor Graphics, Synopsys, Logic Vision [www.cadence.com, www.logicvision.com, www.simucad.com, www.syntest.com, www.synopsys.com, www.mentorgraphics.com], ориентированные на обработку кристаллов размерностью до 100 тыс. вентилях, затрачивают несколько часов только на моделирование неисправностей. Время анализа становится неприемлемым, если в качестве объекта выступает устройство, имеющее миллионы вентилях. Актуальным представляется решение проблемы повышения на порядок быстродействия анализа сложной цифровой системы на стадии ее проектирования в целях построения тестов верификации и анализа их качества. В рамках решения упомянутой проблемы предлагается структурный анализ цифровых проектов, позволяющий повысить быстродействие средств автоматической гене-

на этапе трассировки соединений // Электронное моделирование. 1984. № 1. С. 81-86. 3. Алипов Н.В., Литвинова Е.И. Трассировка многослойных печатных плат на основе крупнодисcretной модели // Труды УНИИРТ, Одесса. 1995. № 3. С. 72-76.

Поступила в редколлегию 12.03.2002

Рецензент: д-р техн. наук, проф. Петров Э.Г.

Алипов Николай Васильевич, д-р техн. наук, профессор кафедры проектирования и эксплуатации электронных аппаратов ХНУРЭ. Научные интересы: алгоритмизация задач автоматизированного проектирования электронно-вычислительных средств, защита информации. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 40-94-94.

Литвинова Евгения Ивановна, канд. техн. наук, доцент кафедры технологии и автоматизации производства РЭС и ЭВС ХНУРЭ. Научные интересы: алгоритмизация задач автоматизированного проектирования электронных вычислительных средств. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 40-94-94, 40-94-85.

рации тестов и моделирования неисправностей путем предварительного определения сходящихся разветвлений (СР) и использования данной информации при решении задач тестирования.

Объект тестирования — цифровой проект, представленный в форме структуры булевых уравнений, реализуемых в кристаллах программируемой логики.

Цель исследования — разработка алгоритмов структурного анализа цифровых проектов для идентификации СР в комбинационных и последовательностных схемах.

Задачи исследования:

1. Создание структурной модели цифровой схемы для синтеза тестов и моделирования путем использования процедуры обратной суперпозиции.
2. Разработка алгоритмов структурно-функционального анализа для комбинационных и последовательностных цифровых систем в целях определения множества сходящихся разветвлений и реконфигурации структуры схемы для реализации процедуры суперпозиции.
3. Программная реализация алгоритмов структурного анализа цифровых систем и сравнение их эффективности на представительной выборке комбинационных и последовательностных схем.

Исходная информация для разработки алгоритмов структурного анализа представлена публикациями: ВДР-метод (Backtraced Deductive-Parallel) моделирования неисправностей [1,2], дедуктивные модели транспортирования дефектов [4-7], параллельный метод обработки списков неисправностей функционального элемента [6] и алгоритм обратного прослеживания примитивов [8] при обработке цифрового устройства.

2. Формулировка проблемы структурного анализа

Дедуктивно-параллельный метод обратного моделирования неисправностей ориентирован на обработку комбинационных и последовательностных схем, которые содержат незначительный процент (<20%) сходящихся разветвлений.

Основная идея повышения быстродействия моделирования неисправностей связана с преобразованием СР в псевдывыходы в целях последующего применения процедуры суперпозиции для древовидных структур и их необработки в случае фиксации непроверяемости линий сходящихся разветвлений.

Стратегия ОДП-метода [2] моделирования неисправностей цифрового устройства с предварительным структурным анализом включает следующие шаги:

1. Идентификация линий сходящихся разветвлений, инвариантных по отношению к тест-векторам. Вычислительная сложность данной процедуры $Q_T = n^2$, но она выполняется на стадии предварительного анализа и только один раз, поэтому практически не влияет на быстродействие моделирования тест-векторов.
2. Моделирование неисправностей линий СР на тест-векторе. Модификация схемной структуры путем преобразования сходящихся разветвлений в псевдывыходы цифрового устройства.
3. Вычисление линий подграфов схемы, моделирование неисправностей которых на тест-векторе не должно проводиться вследствие существования формального доказательства их непроверяемости.
4. Определение древовидных фрагментов графа цифрового устройства, корректных для выполнения суперпозиции кубов неисправностей элементов на тест-векторе.
5. Выполнение процедуры суперпозиции векторов проверяемых неисправностей примитивов на скорректированной модели цифрового устройства.

Итак, необходимым условием эффективного применения ОДП-метода [2] является определение множества сходящихся разветвлений V^{RC} на фоне несходящихся V^N , а также линий, охваченных глобальными обратными связями V^O . Поэтому структурная модель цифровой схемы представлена в виде следующих уравнений:

$$\begin{aligned} V &= \{V^{RC}, V^N, V^O, V^C\}; \\ \{V^{RC}, V^N, V^O\} \cup V^C &= V; \\ \{V^{RC}, V^N, V^O\} \cap V^C &= \emptyset; \\ V^{RC} \cap V^N &= \emptyset; V^{RC} \cup V^N = V^R; \\ V^O \cap V^{RC} &\neq \emptyset, \end{aligned} \quad (1)$$

где V^C — линии, дополняющие $\{V^{RC}, V^N, V^O\}$ до полного множества; V^R — множество линий разветвления.

Наличие подмножества V^O , делающего схему последовательностной, значительно усложняет алгоритмы поиска СР. Поэтому их необходимо рассматривать отдельно для двух типов объектов — комбинационных и последовательностных схем.

3. Метод структурного моделирования

Рассматривается алгоритм определения сходящихся разветвлений на основе анализа графовой структуры комбинационного цифрового устройства. Предельная сложность выполнения алгоритма равна квадрату числа линий в схеме. Для комбинационных устройств, которые не имеют контуров — глобальных обратных связей, основные шаги представлены пунктами:

1. Разбиение всех линий V (1) цифрового устройства на подмножества:

$$V = (V^Y, V^S, V^R), \quad (2)$$

где Y — идентификатор линии, относящейся к внешним выходам; S — обозначение линии, имеющей одного преемника, соединенной с одним элементом; R — идентификатор линии разветвления, имеющей более одного элемента-преемника.

Определение множества преемников для каждой линии схемы путем формирования вектора числа преемников:

$$\begin{aligned} V &= (V_1, V_2, \dots, V_i, \dots, V_n), \\ V_i &= \begin{cases} 0 \leftarrow V_i \in V^Y; \\ 1 \leftarrow V_i \in V^S; \\ \geq 2 \leftarrow V_i \in V^R, \end{cases} \end{aligned} \quad (3)$$

2. Выбор очередной линии $V_i \in V^R$ для определения ее принадлежности к множеству сходящихся разветвлений $V_i^R \in V^{RC}$. Данная процедура выполняется путем логического моделирования (прямой импликации) графовой структуры от линии $V_i^R \in V$ на множество всех ее преемников $f(V_i^R) \subseteq V$ до внешних выходов схемы. Первоначально все линии обнуляются $\forall_{i=1}^n V_i = 0$.

3. Присвоение рассматриваемой линии разветвления значения 1: $V_i^R = 1$. После этого реализуется последовательность операций $V_j = V_j + 1 \leftarrow V_j = f(V_i^R)$, ($j = \overline{1, n}; j \neq i$) для всех линий V_j , являющихся преемниками для V_i : $V_j = f(V_i^R)$. Если на некотором шаге будет зафиксирован результат $V_j = V_j + 1 = 2$, являющийся критерием сходимости для разветвления V_i^R , то оно заносится в список $V_i^R \in V^{RC}$ и осуществляется переход к пункту 2.

Повторение пунктов 2 и 3 выполняется для всех линий разветвления.

В качестве примера применения структурного анализа выступает схема, представленная на рис. 1, где линии 15, 17, 19 есть сходящиеся разветвления.

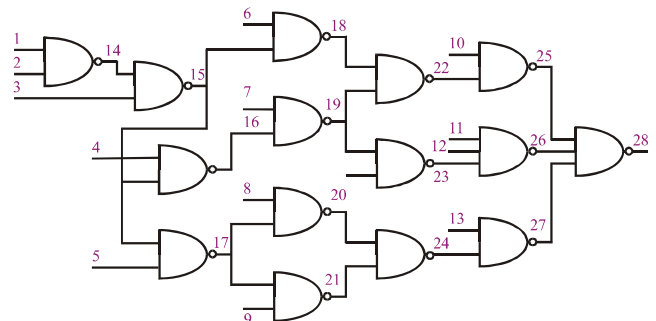


Рис. 1. Схема с разветвлениями

В результате выполнения структурного моделирования по описанному алгоритму схема реконфигурируется в четыре древовидные структуры, определяемые подграфами с корневыми вершинами, являющимися выходами или псевдывыходами устройства (рис. 2).

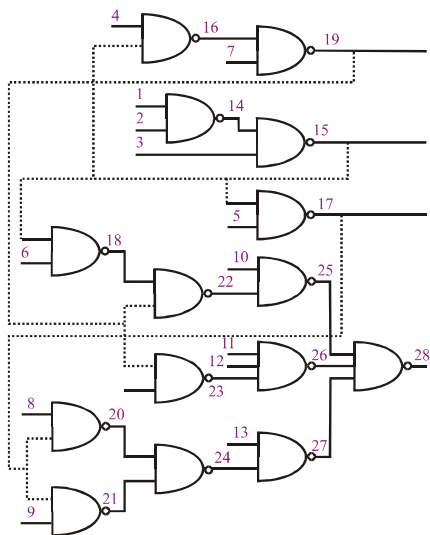


Рис. 2. Древоподобные фрагменты схемы

Моделирование неисправностей такой схемы с помощью обратной суперпозиции требует уже линейных затрат памяти и времени в функции от числа эквивалентных линий и квадратичных затрат для обработки СР: $Q = (r^2 / W) + n_r + n_p + (n - r - r^0)$, где (r^2 / W) – время моделирования неисправностей r сходящихся разветвлений, число которых определяется как $r = 0.2 \times n$; $n_r = n$ – время реконфигурирования примитивов схемы на входном наборе; $n_p = n$ – время поиска подграфов линий, соответствующих непроверяемым сходящимся разветвлениям; $(n - r - r^0) = n - 0.2 \times n - 0.4 \times n = 0.4 \times n$ – время выполнения процедуры суперпозиции на множестве линий схемы без СР и предшественников для непроверяемых сходящихся разветвлений. Учитывая фактические значения указанных параметров в функции от числа линий схемы, можно получить следующую оценку быстродействия BDP-метода:

$$Q = [(0.2 \times n)^2 / W] + n + n + (n - 0.2 \times n - 0.4 \times n) = [(0.2 \times n)^2 / W] + 2.4 \times n.$$

Таким образом, выигрыш в быстродействии предложенного метода тем больше, чем меньше процент СР в схеме цифрового устройства.

4. Метод анализа матрицы смежностей

Основан на использовании матрицы смежностей, описывающей граф цифрового устройства. Пусть имеется ориентированная структура, представленная на рис. 3.

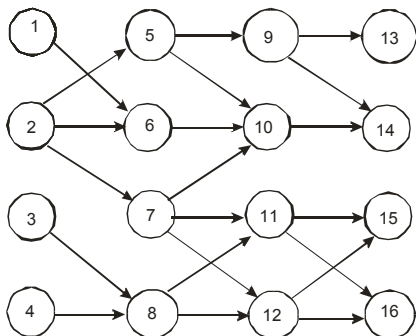


Рис. 3. Граф со сходящимися разветвлениями

Ее описание представляется в виде матрицы смежностей $M = [M_{ij}](i, j = \overline{1, n})$:

M_{ij}	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

где n – число вершин графа, а элемент или координата матрицы определяется так:

$$M_{ij} = \begin{cases} 1 \leftarrow \exists U_{ij}; \\ 0 \leftarrow \bar{\exists} U_{ij}. \end{cases}$$

Здесь U_{ij} – ориентированная дуга, соединяющая смежные вершины V_i и V_j . Метод выявления СР основывается на анализе матрицы смежностей, которая имеет существенную для предлагаемого ниже алгоритма информацию, расположенную над главной диагональю, что является следствием ориентированности графа. Алгоритмическая реализация метода анализа матрицы смежностей представлена на рис. 4.

В целях уточнения функций блоков ниже представлено описание особенностей алгоритмической реализации метода выявления СР:

- 1-2. Цикл по строкам матрицы в целях ее анализа: $i = 1, n$. После его окончания – конец алгоритма.
3. Присвоение начальных значений: S – вектору идентификации разветвлений, K – параметру количества единичных координат в вектор-строке.
4. Анализ элементов текущей строки путем выполнения цикла $j = i + 1, n$. После завершения цикла ($j > n$) осуществляется переход к п. 7.
5. Проверка условия равенства текущего элемента строки M_{ij} единице:
 - 5.1. Если $M_{ij} = 1$, то выполняется п. 6.
 - 5.2. Иначе – переход к п. 2.
6. Инкремент j -й координаты вектора идентификации разветвлений ($S_j = S_j + 1$) и подсчет количества единичных координат в вектор-строке S : $K := K + M_{ij}$. Переход к п. 2.
7. Проверка условия $K > 1$: если количество единичных координат K вектор-строки S больше единицы ($K > 1$), выполняется п. 8. Иначе – переход к п. 2.
8. Сообщение: в вершине i имеется разветвление. Считывание меток столбцовых координат j_1, j_2, \dots, j_k из массива S и организация цикла по ним.
9. Присвоение начальных значений: вектор-строке R идентификации сходящений ($R := 0$) и его j -м координатам ($R_j := 0$).

Сходящемуся разветвлению могут предшествовать вершины (линии), которые не являются таковыми, но проявляются на вершине-схождении как СР. Примером таких линий есть вершины, составляющие путь без ветвлений, заканчивающийся на линии СР.

Утверждение 3. Для того чтобы устранить всех предшественников сходящегося разветвления, входящих в множество V^j для вершины V_j , но не являющихся СР, необходимо и достаточно вычистить из множества V^j объединение пересечений всех пар сочетаний $C_{n_j}^2$.

Процедура дедуктивного анализа: если граф – ориентированный, без контуров глобальных обратных связей, то стратегия поиска сходящихся разветвлений сводится к одноразовому проходу по всем его вершинам на основе процедуры дедуктивной обработки каждой из них, включающей операции:

$$\begin{aligned} 1) V^j &= \bigcup_{C_{n_j}^2} [f_p^{-1}(V_j) \bigcap_{q=i+1, n_j-1}^{p=i} f_q^{-1}(V_j)]; \\ 2) V^j &= V^j \setminus \bigcup_{C_{m_j}^2} [V_j^p \bigcap_{q=i+1, m_j-1}^{p=i} V_j^q]; \\ 3) V_j &= [\bigcup_{i=1}^{n_j} f_i^{-1}(V_j) \cup V_j] \setminus V^j, \end{aligned} \quad (5)$$

где $f_i^{-1}(V_j)$ – элемент прообраза $f^{-1}(V_j)$, число которого для вершины V_j равно n_j .

Первое уравнение предназначено для определения СР путем выполнения операции объединения пересечений всех пар прообразов анализируемой вершины, число которых равно количеству сочетаний – $C_{n_j}^2$. Второе – для исключения из списка СР вершин, не являющихся сходящимися разветвлениями в соответствии с утверждением 3. Третье – для формирования списка предшественников каждой вершины.

Для примера графа (см. рис. 5) последовательная обработка всех вершин по правилам (5) представлена результатами вычислений:

$$\begin{aligned} V^1 &= \emptyset; V_1 = \{1\}; \\ V^2 &= \emptyset; V_2 = \{2\}; \\ V^3 &= \emptyset; V_3 = \{3\}; \\ V^4 &= (V_1 \cap V_2) = \emptyset; V_4 = \{1, 2, 4\}; \\ V^5 &= (V_1 \cap V_2) \cup (V_1 \cap V_3) \cup (V_2 \cap V_3) = \emptyset; V_5 = \{1, 2, 3, 5\}; \\ V^6 &= (V_2 \cap V_3) = \emptyset; V_6 = \{2, 3, 6\}; \\ V^7 &= (V_4 \cap V_5) = \{1, 2\}; V_7 = \{1, 2, 3, 4, 5, 7\} \setminus \{1, 2\} = \{3, 4, 5, 7\}; \\ V^8 &= (V_4 \cap V_6) = \{2\}; V_8 = \{1, 2, 3, 4, 6, 8\} \setminus \{2\} = \{1, 3, 4, 6, 8\}; \\ V^9 &= \emptyset; V_9 = \{2, 3, 6, 9\}. \end{aligned}$$

Объединение всех подмножеств $V^{RC} = \bigcup_{j=1}^n V^j$ дает полное множество сходящихся разветвлений V^{RC} графа цифровой структуры. Для данного примера $V^R = V^1 \cup V^2 \cup V^3 \cup V^4 \cup V^5 \cup V^6 \cup V^7 \cup V^8 \cup V^9 = \emptyset \cup \emptyset \cup \emptyset \cup \emptyset \cup \emptyset \cup \emptyset \cup \emptyset \cup \{1, 2\} \cup \{2\} \cup \emptyset = 1, 2$.

Учитывая тот факт, что линии, уже определенные как сходящиеся разветвления, являются избыточными для дальнейшего анализа структуры, их можно исключать из списков при обработке каждой вершины. Однако множество СР, определяемое в процессе обработки, необходимо накапливать в составе отдельного подмножества V^{RC} . С учетом сказанного возникает необходимость модификации уравнения (1) в (5). В этом случае выражение процедуры дедуктивного анализа вершин графа цифровой схемы модифицируется к виду

$$\begin{aligned} 1) V^j &= \bigcup_{C_{n_j}^2} [f_p^{-1}(V_j) \bigcap_{q=i+1, n_j-1}^{p=i} f_q^{-1}(V_j)]; \\ 2) V^j &= V^j \setminus \bigcup_{C_{m_j}^2} [V_j^p \bigcap_{q=i+1, m_j-1}^{p=i} V_j^q]; \\ 3) V^{RC} &= V^{RC} \cup V^j; \\ 4) V_j &= [\bigcup_{i=1}^{n_j} f_i^{-1}(V_j) \cup V_j] \setminus V^{RC}. \end{aligned} \quad (6)$$

Отличие процедуры (6) от (5) заключается в накоплении множества СР и его вычитании из списков предшественников для каждой вершины. Выражение (6) следует использовать для уменьшения списков прообразов обрабатываемой вершины, которое будет тем существенней, чем выше коэффициенты разветвления у линий предшественников, уже отнесенных к множеству V^{RC} .

6. Метод параллельного структурного моделирования графов на дедуктивной модели вершин

В качестве идентификатора вершины V_j выступает уже вектор $V_j \in V$, имеющий в j -й позиции значение 1. Тогда совокупность векторов, обозначающих все множество вершин в графе, будет представлена единичной матрицей

$$V = \|V_{ij}\|_{(i, j=\overline{1, n})}; V_{ij} = \begin{cases} 1 & \leftarrow i = j; \\ 0 & \leftarrow i \neq j. \end{cases}$$

Учитывая изоморфизм физм булевой алгебры и алгебры множеств, выполним преобразование выражения (5) к алгебро-логическому виду:

$$\begin{aligned} 1) V^j &= \bigvee_{C_{n_j}^2} [f_p^{-1}(V_j) \bigwedge_{q=i+1, n_j-1}^{p=i} f_q^{-1}(V_j)]; \\ 2) V^j &= V^j \wedge \neg [\bigvee_{C_{m_j}^2} (V_j^p \bigwedge_{q=i+1, m_j-1}^{p=i} V_j^q)]; \\ 3) V_j &= [\bigvee_{i=1}^{n_j} f_i^{-1}(V_j) \vee V_j] \wedge \bar{V}^j, \end{aligned} \quad (7)$$

где операции конъюнкции и дизъюнкции выполняются над соответствующими векторами матрицы V . Аналогичные преобразования над (6) дают результат:

$$\begin{aligned}
1) V^j &= \bigvee_{C_{n_j}^2} [f_p^{-1}(V_j) \bigwedge_{i=1, n_j-1}^{p=i} f_q^{-1}(V_j)]; \\
2) V^j &= V^j \wedge \neg [\bigcup_{C_{m_j}^2} (V_j^p \bigwedge_{i=1, m_j-1}^{p=i} V_j^q)]; \\
3) V^{RC} &= V^{RC} \bigvee V^j; \\
4) V_j &= [\bigvee_{i=1}^{n_j} f_i^{-1}(V_j) \bigvee V_j] \wedge \bar{V}^{RC}.
\end{aligned}
\tag{8}$$

Преимущества использования формул (7) и (8) заключаются в следующем: 1) Параллельность выполнения операций над вершинами графа позволяет в десятки и сотни раз повысить быстродействие процедуры анализа для вершин, имеющих большое число предшественников, путём увеличения памяти, необходимой для хранения матрицы идентификаторов вершин. 2) Повышение быстродействия формирования списка CP, благодаря исключению операций над элементами множества, которые заменяются параллельными операциями над векторами – идентификаторами вершин.

В качестве примера применения формул (8) приводится последовательность вычислений для нахождения множества CP графа (см. рис. 5):

1. Первоначальное задание матрицы V^0 :

V^0	1	2	3	4	5	6	7	8	9
1	1	0	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0
3	0	0	1	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0
5	0	0	0	0	1	0	0	0	0
6	0	0	0	0	0	1	0	0	0
7	0	0	0	0	0	0	1	0	0
8	0	0	0	0	0	0	0	1	0
9	0	0	0	0	0	0	0	0	1

и инициализация вектора сходящихся разветвлений:

$$V^{RC} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

2. Последовательная обработка вершин, заданных строками матрицы, в соответствии со структурой графа приводит к формированию результирующей матрицы V^1 :

V^1	1	2	3	4	5	6	7	8	9
1	1	0	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0
3	0	0	1	0	0	0	0	0	0
4	1	1	0	1	0	0	0	0	0
5	1	1	1	0	1	0	0	0	0
6	0	1	1	0	0	1	0	0	0
7	0	0	1	1	1	0	1	0	0
8	1	0	1	1	0	1	0	1	0
9	0	1	1	0	0	1	0	0	1

3. Накопление CP в процессе обработки вершин приводит к окончательному результату в виде вектора V^{RC} :

$$V^{RC} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

где значение 1 свидетельствует о принадлежности линии к подмножеству сходящихся разветвлений.

7. Структурный анализ последовательных схем

Особенность анализа последовательных схем методом структурного моделирования связана с получением идентичных списков прообразов для каждой вершины, входящей в контур глобальной обратной связи.

Здесь прежде всего речь идет о синхронных устройствах, основное отличие которых от асинхронных заключается в отсутствии глобальных обратных связей, которые практически не применяются при проектировании сложных цифровых систем ввиду наличия в таких схемах состязаний сигналов. Поэтому далее рассматриваются схемы, имеющие локальные обратные связи, являющиеся неизменным атрибутом любого триггера, который может превратить комбинационную схему в элемент памяти. Примером описания триггера может служить система уравнений, представленная в формате BNF (Boolean Normal Form):

$$S2_tmp0 = C_tmp0 * Q3_tmp0(t-1);$$

$$Q2_tmp0(t) = !S2_tmp0 + (D_tmp0 * Q2_tmp0(t-1));$$

$$R3_tmp0 = !(D_tmp0 * Q2_tmp0(t-1));$$

$$Q3_tmp0(t) = !C_tmp0 + (R3_tmp0 * Q3_tmp0(t-1));$$

$$Q(t,0) = !Q3_tmp0(t) + (Q2_tmp0(t) * Q(t-1)).$$

Данному описанию ставится в соответствие логическая схема триггера, имеющая три обратные связи (рис. 6).

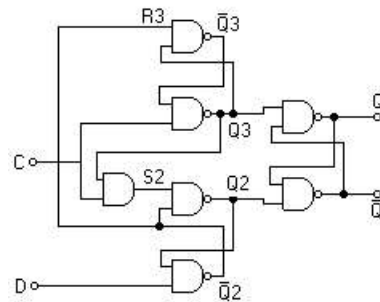


Рис. 6. Пример триггерной структуры

Графовая модель триггерной структуры, представленная на рис. 7, содержит 8 линий, среди которых – 3 сходящихся разветвления (C, Q3, Q2) и 6 линий, охваченных обратными связями (Q3, Q-bar_3, Q2, Q-bar_2, Q, Q-bar).

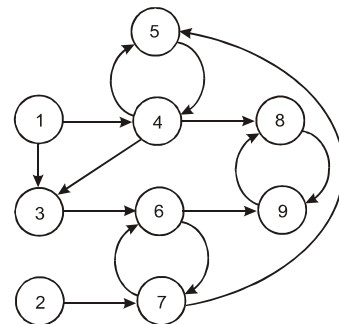


Рис. 7. Граф триггерной структуры: 1 – C, 2 – D, 3 – S2, 4 – Q3, 5 – Q-bar_3, 6 – Q2, 7 – Q-bar_2, 8 – Q, 9 – Q-bar

В соответствии с утверждениями 1 и 2 критерием отличия линии, входящей в контур, от сходящегося разветвления является наличие идентичных экстр-

раобразов для более, чем одной вершины графа цифрового устройства.

Однако проблема заключается в том, что контурные вершины могут быть сходящимися разветвлениями. Как в этом случае рассматривать задачу определения СР? Существует два возможных варианта ее решения:

1) Относить все контурные вершины к множеству СР.

2) Считать контурную вершину сходящимся разветвлением, если она имеет не менее двух исходящих дуг.

В первом случае значительно увеличится число СР, если схема имеет большое число триггеров. Второй вариант более предпочтителен, поскольку в реальных триггерных структурах он отсекает значительное число линий, входящих в контуры. Таким образом, следующее определение дает возможность идентифицировать СР для цифровых схем с контурами.

Определение. Для синхронных схем, имеющих обратные связи благодаря наличию триггеров, контурная линия является СР, если она имеет более одной исходящей дуги, а также схождение на линии, не принадлежащей к данному контуру.

Для структуры, представленной на рис. 8, сходящимися разветвлениями являются линии (3,4), входящие в контур и имеющие схождение на линиях 5 и 6, не принадлежащих к контуру, образованному линиями 3 и 4.

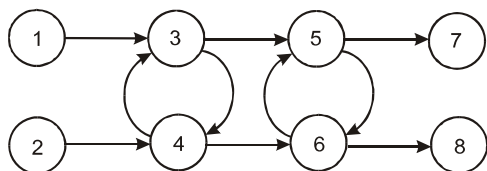


Рис. 8. Граф с контурами

Что касается ранее рассмотренного графа (см. рис. 6), то здесь в качестве СР выступают вершины С, Q3, Q2, которые удовлетворяют всем требованиям определения.

8. Заключение

Предложены методы структурного анализа цифровых систем, которые используются на стадии предварительной обработки схем в целях повышения быстродействия методов моделирования неисправностей по стратегии обратного прослеживания дефектов. Главные методы анализа основаны на дедуктивном подходе к структурному моделированию графа, который позволяет определять все сходящиеся разветвления в цифровых схемах за один проход. Программная реализация структурного анализа позволяет в десятки раз повысить быстродействие нахождения СР по сравнению с классическими методами и тем самым в несколько раз улучшить количественные показатели быстродействия программных средств моделирования неисправностей.

Дальнейшее развитие структурного анализа для моделирования неисправностей цифровых схем должно охватывать вопросы, связанные с:

1) исследованием компенсации и возникновения ложных дефектов для всех промышленно используемых типов триггеров;

2) теоретическим доказательством корректности выполнения обратной суперпозиции неисправностей для триггерных структур;

3) исследованием проблемы структурного анализа для асинхронных схем с глобальными обратными связями.

Литература: 1. Хаханов В.И., Сысенко И.Ю., Колесников К.В. Дедуктивно-параллельный метод моделирования неисправностей на реконфигурируемых моделях цифровых систем // Радиоэлектроника и информатика. 2002. №1. С. 98-105. 2. Хаханов В.И., Колесников К.В., Хаханова А.В. BDP-метод моделирования неисправностей для синтеза тестов цифровых проектов // Радиоэлектроника и информатика. 2002. №2. С. 60-66. 3. Hahanov V.I., Babich A.V., Hyduke S.M. Test Generation and Fault Simulation Methods on the Basis of Cubic Algebra for Digital Devices. Proceedings of the Euromicro Symposium on Digital Systems Design DSD2001. Warsaw, Poland. September, 4-6, 2001. P. 228-235. 4. Хаханов В.И., Хак Х.М. Джахирул, Масуд М.Д. Мехеди. Модели анализа неисправностей цифровых систем на основе FPGA, CPLD // Технология и конструирование в электронной аппаратуре. 2001. № 2. С. 3-11. 5. Хаханов В.И., Сысенко И.Ю., Хак Х.М. Джахирул, Масуд М.Д. Мехеди. Кубическое моделирование неисправностей цифровых проектов на основе FPGA, CPLD // Радиоэлектроника, информатика, управление. 2001. № 1. С. 123-129. 6. Abramovici M., Breuer M.A. and Friedman A.D., Digital System Testing and Testable Design, Computer Science Press, 1998. 652 p. 7. Хаханов В.И. Техническая диагностика элементов и узлов персональных компьютеров. К.: ИЗМН. 1997. 308с. 8. Убар Р.Р. Анализ диагностических тестов для комбинационных цифровых схем методом обратного прослеживания неисправностей // Автоматика и телемеханика. 1977. №8. С.168-176.

Поступила в редколлегию 16.05.2002

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

Хаханов Владимир Иванович, д-р техн. наук, профессор кафедры АПВТ ХНУРЭ. Научные интересы: техническая диагностика вычислительных устройств, систем, сетей и программных продуктов. Увлечения: баскетбол, футбол, горные лыжи. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 40-93-26. E-mail: hahanov@kture.kharkov.ua

Чумаченко Светлана Викторовна, канд. физ.-мат. наук, доцент кафедры АПВТ ХНУРЭ. Научные интересы: методы математической физики и дискретной математики в автоматизации проектирования радиоэлектронных устройств. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 40-93-26. E-mail: ri@kture.kharkov.ua

Колесников Константин Васильевич, ст. преподаватель кафедры КС Черкасского государственного технологического университета. Научные интересы: тестирование цифровых систем и сетей. Увлечения: живопись – постимпрессионизм, поэзия – серебряный век, символизм. Адрес: Украина, 18006, Черкасы, бульв. Шевченко, 460, тел. 43-74-28, e-mail: kvvk601@mail.ru

Хаханова Анна Владимировна, магистрант кафедры АПВТ ХНУРЭ. Научные интересы: моделирование и диагностика цифровых устройств. Увлечения: аэробика, литература. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 40-93-26.