

СИНТЕЗ САМОПРОВЕРЯЕМОГО КОМБИНАЦИОННОГО ДЕТЕКТОРА РАВНОВЕСНЫХ КОДОВ

Предлагается метод синтеза комбинационного детектора (m, n) -кодов. Метод основан на разложении кодов по заданному разбиению компонент кодов на два подмножества. Многократное применение разложения приводит к представлению всевозможных (m, n) -кодов формулой над множеством функций $\{\wedge, \vee, D_p^q(X')\}$. Здесь $D_p^q(X')$ – ДНФ, представляющая всевозможные (q, p) -коды в подмножестве переменных X' , $X' \subset X$, где $X = \{x_1, \dots, x_n\}$ – множество переменных, сопоставляемых входам детектора, $q \leq m$, $p < n$. Обсуждается возможность синтеза детектора в базе FPGA (field of programmable gate array), а также проблема обеспечения его самопроверяемости в классе неисправностей, обусловленных этим базисом.

Самопроверяемые детекторы комбинационного типа, описанные в работах [1, 2] ориентированы на относительно небольшое число равновесных кодов (или кодов Бергера), достигаемых проверяемым дискретным управляющим устройством на его выходах в исправном состоянии. При этом в работе [1] допускаются как равновесные коды, так и коды Бергера, предполагается реализация детектора на ПЛМ. В работе [2] вообще не требуется кодировать выходы проверяемого устройства, однако предполагается специальная двухуровневая факторизационная реализация совместно управляющего устройства и детектора.

В нашей работе предлагается компактная, по сравнению с [1, 2], реализация самопроверяемого комбинационного детектора, сложность которого линейно зависит от длины n равновесного кода. Детектор реализует все (m, n) -коды, либо их подмножества, содержащие (m, n) -коды, достигаемые на выходах проверяемого устройства; его реализация ориентирована на FPGA (field of programmable gate arrays) технологию [3]. Никаких ограничений на реализацию проверяемого дискретного управляющего устройства не накладывается. Важно лишь, чтобы его неисправности проявляли себя на выходах устройства монотонным образом. К выходам проверяемого управляющего устройства подключены входы детектора (рис. 1). Выходы $y_{m+1}, \dots, y_s$ обеспечивают равновесность кодов.

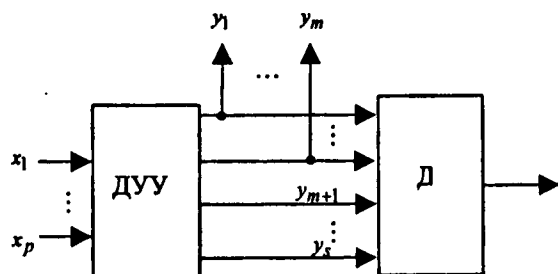


Рис. 1

Пусть детектор реализует функцию f , принимающую значение 1 на (m, n) -коде и 0 в остальных случаях. Дискретное управляющее устройство (ДУУ) проявляет любую неисправность из рассматриваемого класса его неисправностей монотонным образом. Имеется в виду следующее. Пусть α – произвольный тестовый набор для некоторой неисправности w , $w \in W$, β, β^w – выходы исправного и неисправного устройств соответственно, сопоставляемые α . В случае монотонного проявления неисправностей из W β, β^w – сравниваемые векторы (наборы), т.е. либо $\beta \leq \beta^w$, либо $\beta^w \leq \beta$. Это значит, что для любого α и

любой неисправности w мы имеем на выходах устройства вектор β^w , не являющийся (m, n) -кодом. Такой вектор обращает в 0 функцию f , реализуемую детектором. Следовательно, любое проявление неисправности из W немедленно обнаруживается детектором.

Представим функцию, реализуемую детектором, в виде ДНФ. Она является совершенной, одновременно сокращенной, кратчайшей и минимальной, так как склеивания между конъюнкциями совершенной ДНФ, реализующей (m, n) -коды, невозможны. Обозначим ее через $D_n^m = k_1 \vee k_2 \vee \dots \vee k_{C_n^m}$.

Любые две конъюнкции k_i, k_j из D_n^m ортогональны не менее чем по двум переменным. Более того, они инверсно ортогональны не менее чем по двум переменным.

Конъюнкции k_i, k_j инверсно ортогональны по переменным x_i, x_j , если в k_i x_i содержится без инверсии, а в k_j – с инверсией; в то же время x_j содержится в k_i с инверсией, а в k_j – без инверсии. Например, конъюнкции $\bar{x}_1 \bar{x}_2 x_3 x_4, x_1 \bar{x}_2 \bar{x}_3 x_4$ ДНФ D_n^m , представляющей $(2, 4)$ -коды, инверсно ортогональны по x_1 и x_3 .

Число конъюнкций D_n^m , реализующей всевозможные (m, n) -коды, равно C_n^m , что уже при $n=10$, $m=5$ достигает 252, а число букв – 1260. Использовать такую ДНФ в качестве задания на синтез для последующего применения известных двухуровневых [4] и многоуровневых [5] методов синтеза не представляется целесообразным ввиду громоздкости получаемых в результате схем.

Предлагается специальный метод разложения D_n^m по подмножествам переменных.

1. Декомпозиционный метод синтеза детектора равновесных кодов

1.1. Построение скобочной формы A

Представим $D_n^m(X), X = \{x_1, \dots, x_n\}$ формулой A над множеством функций $\wedge, \vee, D_p^q(X')$, причем $p \leq k, k < n, X' \subset X$. Здесь k – число входов CLB (configurable logical block) – схем, используемых в FPGA-технологии.

Разобьем переменные множества X ДНФ D_n^m на два подмножества:

$$X^1 = \{x_1, \dots, x_k\}, X^2 = \{x_{k+1}, \dots, x_n\}.$$

Теорема 1.

$$D_n'' = \bigvee_{\substack{i,j \\ i+j=m}} D_k'(X^1) D_{n-k}^j(X^2), \quad (1)$$

$$i \in \{0, 1, \dots, k\}, j \in \{0, 1, \dots, n-k\}.$$

Доказательство. Возьмем произвольный (m, n) -код, представим его вектором α . Подставим α в левую и правую части выражения (1). Покажем, что соотношение (1) выполняется на этом векторе, $D_n''(\alpha) = 1$ по построению ДНФ D_n'' .

Подставим α в правую часть соотношения (1) $\alpha = \alpha^1 \alpha^*$, где α^1 – составляющая кода по переменным X^1 , имеющая вес (число единичных компонент) s ; α^* – составляющая по переменным X^* , имеющая вес t , $s + t = m$. Среди слагаемых правой части соотношения (1) найдется произведение $D_k'(X^1) D_{n-k}^j(X^*)$, так как (1) содержит все произведения, для которых $i + j = m$. Следовательно, $D_k'(\alpha^1) D_{n-k}^j(\alpha^*) = 1$. Поскольку α – произвольный вектор, утверждение теоремы доказано.

В (1) выражения $D_{n-k}^j(X^*)$ будем называть коэффициентами разложения.

Если $(n-k) > k$, то к каждому коэффициенту разложения $D_{n-k}^j(X^*)$ снова применим разложение (1) до тех пор, пока каждая ДНФ в получающейся таким образом скобочной форме будет зависеть не более чем от k переменных и, следовательно, не подлежит дальнейшему разложению. Полученное выражение и есть формула A .

Построим скобочную формулу A для ДНФ D_6^3 , положив $k = 2$. Выполним первый шаг разложения

$$X^1 = \{x_1, x_2\}, X^* = \{x_3, x_4, x_5, x_6\},$$

$$D_6^3 = (D_2^0(X^1) D_4^3(X^*) \vee D_2^1(X^1) D_4^2(X^*) \vee D_2^2(X^1) D_4^1(X^*)).$$

Продолжим разложение, положив

$$X^2 = \{x_3, x_4\}, X^3 = \{x_5, x_6\};$$

$$A = D_6^3 = (D_2^0(X^1) (D_2^1(X^2) D_2^2(X^3)) \vee D_2^2(X^2) D_2^1(X^3)) \vee D_2^1(X^1) (D_2^0(X^2) D_2^2(X^3) \vee D_2^1(X^2) D_2^1(X^3) \vee D_2^2(X^2) D_2^0(X^3)) \vee D_2^2(X^1) (D_2^0(X^2) D_2^1(X^3) \vee D_2^1(X^2) D_2^0(X^3))).$$

Первому шагу разложения соответствуют скобки нулевой глубины формулы A , второму – скобки первой глубины.

1.2. Свойства формулы A

Для рассмотрения свойств формулы A воспользуемся представлением ее в виде дерева G_A , сопоставим ДНФ $D_p^q(X')$, $p \leq k$ концевые вершины. Считая операцию конъюнкции «главнее» операции дизъюнкции, мы опускали лишние скобки в A . В дереве G_A операциям конъюнкции будут сопоставлены соответствующие внутренние вершины, из которых исходит ровно два ребра.

Число ребер, исходящих из вершины, помеченной символом $\vee$, определяется соотношениями (1), применяемыми при построении A . Заметим, что ребро, исходящее из вершины, отмеченной $\vee$, не заканчивается концевой вершиной (листом). В то же время одно из ребер, исходящее из вершины, помеченной $\wedge$, обязательно заканчивается листом.

Будем упорядочивать концевые и внутренние вершины слева направо так, как они записаны в формуле A . Для рассматриваемого примера имеем рис. 2. Будем иметь в виду, что каждый шаг разложения порождает два яруса дерева G_A .

Листья одного и того же яруса дерева могут быть помечены одним и тем же выражением $D_p^q(X')$. Это дает возможность объединять такие вершины в одну и превращать дерево в граф. Однако здесь мы не будем этого делать, чтобы не усложнять представление формулы A .

Выделим в G_A существенное поддерево H_i [6], двигаясь от листьев нижнего яруса дерева. Включаем в H_i две концевые вершины (два листа), смежные с одной и той же неконцевой вершиной, помеченной символом $\wedge$. Добавим инцидентное ей ребро, включив вершину, отмеченную символом $\vee$. Двигаясь далее к корню дерева по инцидентному ей ребру, включаем вершину, помеченную символом $\wedge$ вместе с другим, инцидентным ей ребром, помеченным концевой вершиной вида $D_p^q(X')$ и т.д., пока не дойдем до корня. На рис. 2 существенные поддерева, имеющие общие вершины и ребра, выделены пунктирными линиями, а отдельно отстоящие от них – жирными линиями.

Предложение 1. Каждому существенному поддереву сопоставляется логическое произведение вида

$$D_k^i(X^1) D_k^j(X^2), D_k^i(X^*) D_p^{i_1 \dots i_s}(X^{s+1}),$$

где $p \leq k$, s – число шагов разложения, $i_1 + i_2 + \dots$

$$+ i_{s+1} = m, X^1 \cup X^2 \cup \dots \cup X^{s+1} = X.$$

Получение формулы A путем разложения (1) ДНФ $D_n''(X)$ на фрагменты $D_p^q(X')$, $p \leq k$ связано с возможностью реализации таких фрагментов так называемыми конфигурационными логическими блоками CLB (configurable logical block) в рамках LUT based FPGA-технологии. Будем полагать $k = 4$ или $k = 8$ [Xilinx]. G_A (формула A) порождает разбиение X на $(s+1)$ подмножеств. Назовем комбинацию чисел $i_1, \dots, i_{s+1}$ распределением весов булева вектора по заданному разбиению его компонент на подмножества или просто распределением весов по разбиению. В нашем примере существенному поддереву, ребра которого отмечены жирной линией, сопоставляется произведение $D_2^2(X^1) \times D_2^0(X^2) D_2^1(X^3)$. Разбиение X представлено подмножествами

$$X^1 = \{x_1, x_2\}, X^2 = \{x_3, x_4\}, X^3 = \{x_5, x_6\},$$

распределение весов по разбиению для этого поддерева: $i_1 = 2, i_2 = 0, i_3 = 1$.

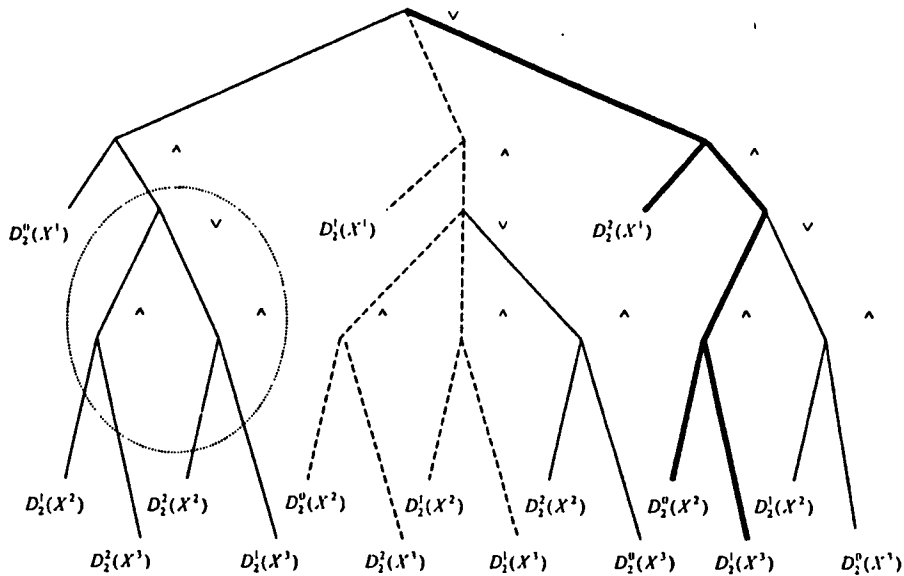


Рис. 2. Дерево G_A для ДНФ D_6^3 при $k = 2$

Предложение 2. Различным существенным поддеревам G_A сопоставляется одно и то же разбиение множества X на подмножества $X^1, \dots, X^{s+1}$.

Предложение 3. Различным существенным поддеревам G_A сопоставляются различные распределения весов по разбиению.

Предложение 4. Различные распределения весов по разбиению отличаются друг от друга не менее чем по двум компонентам.

Предложение 5. Существенное поддерево представляет функцию

$$D_k^1(X^1) \cdot D_k^2(X^2) \cdot \dots \cdot D_k^s(X^s) \cdot D_k^{s+1}(X^{s+1})$$

(произведение ДНФ), задающую все (m, n) -коды с распределением $(i_1, i_2, \dots, i_s, i_{s+1})$ весов. Назовем это произведение ДНФ характеристикой существенного поддерева. Характеристика существенного поддерева представляет подмножество (m, n) -кодов, которые могут быть представлены ДНФ, полученной перемножением соответствующих $D_p^q(X^r)$ этой характеристики.

Предложение 6. Каждому (m, n) -коду сопоставляется единственное существенное поддерево в G_A . Подстановка сопоставляемого (m, n) -коду булева вектора в формулу A обращает в единицу ДНФ, порожденную характеристикой поддерева G_A , и только эту ДНФ.

Предложение 7. Объединение (m, n) -кодов, представляемых характеристиками всех существенных поддерева G_A , есть множество всех (m, n) -кодов.

В [6] предложен метод перебора всех существенных поддерева. Зная множество $M(m, n)$ -кодов, достижимых на выходах проверяемого ДУУ, можно упростить дерево G_A (формулу A), исключая в G_A существенные поддерева, реализующие (m, n) -коды, не принадлежащие M . Операция исключения существенного поддерева сводится к замене на константу 0 пары $D_p^q(X^r)$ нижнего яруса дерева из характеристики этого поддерева.

Речь идет о $D_p^q(X^r)$, принадлежащих только рассматриваемому поддереву. Приписывание констант 0 конечным вершинам существенных поддерева приводит к упрощению G_A по правилам, описанным в [6].

1.3. Построение схемы самопроверяемого детектора по формуле A

Структура формулы A представлена деревом G_A (возможно, упрощенным). Каждый элемент (CLB), реализующий $D_p^q(X^r)$, удовлетворяет условию $p \leq k$. Здесь $k \leq 4$ или $k \leq 8$. Мы имеем в виду LUT-based FPGA-технологии [3].

Рассмотрим в схеме детектора неисправности следующего вида. Во-первых, одиночные константные неисправности на внешних полюсах CLB, затем одиночные константные неисправности на входах схемы и, наконец, неисправности, связанные с произвольным изменением функции, реализуемой одним из CLB схемы. Если CLB реализует две функции, то вторая функция исправна. Все эти неисправности образуют множество V .

Рабочей областью функционирования схемы детектора является подмножество $M(m, n)$ -кодов, достигаемых на выходах проверяемого дискретного управляющего устройства. Не все неисправности из V могут проявляться в M . Допускается накопление не более одной необнаружимой неисправности из V .

Неисправность в системе устройство – детектор должна обнаруживаться, если в схеме детектора появилась обнаружимая в M неисправность $v \in V$ и проверяемое дискретное управляющее устройство исправно. Возможна другая ситуация. Схема детектора исправна или в ней присутствует единственная необнаружимая неисправность. Неисправность дискретного управляющего устройства приводит к появлению неравновесного кода на его выходах, что обнаруживается детектором. Систему, удовлетворяющую таким условиям, назовем полностью самопроверяемой.

Обеспечим полную самопроверяемость системы управляющее устройство–детектор простым дублированием схемы C , реализующей A . В этом случае система приобретает вид, представленный на рис. 3.

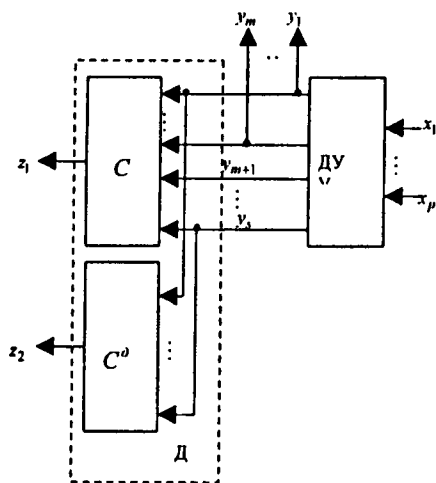


Рис. 3

Остановимся на реализации схемы C по формуле A (представляющему ее дереву G_A , в том числе упрощенному). Пусть каждая функция вида $D_p^q(X')$ реализуется на одном из выходов CLB. Один и тот же CLB можно использовать для реализации двух функций одного и того же множества переменных. Это необходимо учитывать при покрытии различных функций $D_p^q(X')$ с помощью CLB для сокращения числа CLB-элементов. Выражение вида

$$u_1 u_{k+2} \vee \dots \vee u_{k+1} u_{2(k+1)} \quad (2)$$

можно реализовать тремя CLB. В (2) переменные $u_1 \dots u_{k+1}$ сопоставляются функциям D_p^q , а переменные $u_{k+2} \dots u_{2(k+1)}$ либо функциям того же вида, либо скобкам одной и той же глубины. ДНФ (2) задана поддеревом, обведенным в G_A (рис. 2). Число конъюнкций в (2) может быть менее $(k+1)$ и определяется соотношением (1). Пусть $k=8$. С помощью CLB1 реализуем первые 4 конъюнкции из (2), затем с помощью CLB2 – следующие 4 конъюнкции и, наконец, CLB3 реализует функцию

$$\omega_1 \vee \omega_2 \vee u_{k+1} u_{2(k+1)}, \quad (3)$$

где ω_1, ω_2 – переменные, сопоставляемые выходам CLB1 и CLB2 соответственно.

Обозначим дубль схемы C через C^d , входы схем C, C^d общие. Схема и ее дубль образуют детектор D . Если на обоих выходах детектора достигается значение 1 – система исправна. Покажем, что система, представленная на рис. 2, полностью самопроверяемая.

2. Самопроверяемость системы ДУУ – детектор

Рассмотрим неисправности детектора из множества V и неисправности проверяемого ДУУ из мно-

жества W . Всякая $w \in W$ проявляет себя на выходах этого устройства монотонным образом.

Будем считать, что в системе устройство – детектор сначала возможно появление одной неисправности либо $v \in V$, либо $w \in W$. Если детектор исправен, $w \in W$ обнаружима в рабочей области функционирования устройства на выходах детектора.

Если первой появляется неисправность в детекторе, она может оказаться необнаружимой в системе устройство – детектор в рабочей области функционирования устройства. Тогда детектор накапливает эту неисправность. Будем предполагать, что в нем возможно накопление не более одной неисправности.

Обозначим через M^* объединение множества M со всеми M^* . Здесь M^* – множество векторов, достижимых на выходах ДУУ в присутствии неисправности w .

Рассмотрим одиночную константную неисправность v детектора на его входном полюсе. Обозначим их множество через $V'', V'' \subset V$.

Теорема 2. Одиночная константная неисправность на входном полюсе детектора обнаружима в M .

Доказательство. Предполагаем, что в M содержатся равновесные коды как с единичным значением переменной x_i , так и с нулевым значением переменной x_i , сопоставляемой неисправному полюсу (иначе эту переменную можно было бы исключить, сохранив равновесность кодов в M). Это значит, что при неисправности константа ноль вектор из M с единичным значением компоненты x_i заменяется на некодовый, что приводит к появлению на выходах детектора $z_1 z_2$ значений 00. Все элементы схемы детектора исправны. Аналогично при неисправности константа один, вектор из M с нулевым значением компоненты x_i заменяется на некодовый. Это приводит к появлению на выходах детектора $z_1 z_2$ значений 00. Теорема доказана.

Рассмотрим неисправность $v, v \in \{V \setminus V''\}$.

Теорема 3. Неисправность $v \in \{V \setminus V''\}$ обнаружима в M^* .

Доказательство. Рассматриваемая неисправность может проявить себя на некотором (m, n) -коде из M . Так как неисправность возможна только в одной из дублирующих подсхем детектора, то в этом случае на его выходах достигаются наборы 01 или 10.

Если неисправность не проявляет себя в M , то она остается необнаружимой в детекторе, накапливаясь в нем. При появлении монотонно проявляющейся неисправности в проверяемом ДУУ возможно маскирование некодового набора неисправной подсхемой детектора (другая дублирующая его подсхема исправна). Это приводит к появлению на выходах детектора наборов 01 или 10. Если маскирование не имеет места, тогда первый некодовый набор, реализуемый на выходах проверяемого ДУУ, обеспечивает значение 00 на выходах $z_1 z_2$ детектора. Теорема доказана.

Мы считаем, что каждая последующая неисправность $v \in V$, или $w \in W$ появляется после того, как рабочая область ДУУ в присутствии предыдущей неисправности исчерпана. При таком предположении $w \in W$ всегда обнаружится, а $v \in V$ может остаться необнаружимой. В связи с предполо-

жением о возможности накопления в системе устройство-детектор не более одной неисправности следующая неисправность должна быть обязательно в ДУУ. На основании теоремы 3 заключаем, что она обязательно проявится на выходах детектора в рабочей области функционирования ДУУ.

ЛИТЕРАТУРА

1. I. Levin and M. Karpovsky. On-line self-checking of microprogram control unit // 4th IEEE int. on-line testing workshop. Italy, Capri. July 1998. P. 152–156.
2. I. Levin and V. Sinelnikov, Self-checking of FPGA-based control units // Proceeding of 9th great lakes symposium on VLSI, Ann Arbor, Michigan, March 4–6, 1999, IEEE press. P. 292–295.
3. Xilinx, the programmable logic. Data book, 1996.
4. S. Baranov. Logic synthesis for control automata // Kluwer academic publishers, Dordrecht / Boston / London, 1994.
5. R.K. Brayton, R. Rudell, A. Sagiovanni-Vincentelli and A. R. Wang, MIS: A multiple-level logic optimization program. IEEE Trans. On computer-aided design. Nov. 1987. Vol. 7, P. 1062–1081.
6. Матросова А.Ю. Алгоритмические методы синтеза тестов. Томск: Изд-во ТГУ, 1990. 206 с.

Статья представлена кафедрой программирования факультета прикладной математики и кибернетики Томского государственного университета, поступила в научную редакцию 21 февраля 2000 г.

УДК 519.7

И.А. Панкратова

О СИСТЕМЕ ПРОГРАММ МОДЕЛИРОВАНИЯ ДИНАМИЧЕСКОГО ПОВЕДЕНИЯ ПЕРЕКЛЮЧАТЕЛЬНЫХ СХЕМ С ЗАДАВАЕМОЙ ТОЧНОСТЬЮ

Работа выполнена при финансовой поддержке РФФИ, грант 98-01-00288.

Рассматривается задача моделирования и анализа динамического поведения БИС на транзисторном уровне представления. Моделью БИС служит переключательная схема на полурешетках проводимости и состояний. Описываются алгоритмы и программы имитационного моделирования и анализа схемы с любой наперед заданной точностью.

1. Основные понятия

Переключательная схема служит моделью больших интегральных схем (БИС) и определяется в [1] как тройка конечных множеств (X, Z, Z_0) , где X – множество элементов схемы (которые, в свою очередь, являются схемами); $Z_0 \subseteq Z$ – множество полюсов схемы (среди которых выделяются полюсы источника питания VDD и GND и входные полюсы, через которые на схему подаются воздействия извне); Z – множество узлов схемы, представляющее собой разбиение на множестве полюсов всех ее элементов. Таким образом, схемы имеют иерархическую структуру, на самом нижнем уровне которой находятся базисные элементы – транзисторы различных типов, резистор. Транзистор представляет собой элемент с управляемой проводимостью и задается монотонной функцией проводимости $p=f(s)$; здесь s – состояние затвора транзистора; p – проводимость между его истоком и стоком, $p \in P$, $s \in S$; P и S – полурешетки проводимостей и состояний соответственно. В общем случае $P=\{0, 1, X, 0', 1', X', E\}$; здесь $0, 1, X$ – точки полурешетки, представляющие собой проводимости соответственно разомкнутой, замкнутой и резистивной цепи; остальные элементы – в подходящей степени неопределенные значения: $0'=1+X$, $1'=0+X$, $X'=0+1+X$ (полная неопределенность). Для некоторых классов схем достаточно рассматривать полурешетку $P' \subseteq P$; например, для КМОП-схем $P'=\{0, 1, X'\}$.

Состоянием узла является пара проводимостей от этого узла до полюсов источника питания, т.е. в общем

случае $S=P^2$. При этом $|S|=|P|^2=49$. Однако в ряде случаев нет необходимости различать все состояния; например, те из них, для которых совпадают значения функций проводимости всех базисных элементов, физически эквивалентны. Информация о неразличимых состояниях задается в виде эквивалентности R на множестве точек полурешетки P^2 , т.е. на множестве $\{00, 0X, 01, X0, XX, X1, 10, 1X, 11\}$. Затем по эквивалентности R строится полурешетка $\langle R \rangle$, элементами которой являются интервалы, совпадающие с объединениями некоторых смежных классов эквивалентности R . $\langle R \rangle$ является адекватной моделью P^2 (по теореме 1.4 в [1]), и $|\langle R \rangle| \leq |P^2|$. Замена полурешетки P^2 на $\langle R \rangle$ позволяет понизить мощность алфавита состояний без потери адекватности моделирования; R при этом играет роль точности модели. Например, для R , заданной классами $\{00\}$, $\{X0, 10\}$, $\{11, 1X\}$, $\{01, X1\}$, $\{0X, XX\}$, получается следующая полурешетка $\langle R \rangle$: $\{00, 0'0, 10', 1'1, 1'X, E0, 1'0', E0', EE\}$. Резистор представляет собой элемент с двумя полюсами и постоянной проводимостью между ними, равной X .

Назовем полным состоянием схемы набор состояний всех ее узлов, а входным состоянием – набор состояний всех входных полюсов. Так как каждое полное состояние содержит в себе и входное состояние, то будем обозначать: $r=aq$; r – полное состояние, a – соответствующее ему входное состояние, q – набор состояний всех остальных (не являющихся входными полюсами) узлов. Полное состояние r называется устойчивым, если оно соответствует вычисленным по нему полным проводимостям в схеме, т.е. при неизменных состояниях входных полюсов схема остается в устойчивом состоя-