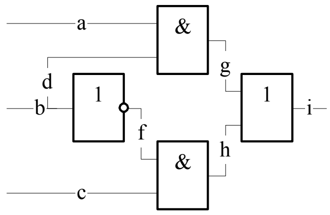
5.1.Метод таблиц истинности

Метод таблиц истинности относится к первой группе и, следовательно, использует моделирование ОД с неисправностями.

При этом анализируются все из 2^n входных наборов схемы, где n — число входов ОД.

После определения классов эквивалентных неисправностей выполняется моделирование, результатом которого является таблица функций различения (ТФР), задающая отношение τ. Затем находится минимальное покрытие этой таблицы.

Ниже приведен пример вычисления теста для схемы, изображенной на рис.5.3. Решение задачи определения классов эквивалентных неисправностей представлено в виде табл.5.2.



Набор	a b c	i
T_0	000	0
T_1	001	0
T_2	010	0
T_3	011	1
T_4	100	1
T_5	101	1
T_6	110	0
T_7	111	1

Рис.5.3.

Таблица 5.2.

Класс	Неисправность
1	a/0, d/0, g/0
2	a/1
3	b/0
4	b/1
5	c/0, f/0, e/1, h/0
6	c/1
7	d/1
8	e/1, f/1
9	g/1, h/1, i/1
10	[/0

Таблица 5.3.

	Классы неисправностей									
	1	2	3	4	5	6	7	8	9	10
T_0						1			1	
T_1				1		1	1		1	
T_2		1							1	
T_3	1		1							1
T_4				1	1					1
T_5					1					1
T_6		1	1					1	1	
T_7	1									1

В результате моделирования этих неисправностей на входных наборах T_0-T_7 получена ТФР, приведенная в табл. 5.3. Наборы T_1, T_3, T_4 и T_6 покрывают ТФР, образуя полный тест схемы.

Функции счета. Структура устройства для компактного тестирования с использованием функций счета показана на рис.3.2.

На одновыходное тестируемое дискретное устройство подается последовательность тестовых наборов T. Сдвиговый m-разрядный регистр хранит m последних результатов $R_i=\{r_i, r_{i+1},..., r_{i+m-1},\}$ [50]. Величина $m \geq 0$ задает число последовательных результатов, подвергаемых анализу с целью определения наличия и числа представляющих интерес признаков сигналов в подпоследовательности R_i последовательности результатов R. Анализ осуществляет схема формирования результата, на выходы которой параллельно поступает последовательность результатов r_i, r_{i+1},..., r_{i+m-1}, а на выходах формируется текущее значение $C_m(R_i)$ соответствующей функции счета. Сумматор вычисляет текущее суммарное значение $S_m(R_i)$ функции счета путем арифметического сложения $C_m(R_i)$ с хранящимся в регистре накопления результатов предыдущим суммарным значением $S_m(R_{i-1})$ функции счета. Окончательное значение $S_m(R) = \sum_{i=1}^{n-m} C_m(R_i)$ функции счета сравнивается с эталонным значением $S_m(R)_{ЭТ}$ этой функции.

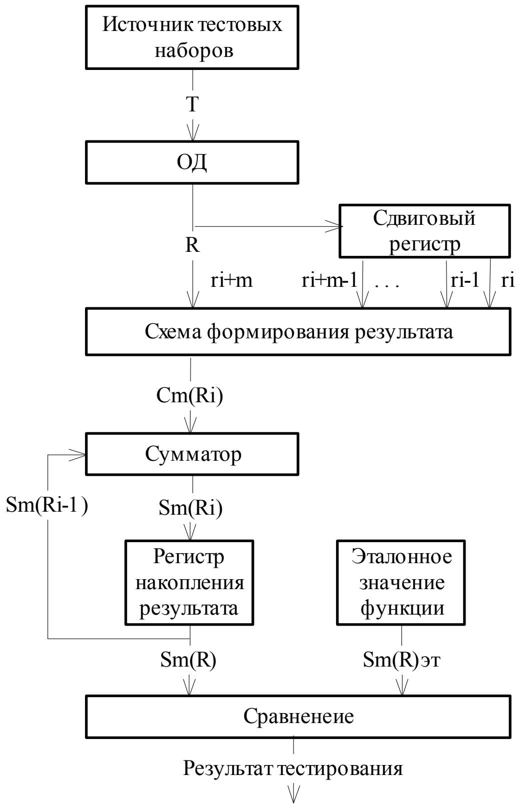


Рис.3.2.

Функция счета характеризуется глубиной памяти m (числом разрядов A) и видом признаков результатов. Наиболее просто реализуется тестирование на основе функций счета, имеющих глубину памяти 0 или 1. Такими функциями являются:

для $m=0$:

а) функция счета единичных значений результатов

$$S_0^1(R)=\sum_{i=1}^{n} r_i \;;$$

б) функция счета числа переходов изменений значений результатов из 0 в 1 и из 1 в 0

$$S_1^2(R)=\sum_{i=2}^{n}\left(r_{i-1}\oplus r_i\right);$$

для $m=1$:

в) функция счета числа повторений значений результатов

$$S_1^3(R)=\sum_{i=2}^{n}\overline{\left(r_{i-1}\oplus r_i\right)};$$

г) функция счета числа фронтов (изменений из 0 в 1)

$$S_1^4(R)=\sum_{i=2}^{n}\left(\bar r_{i-1} r_i\right);$$

д) функция счета числа срезов (изменений из 1 в 0)

$$S_1^5(R)=\sum_{i=2}^{n}\left(r_{i-1}\bar r_i\right).$$

Эффективность компактного тестирования с помощью функций счета зависит как от выбора самой функции, так и от выбора тестовой последовательности. Например, если множество тестовых наборов T разделить на два подмножества (T^0 и T^1) таких, что в первое входят все наборы, на которых выход схемы принимает нулевое значение, а во второе - все наборы, на которых выход схемы принимает единичное значение:

$$T^0=\left\{t_1^0,t_1^0,\ldots,t_{n_0}^0\right\};\quad T^1=\left\{t_1^1,t_1^1,\ldots,t_{n_1}^1\right\},$$

то при чередовании наборов $T_A=t_1^0,t_1^1,t_2^0,t_2^1\ldots$ эталонное значение функции счета числа переходов максимально:

$$S_1^2(R) = 2a$$

где $a=min\{n_0,n_1\}$. И наоборот, при подаче последовательности наборов $T_B = t_1^0, t_2^0, \ldots, t_{n_0}^0, t_1^1, t_2^1, \ldots, t_{n_1}^1$ эталонное значение функции $S_1^2(R)$ минимально и равно 1.

Для компактного тестирования многовыходных ДУ каждому получаемому на выходе ДУ двоичному набору v_i ставится в соответствие его двоичный вес a_i. В качестве функций счета применяются функции

$$S_1^8(V) = \sum_{i=2}^{m} P(a_{i-1} \langle a_i);$$

$$S_1^9(V) = \sum_{i=2}^{m} P(a_{i-1} \rangle a_i);$$

$$S_1^{10}(V) = \sum_{i=2}^{m} P(a_{i-1} \neq a_i),$$

где V – последовательность выходных наборов $v_0, v_1, \ldots, v_m$; P – предикат сравнения весов соседних наборов, принимающий значение 1, если результат сравнения совпадает с заданным предикатом, и значение 0 в противном случае.

Контрольные суммы. При использовании контрольных сумм совокупность результатов тестирования рассматривается как массив чисел, над которым выполняется операция поразрядного или арифметического суммирования [9].

Пусть задано упорядоченное множество из n m-разрядных чисел $\{u_i\}$, где $i=l,2,\ldots,n$; $u_i=u_{i1},u_{i2},\ldots,u_{im}$ соответствующее выходной последовательности ДУ. Используются следующие способы суммирования:

а) поразрядное суммирование по модулю 2:

$$K_2(u_1, u_1, \ldots, u_n) = v_1, v_2, \ldots, v_m; \quad v_i = \bigoplus_{i=1}^{n} u_{j_i};$$

б) арифметическое суммирование по различным модулям:

$$K_M(u_1, u_1, \ldots, u_n) = (u_1 + u_2 + \ldots + u_n)_{\text{mod } M},$$

причем

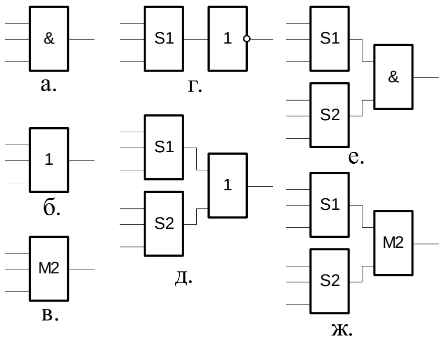
- $M = n(2^m - 1)$ - полная арифметическая сумма;
- $M = 2^m$ - арифметическая сумма без учета переноса из старшего разряда;
- $M = 2^m - 1$ - арифметическая сумма с циклическим переносом в младший разряд.

Синдром. Синдромное тестирование используется при исчерпывающем компактном тестировании.

Синдромом булевой функции называется число

$$S = K/2^n ,$$

где K – число минтермов функции; n – число входов проверяемой схемы [157].



а) $S = 2^{-n}$;

б) $S = 1 - 2^{-n}$;

в) $S = \frac{1}{2}$;

г) $S = 1 - S_1$;

д) $S = S_1 + S_2 - S_1 S_2$;

е) $S = S_1 S_2$

ж) $S = S_1 + S_2 - 2 S_1 S_2$.

Рис.3.3

Синдром используется для тестирования комбинационных схем и требует полного перебора входных наборов. Схема называется синдромно-тестируемой, если любая одиночная неисправность меняет синдром.

На рис.3.3(а, б, в) приведены синдромы простых *n*-входовых комбинационных схем. При неразветвленных входах соотношение между входными и выходными синдромами схемы, на выходе которой стоит инвертор, схема ИЛИ, схема И или схема сложения по модулю 2, показано на рис.3.3(г,д,е,ж).

✎**Пример 3.1.** Дана схема (рис.3.5). Вычислить синдром булевой функции, реализуемой этой схемой:

$$S_1 = 1 - 2^{-2} = 3/4; S_2 = 1 - 2^{-2} = 3/4; S_3 = 2^{-3} = 1/8;$$

$$S_4 = 1 - \left(S_2 + S_3 - S_2 S_3\right) = 7/32;$$

$$S = S_1 S_4 = 21/128; K = 21.$$

Тестовая процедура заключается в подаче на вход схемы всех входных наборов, определении синдрома (обычно с помощью счетчика) и сравнении его с эталонным синдромом (т. е. требуется всего один эталон).

Устройство для тестирования комбинационных схем приведено на рис.3.4.

На проверяемую схему подаются от счетчика все входные наборы. Выход проверяемой схемы соединен со входом счетчика синдрома, который подсчитывает число единиц на выходе проверяемой схемы. После перебора всех выходных наборов производится сравнение полученного и эталонного синдромов.

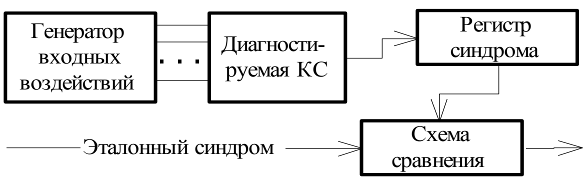


Рис.3.4.

Следует отметить, что единственным различием между синдромом и числом единиц является неявная запятая в регистре (счетчике) синдрома. Если она считается стоящей слева от числа, находящегося в регистре синдрома, то число является синдромом, если справа – то числом единиц.

Для уменьшения длины тестов, равной 2^n, комбинационная схема разбивается на подсхемы, спроектированные таким образом, что каждая из них проверяется по своему синдрому.

Для реализации синдромного тестирования комбинационные схемы должны проектироваться таким образом, чтобы синдром исправной схемы отличался от неисправной.

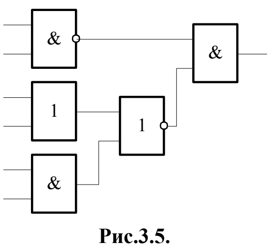


Рис.3.5.

Спектральные коэффициенты. Сжатие информации с помощью спектральных коэффициентов используется при исчерпывающем компактном тестировании комбинационных схем.

Пусть на вход схемы поступает набор $x_1,x_2,...,x_n$, $i_1,i_2,...,i_t$ – номера тех разрядов входного набора функции $F(x_1,x_2,...,x_n)$, зависимость F от которых необходимо определить [50]. *Функциями Уолша* w_i называются функции, принимающие значения ±1 и вычисляемые в соответствии с формулой:

$$w_{i_1,i_2,...,i_l}(x)=\prod_{j=1}^{l}R_{i_j}(x),$$

где $R_{i_j}(x)$ - *функция Радемахера:*

$$R_{i_j}(x)=(-1)^{x_{i_j}}.$$

Всего имеется 2^n функций Уолша. Например, для функции $F(x_1,x_2,x_3)$ имеется восемь функций Уолша: w_0, w_1, w_2, w_{12}, w_3, w_{13}, w_{23}, w_{123}.

Спектральным коэффициентом или коэффициентом Уолша называется функция

$$S(i_1,...,i_l)=\sum_{x=0}^{2^m-1}F(x)\,w_{i_1,i_2,...,i_l}(x),$$

показывающая меру зависимости значения функции от суммы по модулю 2 разрядов $x_{i_1},...,x_{i_l}$ *входного набора;* $S(i_j)$ *– зависимость от* i_j*-го разряда.*

Коэффициенты Уолша можно вычислить также по следующей формуле [50]:

$$S=T_n\cdot F \qquad (3.2)$$

где T_n есть $2^n\times2^n$ – трансформационная матрица, определяемая следующим образом:

$$T_n=\begin{vmatrix}T_{n-1} & T_{n-1}\\ T_{n-1} & -T_{n-1}\end{vmatrix},$$

а T_0=[l]. Строки матрицы T_n представляют собой значения функций Уолша.

➮**Пример 3.2.** Задана функция $f(x_1,x_2,x_3)=x_1x_3+\overline{x_1}x_2x_3$. Найти коэффициенты Уолша этой функции.

Значения функции на всех входных наборах сведены в таблицу:

x_3	x_2	x_1	f
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

Коэффициенты Уолша вычисляются по формуле (3.2) :

$$\begin{bmatrix}S_0\\S_1\\S_2\\S_3\\S_4\\S_5\\S_6\\S_7\end{bmatrix}=\begin{bmatrix}1&1&1&1&1&1&1&1\\1&-1&1&-1&1&-1&1&-1\\1&1&-1&-1&1&1&-1&-1\\1&-1&-1&1&1&-1&-1&1\\1&1&1&1&-1&-1&-1&-1\\1&-1&1&-1&-1&1&-1&1\\1&1&-1&-1&-1&-1&1&1\\1&-1&-1&1&-1&1&1&-1\end{bmatrix}\bullet\begin{bmatrix}0\\1\\0\\1\\1\\0\\0\\0\end{bmatrix}=\begin{bmatrix}3\\-1\\1\\1\\1\\-3\\-1\\-1\end{bmatrix}$$

Двоичный код индекса при коэффициенте Уолша указывает переменную, связанную с этим коэффициентом. S_0 есть не что иное, как число одиночных значений функции. Коэффициент S_1 – мера корреляции между функцией и переменной x_1. Равенство $S_1=2^{n-1}(-2^{n-1})$ свидетельствует о равенстве $f = \overline{x_1}(x_1)$. Коэффициент S_3 – мера корреляции между функцией и суммой по модулю 2 переменных $x_1 \oplus x_2$, S_7 - между f и $x_1 \oplus x_2 \oplus x_3$.

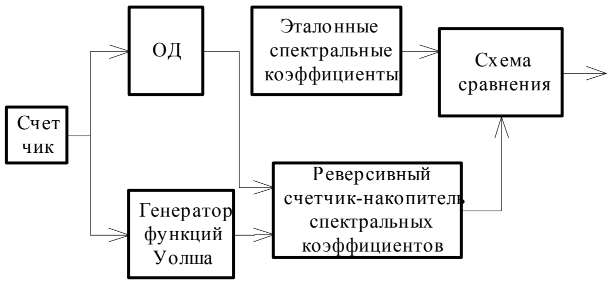
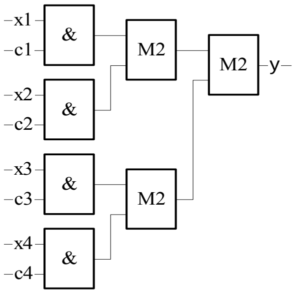


Рис.3.6.

На рис.3.6 показана схема тестирования со сжатием результатов с помощью спектральных коэффициентов. Генератор функций Уолша, показанный на этом рисунке, управляет выбором проверяемого спектрального коэффициента. Накопитель коэффициента представляет собой простой реверсивный счетчик. Схема генератора для функций от четырех переменных приведена на рис.3.7. C_i предназначено для выбора генерируемой функции Уолша.

Генератор для коэффициентов $s_0, s_1, \ldots, s_n$ тривиален. Для s_0 требуется, чтобы все константы c_i были равны нулю, а для $s_1, s_2, \ldots, s_n$ требуется простой мультиплексор.



c1	c2	c3	c4	y
1	0	0	0	x_1
0	1	0	0	x_2
0	0	1	0	x_3
0	0	0	1	x_4
1	1	0	0	$x_1 \oplus x_2$
1	0	1	0	$x_1 \oplus x_3$
...	...	...	...	
1	1	1	1	$x_1 \oplus x_2 \oplus x_3 \oplus x_4$

Рис.3.7.

Для подсчета каждого коэффициента требуется полный прогон тестовых наборов.

Распространена и другая интерпретация спектральных коэффициентов, при которой логическому 0(1) ставится в соответствие число –1(+1). Каждый спектральный коэффициент вычисляется умножением значения функции (+1 или -1) на соответствующее значение (+1 или -1) функций Уолша и суммированием произведений по двум наборам входных переменных. При этом значение каждого коэффициента лежит в пределах от —2^n до $+2^n$.

5.2.D-алгоритм

D-алгоритм является одним из первых разработанных методов синтеза детерминированных тестов для комбинационных схем, в основу которого положена идея активизации пути. Вычисление тестового набора основано на создании условий проявления неисправности и активизации пути от места ее проявления до выхода схемы.

Поскольку D-алгоритм предназначен для синтеза тестов, обнаруживающих неисправности из класса C_1 (см. раздел 4.4.2), то для определения условий проявления неисправностей достаточно анализировать описание элементов схемы.

✎Обычно предполагаемый класс неисправностей определяется как одиночные константные неисправности входов и выходов элементов (линий схемы).

Следует отметить, что для элементов, рассматриваемых отдельно, справедливы все рассуждения, приведенные в начале главы. Таким образом, когда говорят о поиске условий проявления неисправности, то предполагают, что необходимо синтезировать множество тестовых наборов, обнаруживающих ее, т.е. для каждого из элементов схемы необходимо найти тест.

В том случае, когда для элемента E: $X_e \to Y_e$ выбрана неисправность Φ, производится последовательный анализ элементов множества $\Lambda = \left\{ \langle e, \varphi \rangle \mid e \in E,\ \varphi \in \Phi,\ P_X(e) = P_X(\varphi) \right\}$, результатом которого является

$$\mathrm{T} \subseteq \Lambda \mid \forall t = \langle e, \varphi \rangle \in \mathrm{T}:\ P_Y(e) \neq P_Y(\varphi) \qquad (5.5)$$

Используемый в D-алгоритме метод активизации пути предполагает наличие двух стадий. На первой из них определяются условия активизации элементов, на второй осуществляется выбор таких условий для каждого из элементов, которые были бы непротиворечивы для всей схемы в целом.

✎Следует отметить, что состав элементного базиса, в котором синтезируются схемы, изменяется достаточно редко, следовательно, действия, выполняемые на первой стадии могут производиться по мере необходимости. Однако при реализации АСГТ это требует дополнительной информационной избыточности описаний элементов.

Условие активизации элемента E задается парой его состояний $\langle e_i, e_j \rangle$, для которой выполняется условие:

$$\langle e_i, e_j \rangle \colon \left(P_X(e_i) \neq P_X(e_j) \right) \Rightarrow \left(P_Y(e_i) \neq P_Y(e_j) \right). \tag{5.6}$$

Таким образом, определить условия активизации элемента E означает найти множество

$$\Delta \subseteq (X_e \times Y_e)^2 \,, \tag{5.7}$$

для каждого из элементов которого выполняется условие (5.6).

Отношения T и Δ определены в пространстве $(X_e \times Y_e)^2$ (см. выражения (5.5) и (5.7)), следовательно для их описания может использоваться один математический аппарат.

Таблица 5.4.

$\xrightarrow{d}$	0	1
0	0	D
1	$\overline{D}$	1

В D-алгоритме для описания элементов множеств T и Δ используется отображение, введенное Ротом [ ] и названое им d-операцией.

Пусть $S = \{\langle \tilde{s}_1, \tilde{s}_2, \ldots, \tilde{s}_{N+M} \rangle\}$ ($\tilde{s}_i \in \{0,1\}$), тогда для $\langle S^1, S^2 \rangle$ ($S^1 \in S$, $S^2 \in S$) d-операция ставит в соответствие элемент

$$D^{1,2} = \{\langle \tilde{d}_1, \tilde{d}_2, \ldots, \tilde{d}_{N+M} \rangle\} \,, \tag{5.8}$$

в котором $\langle \tilde{s}_i^1, \tilde{s}_i^2 \rangle \xrightarrow{d} \tilde{d}_i$ ($i = \overline{1, N+M}$), $\tilde{d}_i \in \{0,1,D,\overline{D}\}$, а поразрядная операция $\xrightarrow{d}$ задана таблично (см. табл.5.4).

Таким образом, описание элемента состоит из:

- множества кубов неисправности – множества T, заданного в виде (5.8);
- множества кубов распространения (D-кубов) – множества Δ, элементы которого представляют собой описание элемента в алфавите $\{0,1,D,\overline{D}\}$;
- множества кубов вырожденного покрытия – функционального описания элемента в алфавите $\{0,1,x\}$.

✎Примеры кубов распространения, вырожденного покрытия и неисправностей для некоторых элементов функционально-логического уровня приведены в таблицах 5.5, 5.6, 5.7.

Таблица 5.5.

$c = a\&b$			$c = \overline{a\&b}$			$c = a \vee b$			$c = \overline{a \vee b}$		
a	b	c	a	b	c	a	b	c	a	b	c
D	1	D	D	1	$\overline{D}$	D	0	D	D	0	$\overline{D}$
1	D	D	1	D	$\overline{D}$	0	D	D	0	D	$\overline{D}$
$\overline{D}$	1	$\overline{D}$	$\overline{D}$	1	D	$\overline{D}$	0	$\overline{D}$	$\overline{D}$	0	D
1	$\overline{D}$	$\overline{D}$	1	$\overline{D}$	D	0	$\overline{D}$	$\overline{D}$	0	$\overline{D}$	D

Таблица 5.6.

$c=a\&b$			$c=\overline{a\&b}$			$c=a\vee b$			$c=\overline{a\vee b}$		
a	b	c	a	b	c	a	b	c	a	b	c
x	0	0	x	0	1	0	0	0	0	0	1
0	x	0	0	x	1	1	x	1	1	x	0
1	1	1	1	1	0	x	1	1	x	1	0

Таблица 5.7.

	$c=a\&b$	$c=\overline{a\&b}$	$c=a\vee b$	$c=\overline{a\vee b}$
a=0	11 $\overline{D}$	11 D	10 $\overline{D}$	10 D
a=1	01 D	01 $\overline{D}$	00 D	00 $\overline{D}$
b=0	11 $\overline{D}$	11 D	01 $\overline{D}$	01 D
b=1	10 D	10 $\overline{D}$	00 D	00 $\overline{D}$
c=0	11 $\overline{D}$	11 D	01 $\overline{D}$ 10 $\overline{D}$ 11 $\overline{D}$	00 $\overline{D}$
c=1	00 D 01 D 10 D	11 $\overline{D}$	00 D	01 D 10 D 11 D

Если такое описание известно для всех элементов схемы, то действия, выполняемые на второй стадии активизации пути сводятся к решению переборной задачи на множествах кубов неисправности, кубов продвижения и кубов вырожденного покрытия. При этом, определяя условия непротиворечивости выбранных кубов, исходят из того, что состояние каждой линии схемы должно однозначно определять состояние множества входов-выходов, ассоциированных с ней.

Для решения описанной переборной задачи используется алгоритм последовательного перебора с возвратами, на каждом шаге которого уточняется статус схемы. Первоначально полагают, что состояния линий неизвестны, затем выбирают куб неисправности, после чего осуществляют выбор кубов продвижения для элементов. В этом случае алфавит состояний для линии определяется множеством $\{0,1,D,\overline{D},x\}$.

Условие непротиворечивости выбора очередного куба задается поразрядной операцией *d-пересечения* статусов S^i и S^{i+1}, полученных на $i-$ом и $i+1-$ом шаге.

Пусть $\alpha=\{\alpha_1,\alpha_2,...,\alpha_n\},\beta=\{\beta_1,\beta_2,...,\beta_n\},\alpha_i,\beta_i\in\{0,1,x,d,\overline{d}\}$, тогда операция d-пересечения задается правилами:

$$x\Pi\alpha_i=\alpha_i.$$

$$\alpha_i\Pi\beta_i=\begin{cases}\alpha_i,\text{ если }\alpha_i=\beta_i;\\ \varnothing,\text{ если }\alpha_i\neq\beta_i.\end{cases}$$

$$\alpha\Pi\beta=\begin{cases}\langle\alpha_1\Pi\beta_1,\alpha_2\Pi\beta_2,...,\alpha_n\Pi\beta_n\rangle,\text{ если }\forall i{:}\alpha_i\Pi\beta_i\neq\varnothing;\\ \qquad\varnothing,\text{если }\exists i{:}\alpha_i\Pi\beta_i=\varnothing.\end{cases}$$

Последовательность шагов алгоритма можно условно разделить на три этапа. На первом этапе осуществляется d-пересечение с кубами неисправностей. На втором этапе, называемом *d-распространением,* происходит активизация пути от выхода элемента, для которого статус определен одним из кубов неисправности, до внешних выходов схемы. d-распространение состоит в последовательном выполнении шагов, на каждом из которых необходимо найти статусы выходов элементов d-границы.

✎d-граница представляет собой множество элементов, на некоторых входах которых зафиксированы символы D или $\overline{D}$, а состояние выхода неопределено.

На этом этапе осуществляется пересечение с кубами продвижения элементов d-границы.

Третий этап, который называют доопределением, осуществляется только в том случае, когда в схеме после выполнения d-распространения остались линии с неопределенным состоянием. В этом случае производится выбор и пересечение с кубами вырожденного покрытия.

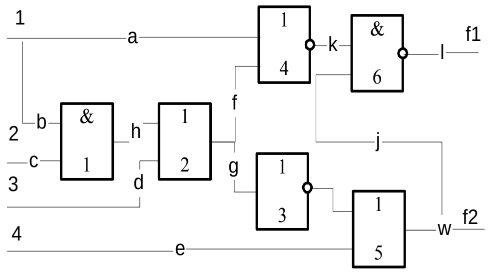


Рис. 5.4.

В таблице 5.6 приведена последовательность шагов вычисления тестового набора для схемы, изображенной на рис.5.4, и неисправности $h_{/0}$.

Исходным состоянием схемы является неопределенное состояние всех линий, обозначаемое символом x . (Для удобства чтения в таблице 5.6 символ x не отображается.) В начале таблицы приведены d-кубы распространения для всех элементов схемы. На первом и втором шаге символ D проталкивается через элементы 2 и 3. Для этого выполняется пересечение с D-кубом распространения.

Таблица 5.6.

Сигналы на линиях схемы													**Номер вентиля**	**Примечание**
a	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**	**l**	**w**		
	1	d					d						1	**D-кубы распространения для вентилей схемы**
	d	1					d							
	1	$\bar{d}$					$\bar{d}$							
	$\bar{d}$	1					$\bar{d}$							
			0		d	d	d						2	
			d		d	d	0							
			0		$\bar{d}$	$\bar{d}$	$\bar{d}$							
			$\bar{d}$		$\bar{d}$	$\bar{d}$	0							
					d	d				$\bar{d}$			3	
					$\bar{d}$	$\bar{d}$				d				
0					d	d				$\bar{d}$			4	
d					0	0				$\bar{d}$				
0					$\bar{d}$	$\bar{d}$				d				
$\bar{d}$					0	0				d				
				0				d	d			d	5	
				d				0	d			d		
				0				$\bar{d}$	$\bar{d}$			$\bar{d}$		
				$\bar{d}$				0	$\bar{d}$			$\bar{d}$		
									1	d	$\bar{d}$		6	
									d	1	$\bar{d}$			
									1	$\bar{d}$	d			
									$\bar{d}$	1	d			
1	1	1					d						—	**Тест-куб неисправности**
1	1	1	0		d	d	d						1	**Проталкивание D через элемент 2**
1	1	1	0		d	d	d	$\bar{d}$					2	**Проталкивание D через элемент 3**
∅	1	1	0		d	d	d	$\bar{d}$		$\bar{d}$			3	**Попытка проталкивания D через элемент 4. Результат — ∅**
1	1	1	0	0	d	d	d	$\bar{d}$	$\bar{d}$			$\bar{d}$	4	**Проталкивание D через элемент 5. Тест вычислен**

На третьем шаге делается попытка проталкивания через элемент 4. Однако пересечение с D-кубом распространения элемент 4 дает $\varnothing$, т.е. пустой куб. Поэтому делается попытка распространения D на выход через элемент 5, которая завершается успешно. Так как значения на всех линиях схем имеют определенное значение, то этап доопределения отсутствует. Таким образом, тестовым для неисправности $h_{/0}$ является набор 1100.

5.3. Метод булевой производной

Метод булевой производной рассчитан на синтез тестов для одиночных константных неисправностей и использует аналитическую форму функционального описания ОД.

Булевой производной функции $f(x_1,x_2,...,x_n)$ по x_i называется функция:

$$^{df(x)}\!/_{dx_i} = f(x_1,x_2,...,x_{i-1},0,x_{i+1},...,x_n) \oplus f(x_1,x_2,...,x_{i-1},1,x_{i+1},...,x_n).$$

Булева производная принимает единичное значение на тех наборах значений логических переменных $x_1,x_2,...,x_n$ (кроме x_i), при которых изменение состояния x_i приводит к изменению значения функции $f(x_1,x_2,...,x_n)$. Тестовым набором для неисправности $x_i/0$ *(*$x_i/1$*)* являются наборы значения логических переменных, при которых функция $x_i \cdot {}^{df(x)}\!/_{dx_i}$ $(\bar{x}_i \cdot {}^{df(x)}\!/_{dx_i})$ принимает единичное значение.

Ниже приведены примеры, поясняющие вычисление тестовых наборов методом булевой производной.

✎ **Пример 5.1.**

Дана схема, реализующая функцию $f(x) = x_1x_2 \vee x_3$. Найти тесты неисправностей $x_1/0$ и $x_1/1$.

$$^{df(X)}\!/_{dx_1} = (1 \cdot x_2 \vee x_3) \oplus (0 \cdot x_2 \vee x_3) = (x_2 \vee x_3) \oplus x_3 = x_2\bar{x}_3.$$

Тестовый набор для x1/0 определяется исходя из условия $x_i \cdot {}^{df(x)}\!/_{dx} = 1$, т.е. $x_1x_2\bar{x}_3 = 1$. Следовательно, входной набор 110 обнаруживает неисправность $x_1/0$. Аналогичным образом, для неисправности $x_1/1$ необходимо, чтобы $\bar{x}_i \cdot {}^{df(x)}\!/_{dx} = 1$, следовательно входной набор 010 обнаруживает неисправность $x_1/1$.

✎ **Пример 5.2.**

Дана схема (рис.5.5), реализующая функцию $f(X) = x_1x_2 \vee x_1\bar{x}_2$ Найти тесты неисправностей $x_2/0$ и $x_2/1$.

$$^{df(X)}\!/_{dx_1} = (1 \cdot x_1 \vee 0 \cdot x_1) \oplus (0 \cdot x_1 \vee 1 \cdot x_1) = 0.$$

Это означает, что *f(X)* не зависит от *x2*.

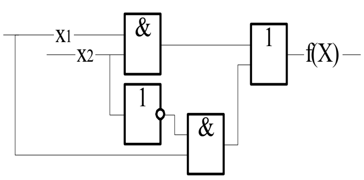


Рис.5.5.

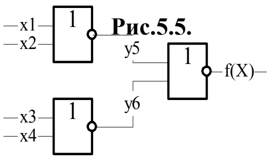


Рис.5.6.

✎ **Пример 5.3.**

Дана схема , изображенная на рис.5.6. Найти тест неисправности *y6/0*.

Сначала необходимо выразить *f(x)* через внутренние переменные схемы:

$$f(X) = \overline{y_5 \vee y_6}.$$

Затем найти булеву производную :

$$^{df(X)}\!/_{dy_6} = \overline{(y_5 \vee 0)} \oplus \overline{(y_5 \vee 1)} = \bar{y}_5 = x_1 \vee x_2.$$

Тестовый набор для неисправности *y6/0* находится исходя из условия $y_6 \cdot {}^{df(X)}\!/_{dy_6} = 1$.

Нетрудно убедиться в том, что тестовые наборы 1x00 и x100 обнаруживают неисправность *y6/0.*

5.4. Метод эквивалентных нормальных форм

Этот метод основан на представлении булевой функции и виде эквивалентной нормальной формы (ЭНФ), описывающей конкретную реализацию схемы. Поскольку ЭНФ представляет собой сумму логических произведений, она соответствует гипотетической схеме нескольких И— ИЛИ. Каждой схеме И соответствует один терм ЭНФ. Из такого представления ЭНФ становится очевидным, что для выявления неисправностей, связанных с переменной x_i, входящей в какой-либо терм ЭНФ, необходимо выполнение следующих условий:

1. равенство нулю всех термов, кроме содержащего переменную x_i*;*
2. равенство единице всех переменных терма, в который входит тестируемая переменная x_i.

Выполнение этих условий обеспечивает тождественное равенство $f(X) \equiv x_i$ и, как следствие этого, выявление неисправностей, связанных с этой переменной, так как неисправность переменной приведет к изменению сигнала на выходе схемы.

Эквивалентная нормальная форма, как и обычная нормальная, вычисляется методом подстановки, с той лишь разницей, что избыточные термы не исключаются, так как они характеризуют конкретную реализацию схемы.

При построении тестов важно не только обеспечить проверку входных переменных, но и всех путей, т.е. необходимо обеспечить проверку одной и той же переменной в разных термах, которым соответствуют разные пути в схеме.

✎ **Пример 5.4.**

Дана схема (см. рис.5.4). Найти тесты неисправностей $a_{/0}, a_{/1}, b_{/0}, b_{/1}$ методом ЭНФ.

Для представления функции f_1 и f_2 в виде ЭНФ необходимо выписать значения всех промежуточных переменных:

$$f_1 = l = \bar{k} \vee \bar{j}, \quad k = \bar{a} \cdot \bar{f}, \quad \bar{k} = a \vee f, \quad j = i \vee e,$$
$$\bar{j} = \bar{i} \cdot \bar{e}, \ i = \bar{g}, \bar{i} = g, \ g = f = h \vee d, \ \bar{g} = \bar{f} = \bar{h} \cdot \bar{d}, \ h = bc,$$
$$h = \bar{b} \vee \bar{c}, \ f_2 = w = j = i \vee e.$$

После выполнения всех подстановок, которые выполняются с выносом замещаемых переменных в квадратные скобки и сохранением их в выражении, ЭНФ для функций f_1 и f_2 принимает вид :

$$f_1 = [\bar{k}](a \vee f) \vee [\bar{j}]\bar{i} \cdot \bar{j} = [\bar{k}]a \vee [\bar{k} f](h \vee d) \vee [\bar{i} \cdot \bar{j}]g\bar{e} =$$
$$= [\bar{k}]a \vee [\bar{k} f]h \vee [\bar{k} f]d \vee [\bar{j} \cdot \bar{i}g]h\bar{e} \vee [\bar{j} \cdot \bar{i}g]d\bar{e} =$$
$$= [\bar{k}]a \vee [\bar{k} fh]bc \vee [\bar{k} f]d \vee [\bar{j} \cdot \bar{i}gh]bc\bar{e} \vee [\bar{j} \cdot \bar{i}g]d\bar{e};$$

$$f_2 = i \vee e = [i]\bar{g} \vee e = [i\bar{g}]\bar{h} \cdot \bar{d} \vee e = [i\bar{g} \cdot \bar{h}](\bar{b} \vee \bar{c})\bar{d} \vee e =$$
$$= [i\bar{g} \cdot \bar{h}]\bar{b} \cdot \bar{d} \vee [i\bar{g} \cdot \bar{h}]\bar{c} \cdot \bar{d} \vee e.$$

Как видно из выражений ЭНФ булевых функций f_1 и f_2 полностью соответствуют конкретной реализации схемы, а каждому терму соответствует путь распространения сигнала от входа схемы к ее выходу. Из выражения для f_1 следует, что для определения тестов неисправностей $a_{/0}$ и $a_{/1}$ необходимо обеспечить равенство нулю всех термов, кроме $[k]a$, т.е. c=0, d=0.Таким образом, тестовым для неисправности $a_{/0}$ будет набор a=1, c=0, d=0, e=x, а для неисправности $a_{/1}$ – набор a=0, c=0, d=0, e=x.

Теста неисправности $b_{/0}$ по пути *kfh* или *jigh* не существует, так как для него требуется b=1, а следовательно, и a=1, в результате чего не все термы (кроме содержащего тестируемую переменную) устанавливаются в 0. Тест неисправности $b_{/0}$ может быть найден из выражения для f_2 по пути *igh*. Этим тестом будет набор b=1, d=1, c=0, e=0, т.е. b=1, d=0, c=1, e=0. Тестом неисправности $b_{/1}$ по этому же пути будет набор b=0, d=0, c=1, e=0.

5.5. Синтез детерминированных тестов для последовательностных схем

Рассмотренные выше методы синтеза детерминированных тестов предназначены для комбинационных устройств. Если ОД имеет внутренний алфавит, то в этом случае задача синтеза тестов решается гораздо сложнее. Существует несколько подходов к ее решению.

Первый из них связан с попытками разработки для последовательностных устройств аналогичных описанным в предыдущих разделах методов. Однако такой подход оказался мало перспективным и представляет в настоящее время чисто ретроспективный интерес. Среди причин, обусловивших такое положение, весомое значение имеют следующие. Во-первых, рост объема вычислений при синтезе теста связан с тем, что приходится анализировать формирование выходных сигналов, по которым анализируется поведение ОД, в каждом из возможных состояний. Во-вторых, прежде, чем использовать тест, необходимо установить ОД в известное, чаще всего, начальное состояние, т.е. использовать специальную *установочную последовательность* входных воздействий.

✎Для последовательностных схем известен метод синтеза установочных последовательностей, предложенный Хенни [134]. В этом методе ОД представляется автоматной моделью, описанной в табличном виде. Следовательно, неисправности, обнаруживаемые этими тестами, относятся к классу A_0 . Вместе с тем верхняя граница длины тестовой

последовательности для ОД, имеющего n состояний, составляет величину $n!$, что абсолютно неприемлемо с практической точки зрения.
Известны также исследования Поджа и Маккласки [151] в этой области, которые не выходят за рамки общих положений, приведенных в начале главы и имеют все те же недостатки, что и метод Хенни. Следует отметить, что оба метода неприменимы, если описание ОД не представлено автоматной моделью.

Второй подход связан с моделированием ОД. Наряду с достоинством, которое заключается в том, что методы этой группы позволяют синтезировать тесты для довольно больших схем, имеется ряд недостатков, связанных с тем, что методы этой группы не гарантируют получения оптимальных, и даже полных, тестовых последовательностей.

Таким образом, в настоящее время по-прежнему актуальны исследования, связанные с разработкой методов тестирования рассматриваемых ОД. Эти исследования развиваются в двух взаимосвязанных направлениях. Первое из них связано с разработкой методов, ориентированных на уровни описание ОД, близкие к системному. (Подробно эти методы рассмотрены в следующей главе.) Второе – с разработкой специальной методологии тестопригодного проектирования устройств.

↳Простейшими примерами такого проектирования последовательностных схем является использование цепей сброса в начальное состояние, которое позволяет свести к минимуму длину установочной последовательности, а также разрыв (физический или логический) цепей обратных связей, который позволяет в режиме тестирования перевести ОД в класс комбинационных устройств, методы синтеза тестов для которых хорошо изучены.

Оба направления настолько взаимосвязаны, что методы синтеза тестов требуют применения специальной методологии проектирования, а методы тестопригодного проектирования, в свою очередь, обусловливают не только сложность тестирования, но и возможности применения в целом тех или иных методов тестирования. Этим проблемам посвящена заключительная глава книги.