

В [4-3, 4-4] рассматривается метод построения контролирующего автомата с использованием разбиений со свойством подстановки на множестве состояний исходного автомата.

В [4-7] рассматривается задача построения контролирующего автомата (в терминах таблицы переходов-выходов) в случае, когда в исходном автомате возможны неисправности из некоторого произвольного наперед заданного множества. Формальный метод построения контролирующего автомата в случае, когда возможные ошибки исходного автомата могут быть описаны в терминах бинарных отношений на множестве его выходных сигналов, предложен в [4-8].

В [4-9] приводится процедура построения контролирующего автомата для автомата с конечной памятью.

В [4-11] показано, что при контроле автомата, в котором возможны произвольные кратные неисправности, число состояний контролирующего автомата не может быть меньше числа состояний контролируемого автомата.

ГЛАВА ПЯТАЯ

МЕТОДЫ ПОСТРОЕНИЯ САМОПРОВЕРЯЕМЫХ СХЕМ ВСТРОЕННОГО КОНТРОЛЯ ДЛЯ ДИСКРЕТНЫХ УСТРОЙСТВ

5.1. ПОСТРОЕНИЕ САМОПРОВЕРЯЕМЫХ СХЕМ ВСТРОЕННОГО КОНТРОЛЯ ДЛЯ КОМБИНАЦИОННЫХ УСТРОЙСТВ

В § 4-1 была рассмотрена процедура получения задания на синтез несамопроверяемых СВК для комбинационных устройств. Получаемые согласно этой процедуре СВК обеспечивают обнаружение всех одиночных логических неисправностей комбинационного устройства. В этом параграфе рассматривается процедура получения задания на синтез самопроверяемых СВК. При этом самопроверяемые СВК также должны обнаруживать все одиночные логические неисправности комбинационного устройства, а также все одиночные константные неисправности входов и выходов элементов СВК.

ПОЛУЧЕНИЕ ЗАДАНИЯ НА СИНТЕЗ

Рассмотрим сначала случай, когда СВК имеет два выхода f_1 и f_2 (табл. 5-1). О назначении выходных функций f'_1, f'_2 и f'_3 в табл. 5-1 будет сказано позднее. В подтаблице правильных значений (в таблице истинности исправной схемы) значения функций f_1 и f_2 должны быть противоположными, причем каждая из функций f_1 и f_2 должна иметь в этой подтаблице как значения, равные единице, так и значения, равные нулю. Это значит, что в подтаблице правильных значений пары значений функций f_1, f_2 будут вида 10 и 01. В подтаблицах неправильных значений (в таблицах истинности неисправных схем) значения функций f_1 и f_2 должны быть отличными от значений 01 и 10, т. е. одинаковыми — 00 или 11.

Таблица 5-1

x_1	...	x_n	z_1	...	z_k	f_1	f_2	f'_1	f'_2	f'_3	Схема
α_{11}	...	α_{1n}	β_{11}	...	β_{1k}	1	0	γ^1_1	γ^1_2	γ^1_3	Исправная
.		
.		
α_{r1}	...	α_{rn}	β_{r1}	...	β_{rk}	0	1	γ^r_1	γ^r_2	γ^r_3	С инвертором на выходе первой подсхемы
α_{11}	...	α_{1n}	β^1_{11}	...	β^1_{1k}	0	0	0	0	0	
.		
α_{r1}	...	α_{rn}	β^1_{r1}	...	β^1_{rk}	1	1	0	0	0	
.		
.		
α_{11}	...	α_{1n}	β^m_{11}	...	β^m_{1k}	0	0	0	0	0	С инвертором на выходе m -й подсхемы
.		
.		
α_{r1}	...	α_{rn}	β^m_{r1}	...	β^m_{rk}	1	1	0	0	0	
.		
.		

Дадим формальное определение самопроверяемой СВК. Схема встроенного контроля с двумя выходами $/1, f_2$ является полностью самопроверяемой, если она обладает следующими двумя свойствами: а) самотестируемости — все неисправности СВК из заданного класса проявляются на выходах f_1, f_2 в виде пар значений 00 или 11 хотя бы на одном входном наборе из множества $\{X\}_p$ при исправном КУ; б) защищенности от неисправностей — каждая неисправность СВК из заданного класса проявляется на выходах f_1, f_2 только в виде пары значений 00 или 11. Поскольку в дальнейшем частично самопроверяемые СВК не рассматриваются, то из словосочетания «полностью самопроверяемые СВК» исключено слово «полностью».

При произвольной реализации СВК, имеющая два выхода f_1 и f_2 , будет гарантированно обнаруживать неисправности только контролируемого комбинационного устройства, нарушающие его правильное функционирование. Поэтому одним из условий построения самопроверяемой СВК является требование отдельной реализации функций f_1 и f_2 . Обоснование этого требования состоит в следующем. Неисправность СРК будет обнаружена лишь при изменении значений выходов f_1 и f_2 с 01 или 10 на 00 или 11, т. е. при инвертировании значения только одного выхода. Отдельная реализация функций f_1 и f_2 гарантирует, что любая одиночная константная неисправность на входах и выходах элементов СВК не приведет к инвертированию обоих выходов схемы. Таким образом, при отдельной реализации функций f_1 и f_2 заведомо выполняется свойство защищенности от неисправностей. Поэтому в дальнейшем исследуются только условия обеспечения самотестируемости СВК.

Далее, СВК самопроверяема, если все одиночные константные неисправности схем, реализующих функции f_1 и f_2 , обнаруживались на множестве наборов $\{\alpha_{j1}, \dots, \alpha_{jn}, \beta_{j1}, \dots, \beta_{jk}\}$, $j=1, \dots, g$. В этом случае пары значений сигналов на выходах f_1, f_2 вида 01, 10 будут свидетельствовать о правильном функционировании комбинационного устройства и об отсутствии одиночных неисправностей в СВК. Пары значений сигналов на выходах f_1, f_2 вида 00, 11 будут свидетельствовать о наличии одиночной неисправности в блоке, содержащем комбинационное устройство и СВК.

В настоящем параграфе рассмотрены условия, при выполнении которых возможно построение самопроверяемых СВК по общей таблице истинности (табл. 5-1) для комбинационных устройств. В § 5-3 и 5-4 описаны регулярные методы, обеспечивающие построение самопроверяемых СВК специального вида (схем так называемой типовой реализации) для комбинационных устройств и автоматов с памятью. В § 5-2 рассмотрены методы построения самопроверяемых СВК для кодов m из n .

Как уже отмечалось, при кодировании общей таблицы истинности значениями функций f_1 и f_2 в подтаблице правильных значений должны быть пары как вида 10, так и вида 01 и в подтаблицах неправильных значений — 00 или 11 (в последнем случае не обязательно должны быть обе пары значений). Однако в зависимости от порядка приписывания той или иной строке табл. 5-1 пары значений функций f_1 и f_2 существенно может изменяться как сложность схемы контроля, так и возможность ее построения в виде самопроверяемой СВК. Будем считать, что строки таблицы произвольно заполняются парами значений функций f_1 и f_2 с обязательным соблюдением отмеченного выше необходимого условия.

После заполнения общей таблицы истинности значениями функций f_1 и f_2 необходимо определить подмножества существенных входных переменных из множества $x_1, \dots, x_n, z_1, \dots, z_k$ для каждой функции f_1 и f_2 . Заметим, что функции f_1 и f_2 , как правило, являются не полностью определенными.

Для получения одной из возможных совокупностей существенных переменных функций $f_1(f_2)$ воспользуемся алгоритмом А5-1. Операции этого алгоритма следующие:

1. Найти в общей таблице истинности среди переменных $x_1, \dots, x_n, z_1, \dots, z_k$ такие переменные, для которых существует хотя бы одна пара наборов, соседних по этой переменной и закодированных разными значениями функции $f_1(f_2)$; отметить найденные переменные. Проверить наличие в таблице неотмеченных переменных: а) если неотмеченные переменные есть, то перейти к п. 2; б) если неотмеченных переменных нет, то перейти к п. 3.

2. Вычеркнуть (не восстанавливая предыдущий вы-

черкнутый) в таблице столбец, соответствующий любой одной неотмеченной переменной. Снова проверить наличие в таблице неотмеченных переменных: а) если неотмеченные переменные есть, то вернуться к п. 1; б) если неотмеченных переменных нет, то перейти к п. 3.

3. Конец. Отмеченные (невывернутые) переменные являются существенными для функции $f_1(f_2)$.

Таким образом, в зависимости от выбранного кодирования общей таблицы истинности парами значений функций f_1 и f_2 , а также от порядка вычеркивания неотмеченных переменных и соответствующих им столбцов, будут получены те или иные совокупности существенных переменных для этих функций.

Пример 5-1. Рассмотрим процедуру нахождения существенных переменных при построении самопроверяемой СВК для одноразрядного сумматора, схема i -го разряда которого изображена на рис. 5-1,а. На рис. 5-1,а: a_i, b_i — i -е разряды слагаемых; c_{i-1} — перенос из $(i-1)$ -го разряда сумматора; s_i — i -й разряд суммы; c_i — перенос в $(i+1)$ -й разряд. На рис. 5-1,б приведен обобщенный граф схемы сумматора. Все множество элементов разбивается на два подмножества: $\{1, 2, 3, 4, 5, 6\}$, $\{7, 8, 9, 10\}$. Определим значения, реализуемые исправным сумматором на выходах s_i и c_i и при замене функций элементов 6 и 10 на инверсные функции. В табл. 5-2 приведена общая таблица истинности, закодированная значениями функций f_1 и f_2 . Заметим, что инвертор на выходе 10 инвертирует также все значения функции s_i .

Для функции f_1 согласно п. 1 алгоритма А5-1 будет отмечена только одна переменная a_i . Двумя наборами входных переменных, соседних по a_i и закодированных различными значениями функции f_1 , являются наборы 4 и 16. После последовательного вычеркивания по одной из оставшихся переменных b_i, c_{i-1}, s_i, c_i (последовательного применения пп. 2 и 1 алгоритма А5-1) для каждой из этих переменных не найдется ни одной пары наборов, обладающих указанными свойствами. В результате из общей таблицы истинности будут вычеркнуты все переменные, кроме a_i . Это значит, что для функции f_1 существенной является только одна входная переменная a_i .

Для функции f_2 существенными являются переменные b_i (пара соседних закодированных по-разному наборов 1 и 11), c_{i-1} (наборы 1 и 10) и s_i (наборы 1 и 9).

Самопроверяемая СВК, реализуемая по тупиковой дизъюнктивной нормальной форме в скобочной форме

После выделения существенных переменных можно приступить к реализации функций f_1 и f_2 как функций, зависящих только от существенных переменных. Для этого могут быть привлечены те или иные методы теории релейных устройств и конечных автоматов. Однако

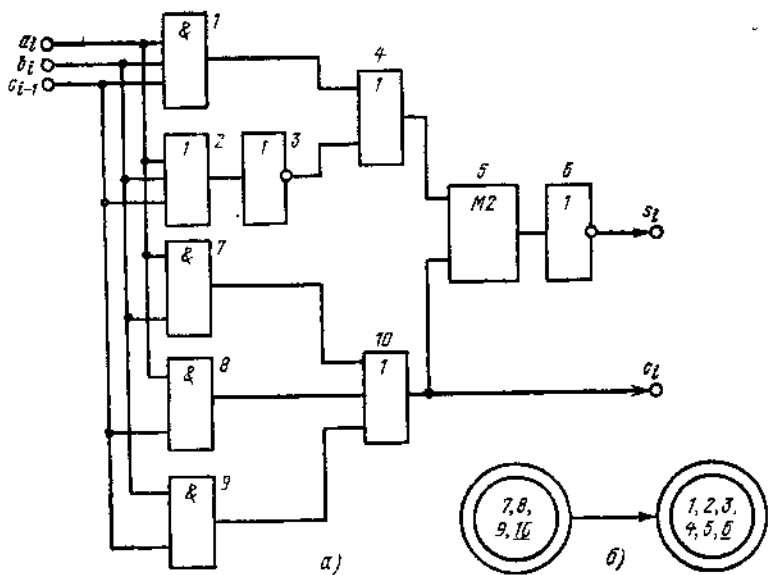


Рис. 5-1. Схема одноразрядного сумматора (а), обобщенный граф (б).

нас будут интересовать реализации некоторых специальных видов, обеспечивающие условия самопроверяемости СВК. К числу таких реализаций относятся реализация тупиковой дизъюнктивной нормальной формы (ДНФ) булевой функции в скобочной форме¹, а также реализация в виде древовидной структуры без разветвлений. Схема является реализацией тупиковой ДНФ в скобочной форме, если получаемое по схеме методом прямой или обратной подстановки [1-5, гл. 3] аналитическое выражение совпадает со скобочной формой, которую можно получить из тупиковой ДНФ путем вынесения некоторых переменных последней за скобки. Древовидная структура без разветвлений не имеет разветвлений внешних "входов, а также выходов элементов.

Опишем условия, при которых указанные реализации функций f_1 и f_2 будут самопроверяемыми.

¹ Тупиковая ДНФ булевой функции представляет собой избыточную дизъюнкцию избыточных конъюнкций (простых импликантов).

Таблица 5-2

№ п/п	a_i	b_i	c_{i-1}	s_i	c_i	f_1	f_2	Схема
1	0	0	0	0	0	1	0	Исправная
2	0	0	1	1	0	1	0	
3	0	1	0	1	0	1	0	
4	0	1	1	0	1	1	0	
5	1	0	0	1	0	0	1	
6	1	0	1	0	1	0	1	
7	1	1	0	0	1	0	1	
8	1	1	1	1	1	0	1	
9	0	0	0	1	0	1	1	С инвертором на выходе элемента НЕ.
10	0	0	1	0	0	1	1	
11	0	1	0	0	0	1	1	
12	0	1	1	1	1	1	1	
13	1	0	0	0	0	0	0	
14	1	0	1	1	1	0	0	
15	1	1	0	1	1	0	0	
16	1	1	1	0	1	0	0	
17	0	0	0	1	1	1	1	С инвертором на выходе элемента ИЛИ.
18	0	0	1	0	1	1	1	
19	0	1	0	0	1	1	1	
20	0	1	1	1	0	1	1	
21	1	0	0	0	1	0	0	
22	1	0	1	1	0	0	0	
23	1	1	0	1	0	0	0	
24	1	1	1	0	0	0	0	

Тупиковая ДНФ, на основе которой получена некоторая скобочная реализация функции, представляет собой для этой скобочной реализации функции эквивалентную форму (ЭФ), в которой удалены индексы расщепленных переменных. С другой стороны, для каждой буквы тупиковой ДНФ булевой функции всегда может быть построен проверяющий эту букву входной набор. Следовательно, совокупность входных наборов, проверяющих все буквы тупиковой формы, является проверяющим тестом относительно одиночных константных неисправностей схемы, представляющей скобочную реализацию тупиковой ДНФ.

Отсюда следует, что СВК с двумя выходами f_1 и f_2 будет самопроверяемой, если для каждой из функций f_1 и f_2 , имеющих отдельные реализации их тупиковых ДНФ в скобочной форме, среди наборов $\{\alpha_{j1} \dots, \alpha_{jn}, \beta_{j1}, \dots, \beta_{jk}\}$, $j=1, \dots, r$, существуют совокупности наборов, проверяющих каждую букву простых импликантов тупиковых форм функций f_1 и f_2 .

Пример 5-2. Построим тупиковую ДНФ, а также соответствующую скобочную форму для СВК одноразрядного сумматора, схема i -го разряда которого дана на рис 5-1, а, и проанализируем, является ли полученная схема самопроверяемой

Общей таблицей истинности для функций f_1 и f_2 является табл. 5-2. В примере 5-1 было найдено, что существенной переменной для функции f_1 является переменная a_i , а функция f_2 имеет три существенные переменные b_i , c_{i-1} и s_i .

Воспользовавшись каким-либо методом минимизации булевых функций, находим тупиковые формы функций f_1 и f_2 :

$$f_1 = \bar{a}_i, \\ f_2 = \bar{b}_i \bar{c}_{i-1} s_i \vee \bar{b}_i c_{i-1} \bar{s}_i \vee b_i \bar{c}_{i-1} \bar{s}_i \vee b_i c_{i-1} s_i.$$

Для f_1 скобочное представление тупиковой ДНФ совпадает с последней, а для f_2 один из возможных вариантов скобочного представления:

$$f_2 = s_i (\bar{b}_i \bar{c}_{i-1} \vee b_i c_{i-1}) \vee \bar{s}_i (\bar{b}_i c_{i-1} \vee b_i \bar{c}_{i-1}).$$

На рис. 5-2 показана соответствующая реализация тупиковых ДНФ в скобочной форме для функций f_1 и f_2 , а также СВК вместе с подключением схемы ко входам и выходам сумматора, представленного прямоугольником, отмеченным символом S.

По схеме рис. 5-2 методом прямой или обратной подстановки можно получить эквивалентную форму (ЭФ) схемы, которая совпадает (с точностью до индексов «расщепляемых» переменных) с тупиковыми формами функций f_1 и f_2 .

Методом [1-5, гл. 3] по тупиковым формам функций f_1 и f_2 находим для них проверяющие тесты (совокупности входных наборов, проверяющих каждую букву тупиковых форм на константы 0 и 1). Для f_1 тест содержит два набора: 0,1 (a_i), а для f_2 — восемь наборов: 000, 001, 010, 011, 100, 101, 110, 111 (b_i, c_{i-1}, s_i).

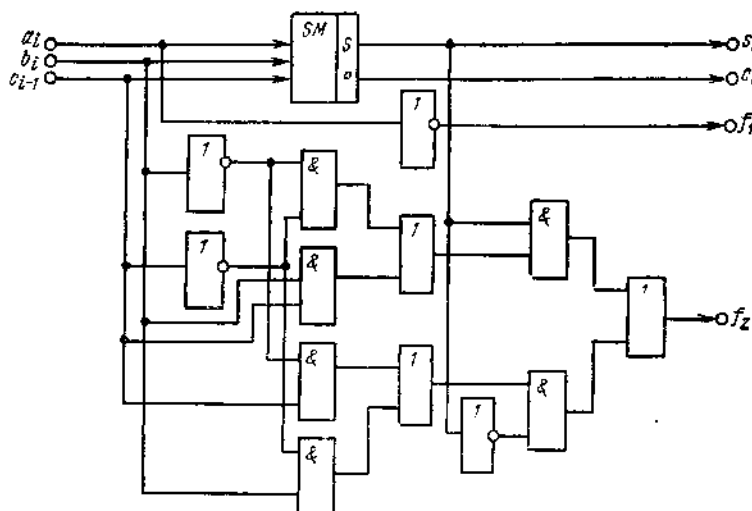


Рис. 5-2. Реализация самопроверяемой СВК для одноразрядного сумматора рис. 5-1,а.

Проверка показывает, что все наборы тестов для f_1 и f_2 содержатся среди входных наборов подтаблицы правильных значений табл. 5-2. Следовательно, построенная СВК для сумматора является самопроверяемой

Самопроверяемая СВК с древовидной структурой

Перейдем теперь к рассмотрению древовидных структур СВК. Заметим, что при фиксированном базисе (простом или сложном) реализация заданной функции древовидной структурой без разветвлений не всегда возможна (в отличие от тупиковых ДНФ в скобочной форме, которая всегда существует для любой функции даже в простом базисе).

Общее условие, выполнение которого необходимо для того, чтобы любая реализация функций f_1 и f_2 была самопроверяемой, состоит в следующем.

Для того, чтобы СВК с двумя выходами f_1 и f_2 была самопроверяемой при произвольной раздельной реализации функций f_1 и f_2 , необходимо, чтобы среди входных наборов $\{\alpha_{j1}, \dots, \alpha_{jn}, \beta_{j1}, \dots, \beta_{jn}\}, j=1, \dots, r$. Для каждой существенной переменной функций f_1 и f_2 су-

уществовала пара наборов, проверяющих эту существенную переменную на константы 0 и 1.

Заметим, что рассмотренная выше скобочная реализация тупиковых форм функций f_1 и f_2 удовлетворяет этому условию.

В соответствии с результатами [1-5, гл. 3] для древовидных схем указанное общее условие конкретизируется следующим образом: СВК с древовидной раздельной реализацией двух выходных функций f_1 и f_2 будет самопроверяемой, если для каждой существенной переменной функций f_1 и f_2 среди входных наборов $\{\alpha_{j1}, \dots, \alpha_{jn}, \beta_{j1}, \dots, \beta_{jk}\}$, $j=1, \dots, r$, найдется пара наборов, соседних по данной существенной переменной и таких, на которых соответствующая функция f_1 или f_2 имеет противоположные значения. Более того, если приведенное условие удовлетворяется, то в процессе правильного функционирования комбинационного устройства будут обнаруживаться (на выходах f_1 и f_2) не только одиночные, но также кратные константные неисправности каждой реализации f_1 и f_2 СВК.

Пример 5-3. Построим задание на синтез СВК с двумя выходами f_1 и f_2 для дешифратора, схема которого приведена на рис. 5-3, а ее обобщенный граф — на рис. 5-4, найдем древовидную реализацию построенной схемы и проверим, является ли последняя самопроверяемой

Для построения общей таблицы истинности дешифратора определим значения, реализуемые дешифратором на выходах z_1, \dots, z_8 на всех восьми наборах его входных переменных x_1, x_2, x_3 при поочередной замене в схеме дешифратора функций элементов И-НЕ₁, $i=2, 4, 6, \dots, 24$, на инверсные функции. В табл. 5-3 ради экономии места приведены только подтаблицы неправильных значений, полученные при инвертировании функций элементов 2, 24. Указанные в табл. 5-3 значения функций f'_1 и f'_2 относятся к рассматриваемому в конце главы примеру 5-16.

Закодируем табл. 5-3 значениями функций f_1 и f_2 следующим образом: в подтаблице правильных значений для первых четырех строк помещаем значения $f_1=1, f_2=0$, а для следующих четырех строк — значения $f_1=0, f_2=1$; в подтаблице неправильных значений для строк, содержащих по одной единице в столбцах z_1, z_2, z_3 и z_4 и одну единицу в столбцах z_5, z_6, z_7 и z_8 , помещаем значения $f_1=1, f_2=1$; в остальных строках подтаблиц неправильных значений помещаем значения $f_1=0, f_2=0$. Анализ табл. 5-3, проведенный по описанному выше алгоритму А5-1 определения существенных переменных, показывает, что для функции f_1 существенными являются переменные z_1, z_2, z_3 и z_4 , а для функции f_2 — переменные z_5, z_6, z_7 и z_8 .

Проверка показывает, что для функций f_1 и f_2 существуют реализации в виде древовидных схем без разветвлений на входах в сложном базисе:

$$f_1 = (z_1 \oplus z_3) \oplus (z_3 \oplus z_4); f_2 = (z_5 \oplus z_6) \oplus (z_7 \oplus z_8),$$

представленные на рис. 5-5 вместе со связями с выходами дешифратора (ДС).

Для каждой существенной переменной z_1, z_2, z_3 и z_4 функции f_1 в подтаблице правильных значений имеется пара соседних наборов значений этих переменных с противоположными значениями функции f_1 . Аналогично для каждой существенной переменной z_5, z_6, z_7 и z_8 функции f_2 также имеется пара наборов, обладающая указанными свойствами. Для z_1 и z_5 эти пары наборов находятся в первой и пятой строках табл. 5-3, для z_2 и z_6 — в строках 2 и 6, для z_3 и z_7 — строках 3 и 7, для z_4 и z_8 — строках 4 и 8. Поэтому получен-

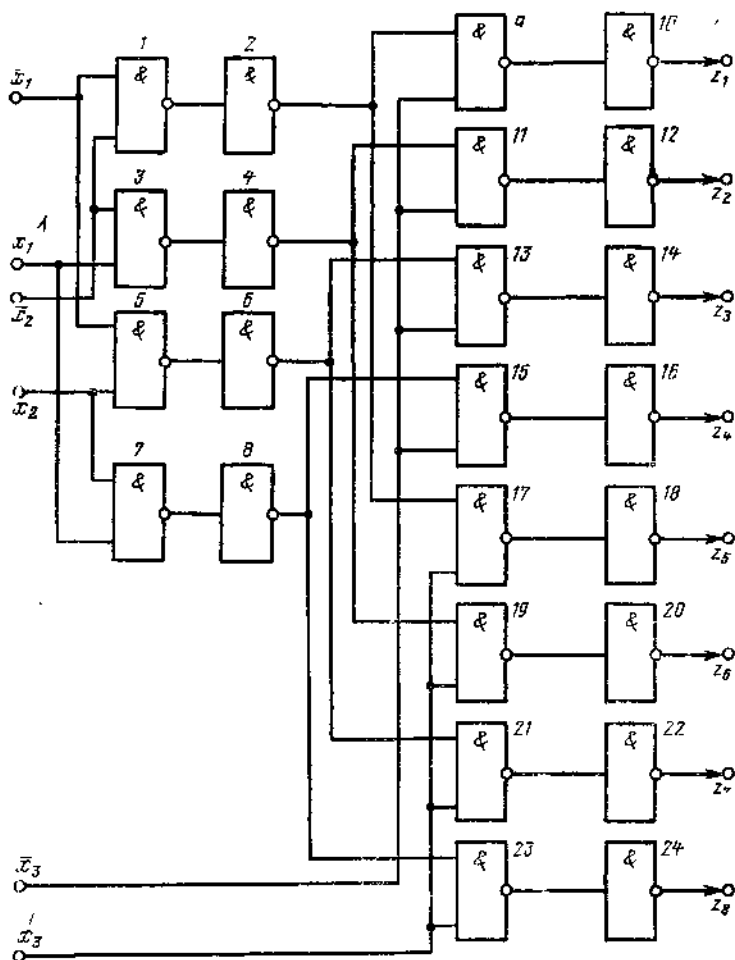


Рис. 5-3. Схема дешифратора.

Подчеркнем, что сформулированные условия самопроверяемости СВК, состоящие в наличии среди входных наборов правильных подтаблиц пар наборов, проверяющих существенные переменные, фактически реализуются в тех случаях, когда все такие пары наборов будут появляться на входах СВК в течение небольших отрезков времени при правильном функционировании контролируемого устройства. Если какие-нибудь пары наборов (или отдельные наборы пар) не появляются на входах СВК в течение длительного периода времени, то вероятность необнаружения в схеме проверяемых этими парами (этими наборами) неисправностей может стать зна-

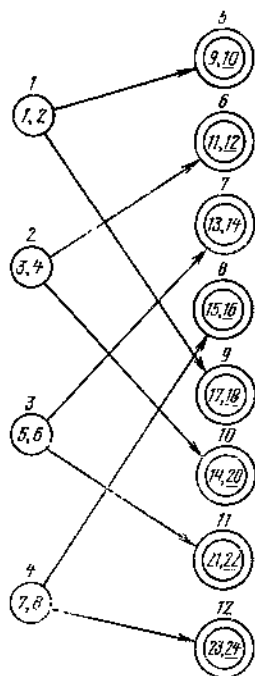


Рис. 5-4. Обобщенный граф схемы дешифратора рис. 5-3.

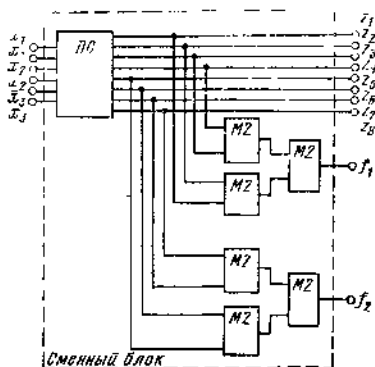


Рис. 5-5. Самопроверяемая СВК для дешифратора рис. 5-3.

чительной. Такие неисправности не будут обнаружены, и, следовательно, возможно возникновение кратных неисправностей СВК.

С учетом сказанного предпочтительной является реализация самопроверяемых СВК древовидными структурами. При других реализациях может потребоваться периодическое тестовое диагностирование СВК, при котором на схему будет подан проверяющий тест.

Таблица 5-3

x_1 x_2 x_3	z_1 z_2 z_3 z_4 z_5 z_6 z_7 z_8	f_1 f_2	f'_1 f'_2	Схема
0 0 0	1 0 0 0 0 0 0 0	1 0	1 0	Исправная
1 0 0	0 1 0 0 0 0 0 0	1 0	0 1	
0 1 0	0 0 1 0 0 0 0 0	1 0	0 1	
1 1 0	0 0 0 1 0 0 0 0	1 0	1 0	
0 0 1	0 0 0 0 1 0 0 0	0 1	0 1	
1 0 1	0 0 0 0 0 1 0 0	0 1	1 0	
0 1 1	0 0 0 0 0 0 1 0	0 1	1 0	
1 1 1	0 0 0 0 0 0 0 1	0 1	0 1	
0 0 0	0 0 0 0 0 0 0 0	0 0	0 0	С инвертором на выходе элемента И-НЕ ₂
1 0 0	1 1 0 0 0 0 0 0	0 0	1 1	
0 1 0	1 0 1 0 0 0 0 0	0 0	1 1	
1 1 0	1 0 0 1 0 0 0 0	0 0	~ ~	
0 0 1	0 0 0 0 0 0 0 0	0 0	~ ~	
1 0 1	0 0 0 0 1 1 0 0	0 0	~ ~	
0 1 1	0 0 0 0 1 0 1 0	0 0	~ ~	
1 1 1	0 0 0 0 1 0 0 1	0 0	~ ~	
...		
...		
...		
0 0 0	1 0 0 0 0 0 0 1	1 1	~ ~	С инвертором на выходе элемента И-НЕ ₂
1 0 0	0 1 0 0 0 0 0 1	1 1	~ ~	
0 1 0	0 0 1 0 0 0 0 1	1 1	~ ~	
1 1 0	0 0 0 1 0 0 0 1	1 1	1 1	
0 0 1	0 0 0 0 1 0 0 1	0 0	~ ~	
1 0 1	0 0 0 0 0 1 0 1	0 0	~ ~	
0 1 1	0 0 0 0 0 0 1 1	0 0	1 1	
1 1 1	0 0 0 0 0 0 0 0	0 0	0 0	

содержащий, в частности, все пары соседних наборов, проверяющих схему. Это целесообразно, например, для самопроверяемых схем защиты объекта с высокими показателями надежности: появление на входах схемы защиты такого объекта пар наборов (наборов), соответствующих аварийным ситуациям, будет чрезвычайно редким.

Рассмотрим случай, когда СВК имеет три выхода f'_1, f'_2, f'_3 . Закодируем общую таблицу истинности (см. табл. 5-1) значениями функций f'_1, f'_2, f'_3 следующим образом: в подтаблице правильных значений в каждой строке из трех значений $\gamma'_1, \gamma'_2, \gamma'_3$ Два значения сделаем равными единице и одно значение — нулю, в подтаблицах неправильных значений значения функций f'_1, f'_2 и f'_3 примем равными нулю. При отдельной реализации функций f'_1, f'_2, f'_3 условия построения самопроверяемой СВК аналогичны условиям, сформулированным выше для самопроверяемых СВК с двумя выходами.

Самопроверяемая СВК с тремя выходами f'_1, f'_2, f'_3 обладает следующими свойствами:

а) значения сигналов 110, 101 и 011 на выходах f'_1, f'_2, f'_3 свидетельствуют о правильном функционировании комбинационного устройства и об отсутствии одиночных неисправностей в СВК;

б) значения сигналов 000 свидетельствуют о неправильном функционировании комбинационного устройства;

в) остальные возможные значения сигналов 100, 010, 001 или 111 свидетельствуют о наличии одиночной неисправности в СВК.

Таким образом, в самопроверяемой СВК с тремя выходами имеется возможность различить, где появилась одиночная неисправность — в комбинационном устройстве или в СВК.

5-2. САМОПРОВЕРЯЕМЫЕ СВК ДЛЯ КОДОВ t ИЗ n

Коды t из n нашли широкое применение в дискретной (вычислительной) технике. Такими кодами называют множества наборов (кодовых слов), содержащих n двоичных символов (нулей и единиц), среди которых всегда t символов равны единице. Можно также рассматривать некоторое дискретное устройство, на выходах которого в процессе правильного функционирования

реализуется код m из n . Прежде чем перейти к анализу кодов m из n , приведем метод построения самопроверяемых СВК для кодов k из $2k$.

Самопроверяемые СВК для кодов k из $2k$

Пусть задано дискретное устройство (ДУ), имеющее $2k$ выходов z_1, \dots, z_{2k} . При правильном функционировании дискретного устройства на его выходах всегда присутствуют k единиц и k нулей. Число всех возможных различных кодовых слов при правильном функционировании дискретного устройства равно:

$$C_{2k}^k = \frac{(2k)!}{k!k!}.$$

Задача состоит в том, чтобы построить самопроверяемую СВК с двумя выходами, пары значений сигналов на которых равны 10 или 01 при наличии k единиц на $2k$ выходах дискретного устройства и 00 (11) при числе единиц, меньшем (большем) k . Очевидно, что если некоторая неисправность дискретного устройства проявляется на его выходах в виде неправильных значений, но при этом число единиц на выходах дискретного устройства не изменяется (равно k), то такая неисправность СВК не будет обнаружена.

Разобьем множество выходов $\{z_1, \dots, z_{2k}\}$ ДУ на два подмножества A и B с равным числом элементов в них $A = \{a_1, \dots, a_k\}$, $B = \{b_1, \dots, b_k\}$, причем $z_{11} = a_1, \dots, z_{1k} = a_k$; $z_{21} = b_1, \dots, z_{2k} = b_k$, где $\{z_{11}, \dots, z_{1k}\} \cup \{z_{21}, \dots, z_{2k}\} = \{z_1, \dots, z_{2k}\}$.

Если число единиц на выходах из подмножества A при правильном функционировании ДУ равно i ($i = 0, 1, \dots, k$), то число единиц на выходах подмножества B будет равно $k-i$.

Определим логическую функцию $T(k_a \geq i)$ следующим образом: функция $T(k_a \geq i) = 1$ в том и только в том случае, когда число единиц k_a на выходах из подмножества A в данный момент времени равно или больше некоторого заданного числа i . Аналогично, $T(k_b \geq k-i) = 1$ в том и только в том случае, если число единиц k_b на выходах из подмножества B в данный момент времени равно или больше некоторого заданного числа $k-i$.

Пример 5-4. Пусть задано дискретное устройство с шестью выходами z_1, \dots, z_6 . На выходах этого ДУ при правильном его функ-

ционировании всегда присутствуют три единицы и три нуля (реализуется код 3 из 6); $k=3$; $2k=6$. Пусть подмножество $A = \{a_1, a_2, a_3\}$ и подмножество $B = \{b_1, b_2, b_3\}$; $z_1=a_1$, $z_2=a_2$, $z_3=a_3$; $z_4=b_1$, $z_5=b_2$, $z_6=b_3$.

Зададим $i=2$, тогда $k-i=1$. Функции $T(k_a \geq i)$ и $T(k_b \geq k-i)$ имеют вид:

$$\begin{aligned} T(k_a \geq 2) &= a_1 a_2 \vee a_1 a_3 \vee a_2 a_3, \\ T(k_b \geq 1) &= b_1 \vee b_2 \vee b_3. \end{aligned}$$

Заметим, что по определению $T(k_a \geq 0) = T(k_b \geq 0) \equiv 1$.

Обозначим выходы самопроверяемой СВК через f_1 и f_2 . Определим функции f_1 и f_2 следующим образом:

$$f_1 = \bigvee_i T(k_a \geq i) T(k_b \geq k-i),$$

где $i = 1, 3, 5, \dots$, $\begin{cases} k, & \text{если } k - \text{нечетное,} \\ k-1, & \text{если } k - \text{четное;} \end{cases}$

$$f_2 = \bigvee_i T(k_a \geq i) T(k_b \geq k-i),$$

где $i = 0, 2, 4, \dots$, $\begin{cases} k, & \text{если } k - \text{четное,} \\ k-1, & \text{если } k - \text{нечетное.} \end{cases}$

Пример 5-5. Для рассматриваемого в примере 5-4 дискретного устройства:

для $i=1,3$

$$f_1 = T(k_a \geq 1) T(k_b \geq 2) \vee T(k_a \geq 3) T(k_b \geq 0),$$

для $i=0,2$

$$f_2 = T(k_a \geq 0) T(k_b \geq 3) \vee T(k_a \geq 2) T(k_b \geq 1).$$

После подстановки в эти формулы выражений для $T(k_a \geq i)$ и $T(k_b \geq k-i)$ получаем:

$$\begin{aligned} f_1 &= (a_1 \vee a_2 \vee a_3) (b_1 b_2 \vee b_1 b_3 \vee b_2 b_3) \vee (a_1 a_2 a_3) \cdot 1, \\ f_2 &= 1 \cdot (b_1 b_2 b_3) \vee (a_1 a_2 \vee a_1 a_3 \vee a_2 a_3) (b_1 \vee b_2 \vee b_3). \end{aligned}$$

Возможная реализация функции f_1 для нечетного k показана на рис. 5-6. Прямоугольниками, обозначенными символами $T(k_a \geq i)$ и $T(k_b \geq k-i)$, представлены двухуровневые схемы типа И-ИЛИ, элементы И которых реализуют конъюнкции соответствующих выражений для $T(k_a \geq i)$ и $T(k_b \geq k-i)$. Число схем И-ИЛИ равно числу нечетных значений $i = 1, 3, \dots, k$. Аналогично может быть реализована функция f_2 с числом схем И-ИЛИ, равным числу четных значений $i = 0, 2, \dots, k-1$.

Пример 5-6. Для рассматриваемого в примерах 5-4 и 5-5 дискретного устройства реализация функций f_1 и f_2 представлена на рис 5-7.

На этом рисунке соответствующие двухуровневые схемы И-ИЛИ реализуют функции $T(k_a \geq 1) = a_1 \vee a_2 \vee a_3$, $T(k_b \geq 2) = b_1 b_2 \vee b_1 b_3 \vee b_2 b_3$,

$$T(k_a \geq 3) = a_1 a_2 a_3, \quad T(k_b \geq 3) = b_1 b_2 b_3, \quad T(k_a \geq 2) = a_1 a_2 \vee a_1 a_3 \vee a_2 a_3, \\ T(k_b \geq 1) = b_1 \vee b_2 \vee b_3$$

Покажем, что если число единиц в кодовом слове, содержащем $2k$ двоичных символов ($2k$ значений выходных функций контролируемого устройства), равно k , то значения сигналов на выходах f_1, f_2 СВК равны 10 или 01. Пусть число единиц на выходах из множества A

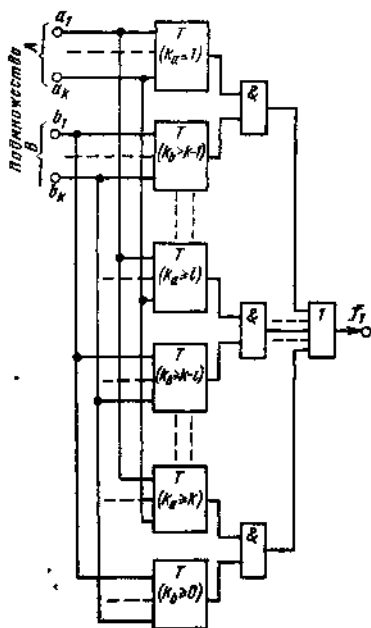


Рис 5-6 Реализация функции f_1 для нечетного k .

Покажем теперь, что СВК, выходные функции f_1 и f_2 которой реализованы в соответствии с рис. 5-6, является самопроверяемой, т. е. все ее одиночные константные неисправности будут обнаруживаться на всем множестве C_{2k}^h кодовых слов. Для этого необходимо и достаточно рассмотреть одиночные неисправности узлов разветвлений (входов схемы в целом) и ветвей разветвлений (входов схем, реализующих функции f_1 и f_2), так как схема рис. 5-6 имеет древовидную структуру с разветвлением только на входах.

Константные одиночные неисправности узлов разветвлений приводят (в моменты их появления) к изме-

равно i_1 , тогда число единиц на выходах из множества B равно $k - i_1$. Найдем подсхему СВК, реализующую конъюнкцию $T(k_a \geq i_1)T(k_b \geq k - i_1)$. Если i_1 — нечетное число, то эта подсхема принадлежит схеме для i_1 если i_1 — четное число, то — схеме для $i_1/2$. При $i_1 = i_1$ выход этой и только этой подсхемы будет равен 1. Поэтому при нечетном i_1 значения сигналов на выходах f_1, f_2 будут 10, а при четном i_1 — 01. Аналогично можно показать, что если число единиц в кодовом слове, содержащем $2k$ двоичных символов, меньше k (больше k), то значения сигналов на выходах f_1, f_2 СВК будут равны 00 (11).

нению числа единиц в поступающих на схему кодовых словах и поэтому будут обнаружены схемой, что было рассмотрено ранее.

В связи с рассмотрением константных неисправностей ветвей разветвлений отметим следующие свойства

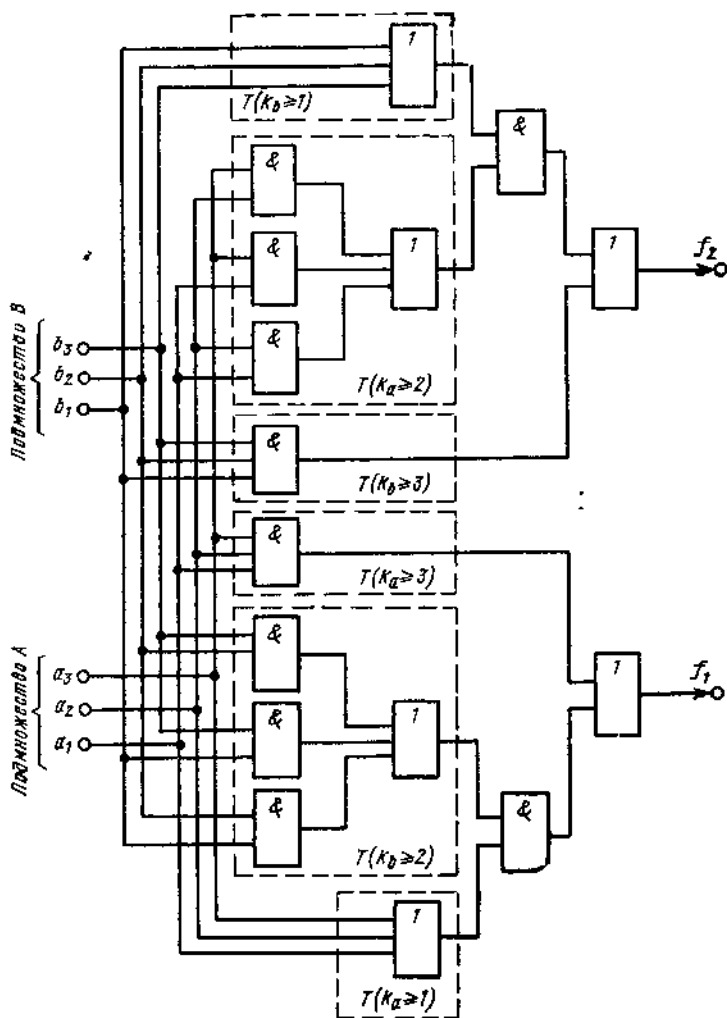


Рис 5-7. Самопроверяемая СВК для кода 3 из 6.

схем, реализующих функции f_1 и f_2 . Дизъюнктивные нормальные формы функций f_1 и f_2 , получаемые после раскрытия скобок в исходных выражениях для этих функций, содержат по $C_{2k}^k/2$ разных неизбыточных конъюнкций, каждая из которых имеет ровно k переменных. В исправной СВК при любом значении i равенство 1 функции f_1 или f_2 обеспечивается за счет равенства 1 единственной такой конъюнкции — назовем ее единичной. Пусть i нечетно, т. е. единичная конъюнкция принадлежит дизъюнктивной нормальной форме функции f_1 , и поэтому $f_1 = 1$, $f_2 = 0$. Каждой букве единичной конъюнкции соответствует определенный вход (ветвь разветвлений) схемы, реализующей функцию f_1 , а от каждого такого входа к выходу f_1 ведет простой (неразветвляющийся) путь. Поэтому появление неисправности, тождественно равной 0 на любом одном или на нескольких входах, представленных буквами единичной конъюнкции (или же на одном или нескольких узлах указанных простых путей), приводит к равенству нулю единичной конъюнкции (или же эквивалентно такому равенству) и как следствие к равенству 0 функции f_1 . Значение $f_2 = 0$ при этом не меняется, т. е. рассматриваемые неисправности обнаруживаются — выходы f_1 , f_2 принимают значения 00. Другие неисправности, тождественно равные нулю, будут обнаруживаться при других значениях i и других кодовых словах, поступающих на схему.

Предположим далее, что при некотором нечетном i (при этом в исправной схеме $f_1 = 1$, $f_2 = 0$) появилась неисправность, тождественно равная 1, на одном или на нескольких входах (ветвях разветвлений) схемы, реализующей функцию f_2 . Если эти входы имели значения 0 и, кроме того, появление на них значений, тождественно равных 1, привело к образованию хотя бы одной единичной конъюнкции в дизъюнктивной нормальной форме функции f_2 , то последняя становится равной 1, и неисправность обнаруживается — выходы f_1 , f_2 принимают значения 11. Другие неисправности, тождественно равные 1, будут обнаруживаться при других значениях i и других кодовых словах, поступающих на схему.

Аналогичные соображения могут быть приведены для четных значений i по обнаружению неисправностей, тождественно равных 0 в схеме для f_2 и тождественно равных 1 в схеме f_1 .

Самопроверяемые СВК для кодов 1 из n

Схема встроенного контроля для кода 1 из n строится путем преобразования на входе схемы этого кода в код k из $2k$. Преобразование осуществляется с помощью самопроверяемого транслятора.

В качестве примера рассмотрим код 1 из 8. Транслятор для этого кода, изображенный на рис. 5-8, преобразует код 1 из 8 в код 3 из 6. Любая одиночная неисправность на входах транслятора, а также на входах и выходах его логических элементов ИЛИ изменит число единиц в выходном кодовом слове.

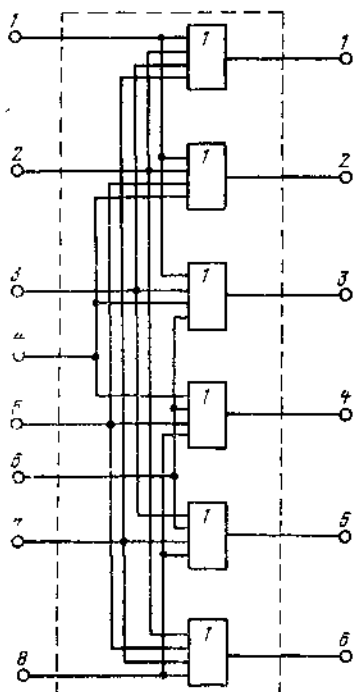


Рис. 5-8. Самопроверяемый транслятор кода 1 из 8 в код 3 из 6.

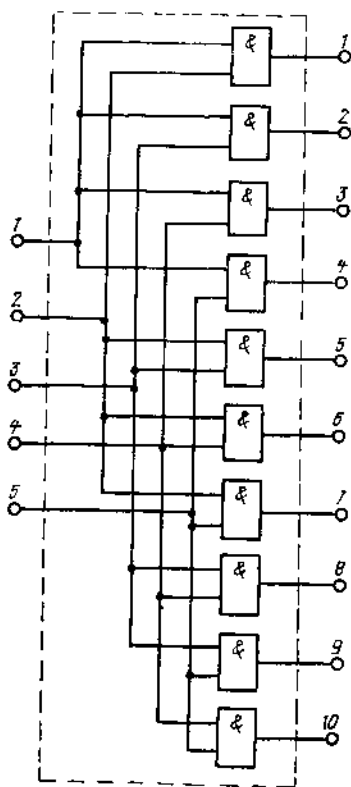


Рис. 5-9. Самопроверяемый транслятор кода 2 из 5 в код 1 из 10.

Входы схем ИЛИ транслятора объединены таким образом, что при правильном функционировании транслятора набор значений сигналов на левых трех выходах транслятора всегда является дополнительным (покомпонентно инверсным) набору значений на правых трех выходах. При этих условиях достаточно существующих восьми наборов (из общего числа $C_6^3=20$ возможных наборов) на выходе транслятора для того, чтобы СВК для кода 3 из 6 (эта схема приведена на рис. 5-7) была самопроверяемой. Подобным образом могут быть построены трансляторы для $8 < n \leq 20$, которые преобразуют код 1 из n также в код 3 из 6. Для $n > 20$ необходимо выбирать $2k > 6$.

При построении самопроверяемых СВК для кодов 1 из n путем их преобразования с помощью трансляторов в код k из $2k$ трудности возникают для значений $n=3$ и $n=7$. При $n=3$ число кодовых комбинаций (это число равно трем) недостаточно ни для какой самопроверяемой СВК k из $2k$. Для $n=7$ семь кодовых комбинаций не могут быть преобразованы в код 2 из 4, и, с другой стороны, этих кодовых комбинаций недостаточно для того, чтобы СВК для кода 3 из 6 была самопроверяемой.

Самопроверяемые СВК для кодов m из n

Задача построения самопроверяемой СВК непосредственно для кода m из n без его преобразования в некоторый другой код в настоящее время не решена.

Одним из возможных подходов к построению самопроверяемых СВК для таких кодов является использование трансляторов, преобразующих код m из n в код 1 из C_n^m . Этот транслятор строится с использованием C_n^m m -входовых схем И. Каждая схема И реализует только одну кодовую комбинацию из общего числа, равного C_n^m .

Построенный таким образом транслятор является самопроверяемым, так как на каждую схему И поступают все проверяющие ее тестовые наборы. В дальнейшем строится самопроверяемая СВК для кода 1 из C_n^m . На рис. 5-9 представлен самопроверяемый транслятор для преобразования кода 2 из 5 в код 1 из 10. Транслятор содержит $C_5^2=10$ двухвходовых элементов И.

Самопроверяемые схемы сжатия для кода 2 из 4

Схемы сжатия используются в качестве самопроверяемых СВК при применении метода дублирования. На рис. 5-10 представлена блок-схема реализации СВК для трехвыходного дискретного устройства (комбинационного или с памятью). Дискретное устройство $ДУ$ имеет три выхода z_1, z_2, z_3 ; $ДУ_2$, являющееся дубликатом $ДУ_1$ с инверторами на выходах, имеет выходы $z'_1 = \bar{z}_1, z'_2 = z_2, z'_3 = z_3$.

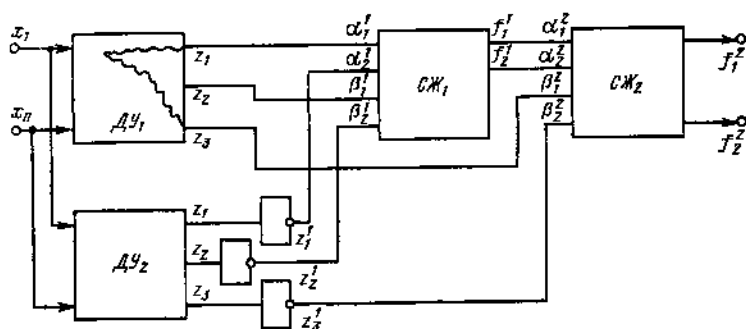


Рис. 5-10 Блок-схема реализации самопроверяемой СВК по методу дублирования

При правильном функционировании $ДУ_1$ и $ДУ_2$ на выходах схем сжатия $СЖ_1$ и $СЖ_2$ реализуются значения 10 и 01. Если в результате неисправности в $ДУ_1$ на выходах z_1 и z_3 появляются неправильные значения, то на вход $СЖ_1$ поступят следующие сигналы: $\alpha_1^1 = \alpha_2^1$ и $\beta_1^1 = \beta_2^1$. В результате на выходах f_1^1, f_2^1 $СЖ_1$ появится пара значений 00 или 11. На выходах f_1^2, f_2^2 схемы сжатия $СЖ_2$ появится пара значений 00 или 11 вне зависимости от значений сигналов на ее входах β_1^2 и β_2^2 (в данном случае $\beta_1^2 = \beta_2^2$).

На рис. 5-11, а и б представлены схемы сжатия, имеющие четыре входа $\alpha_1, \alpha_2, \beta_1, \beta_2$ и два выхода f_1 и f_2 . Таблица истинности для функций f_1, f_2 (левая пара столбцов /1, /2 соответствует схеме сжатия рис. 5-11, а, правая пара столбцов соответствует схеме сжатия рис. 5-11, б) приведена в табл. 5-4.

Таблица 5-4

α_1	α_2	β_1	β_2	f_1	f_2	f_1	f_2
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	1	1
0	1	0	0	0	0	0	0
0	1	0	1	1	0	1	0
0	1	1	0	0	1	0	1
0	1	1	1	1	1	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	1	0	1
1	0	1	0	1	0	1	0
1	0	1	1	1	1	1	1
1	1	0	0	0	0	1	1
1	1	0	1	1	1	1	1
1	1	1	0	1	1	1	1
1	1	1	1	1	1	1	1

Согласно этой таблице значения сигналов на выходах /1, /2 (рис. 5-11,а) равны 10 и 01 только на тех наборах значений входных переменных, на которых $\alpha_1 = \alpha_2$ и $\beta_1 = \beta_2$, т. е. когда $ДУ_1$ и $ДУ_2$ (см. рис. 5-10) выдают

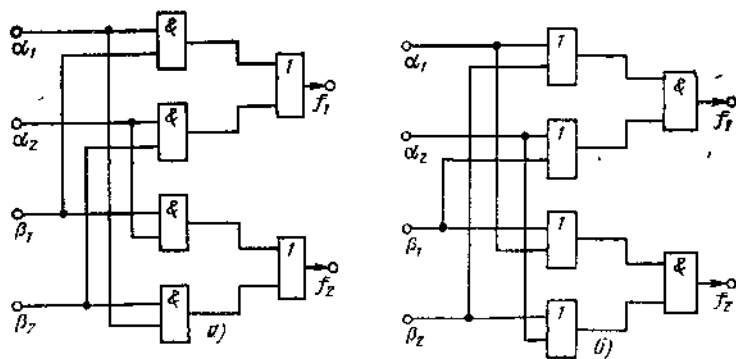


Рис. 5-11 Самопроверяемые схемы сжатия.

правильные значения выходных сигналов. Таким образом, каскадное сжатие выходных сигналов дублированных дискретных устройств $ДУ_1$ и $ДУ_2$ с помощью схем сжатия $СЖ$ обеспечивают обнаружение всех кратных неисправностей одного $ДУ$, а также несопряженных неисправностей обоих $ДУ$.

Схемы сжатия рис. 5-11, а и б являются самопроверяемыми, так как все их одиночные константные неисправности проявляются на выходах $f_1, /2$ в виде пар их значений 00 или 11 на четырех входных наборах $\alpha_1\alpha_2\beta_1\beta_2$: 0101, 0110, 1001, 1010.

5-3. СТРУКТУРНЫЙ МЕТОД ПОСТРОЕНИЯ САМОПРОВЕРЯЕМЫХ СВК ПО МОДУЛЮ 2 ДЛЯ КОМБИНАЦИОННЫХ УСТРОЙСТВ

Методы синтеза СВК с использованием общих таблиц истинности или таблиц переходов-выходов являются универсальными и пригодными для произвольных контролируемых дискретных устройств. Трудности практического использования этих методов определяются, с одной стороны, необходимостью построения общей таблицы истинности или таблицы переходов-выходов и, с другой стороны, традиционными трудностями реализации методов синтеза для сложных устройств.

В настоящем параграфе рассматривается практический метод, который пригоден для произвольных контролируемых комбинационных устройств и ориентирован на типовую реализацию СВК, представленную на рис. 5-12.

Для построения СВК при этом необходимо:

а) выходы z_1, \dots, z_h контролируемого $ДУ$ разбить на $h \geq 1$ групп так называемых независимых выходов;

б) каждую группу $\{z_{11}, \dots, z_{1p_1}\}, \dots, \{z_{h1}, \dots, z_{hp_h}\}$

независимых выходов объединить вместе при помощи схем свертки $M2$;

в) построить дополнительное дискретное устройство $ДДУ$ с h выходами z'_1, z'_2, \dots, z'_h , имеющими значения сигналов, инверсные значениям сигналов на выходах $\varphi_1, \varphi_2, \dots, \varphi_h$ схем свертки: $z'_1 = \varphi_1, \dots, z'_h = \varphi_h$;

г) соединить каскадно (по аналогии с рис. 5-10) $h-1$ четырехвыходовых схем сжатия (см. рис. 5-11) $СЖ_1, \dots, СЖ_{h-1}$, подав на их входы $\alpha^1_1, \alpha^1_2, \beta^1_1, \beta^1_2$, а также $\beta^2_1, \beta^2_2, \dots, \beta^{h-1}_1, \beta^{h-1}_2$ попарно выходы $ДДУ$ и выходы схем

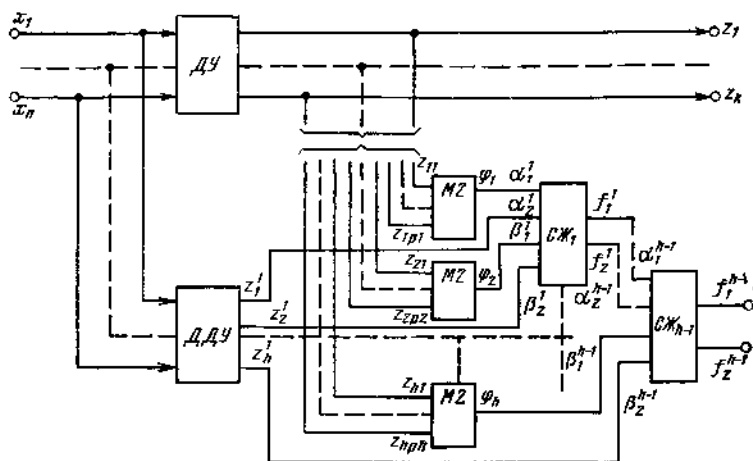


Рис 5-12 Типовая реализация самопроверяемой СВК для комбинационных устройств.

свертки $M2$. Выходы f_i^{h-1} и f_i^{h-2} последней схемы сжатия являются выходами самопроверяемой СВК для ДУ.

Наше внимание будет сосредоточено главным образом на получении функций φ_i и z'_i , $i=1, 2, \dots, h$, реализуемых схемами свертки и соответственно дополнительным дискретным устройством.

Определение групп независимых выходов

Определим понятия зависимых и независимых выходов комбинационного ДУ и их групп. Два выхода комбинационного устройства называются независимыми, если в подтаблицах неправильных значений общей таблицы истинности (табл. 5-1) нет ни одного входного набора, на котором оба выхода имеют неправильные значения. В противном случае эти выходы зависимы. Группа выходов комбинационного устройства образует *группу независимых выходов*, если все пары выходов группы независимы. В противном случае имеет место *группа зависимых выходов*.

Согласно приведенным определениям разбиение на группы независимых выходов можно провести по общей таблице истинности. Однако построение общей таблицы истинности достаточно трудоемко, поэтому процедуру

разбиения выходов комбинационного устройства на группы независимых выходов осуществим по обобщенному графу с использованием аппарата булевых производных.

Вершина i обобщенного графа называется графически существенной для вершины j , если в обобщенном графе существует хотя бы один путь от вершины i к вершине j ; в противном случае вершина i называется графически несущественной для вершины j .

Два выхода комбинационного устройства называются графически зависимыми (независимыми), если в обобщенном графе существует (не существует) вершина, графически существенная для обеих вершин, содержащих эти выходы. Два графически независимых выхода независимы; графически зависимые выходы могут быть как зависимыми, так и независимыми. Группа выходов комбинационного устройства называется группой графически независимых выходов, если все пары выходов группы графически независимы. В противном случае имеет место группа графически зависимых выходов.

Сначала разобьем все множество выходов комбинационного устройства на группы графически зависимых выходов. Построим для обобщенного графа матрицу соединений, в которой подчеркнем номера столбцов, соответствующих вершинам, содержащим элементы с внешними выходами.

Для нахождения групп графически зависимых выходов применим к матрице соединений следующий алгоритм А5-2.

1. Найти столбец, не содержащий ни одной единицы, и отметить его
2. Отметить строки, имеющие номера отмеченных столбцов.
3. Найти столбцы, которые содержат единицы в отмеченных строках, и отметить их: а) если такие столбцы есть, то перейти к п. 2; б) если таких столбцов нет, то перейти к п. 4.
4. Найти неотмеченные строки, содержащие единицы в отмеченных столбцах, и отметить их: а) если такие строки есть, то перейти к п. 5; б) если таких строк нет, то перейти к п. 6.
5. Отметить столбцы, имеющие номера отмеченных строк, перейти к п. 3.
6. Объединить номера подчеркнутых и отмеченных столбцов в одно подмножество и вычеркнуть из матрицы

соединений все отмеченные строки и столбцы. Перейти к п. 1.

После применения описанного алгоритма множество выходов комбинационного устройства будет разбито на группы графически зависимых выходов.

Обоснование алгоритма А5-2 состоит в следующем. Согласно п. 1 алгоритма находится вершина обобщенного графа, которая не имеет входов. После последовательного (возможно многократного) применения п. 2 и 3 алгоритма будут найдены все вершины (в том числе и содержащие внешние выходы), для которых вершина, найденная по п. 1 алгоритма, будет графически существенной. Пункты 4 и 5 алгоритма необходимы для нахождения вершин, которые также графически существенны для вершин, полученных на предыдущих шагах алгоритма. Дальнейшее применение пп. 3 и 2 алгоритма дает вершины, для которых вершина, найденная по п. 1, и вершины, найденные по п. 5, будут графически существенными. В результате будет получена группа вершин, содержащая внешние выходы, в которой для каждой вершины существует другая вершина в этой группе и такая, что обе эти вершины — графически зависимые.

Пример 5-7. Рассмотрим комбинационную схему (см рис. 4-2), обобщенный граф которой представлен на рис. 4-3.

На рис. 4-3 вершина 1 является графически существенной для вершин 3, 4, 5; вершина 2 — графически существенна для вершин 3, 4. Для обобщенного графа рис. 4-8 матрица соединений представлена в табл. 5-5.

Таблица 5-5

	✓	✓	✓	✓	✓
	1	2	3	4	5
✓ 1		1			1
✓ 2			1	1	
✓ 3					
✓ 4					
✓ 5					

Применение п. 1 алгоритма А5-2 выделяет вершину (столбец) 1. После последовательного применения п. 2 и 3 алгоритма все строки и столбцы окажутся отмеченными знаками ✓. Это значит, что вершина 1 является графически существенной для всех вершин, содер-

жащих внешние выходы. Поэтому для рассматриваемого примера получена одна группа графически зависимых выходов, в которую входят вершины 3, 4, 5, содержащие выходы z_1, z_2, z_3 .

Выходы, входящие в одну группу графически зависимых выходов, могут быть зависимыми и независимыми. Произведем выделение в группах графически зависимых выходов всех пар независимых выходов с использованием аппарата булевых производных [1-5].

Рассмотрим два выхода z_{i1} и z_{i2} комбинационного устройства, входящих в одну группу графически зависимых выходов. Найдем вершины обобщенного графа, являющиеся графически существенными для обоих выбранных выходов. Пусть этих вершин будет p и в них входят следующие подмножества элементов схемы устройства: $\{\eta_{11}^i, \dots, \eta_{1t_1}^i\}, \dots, \{\eta_{p1}^i, \dots, \eta_{pt}^i\}$, на выходах которых реализуются функции $\{y_{11}^i, \dots, y_{1t_1}^i\}, \dots, \{y_{p1}^i, \dots, y_{pt}^i\}$. Покажем, что выходы z_{i1} и z_{i2} независимы, если на всех входных наборах из множества $\{X\}_p$ имеет место равенство

$$\bigvee_{i=1}^p \frac{dz_{i1}}{dy_{it_1}^i} \frac{dz_{i2}}{dy_{it_1}^i} = 0,$$

и зависимы в противном случае. Справедливость приведенного утверждения определяется следующими соображениями.

Р а в е $\frac{dz_{i1}}{dy_{it_1}^i} \frac{dz_{i2}}{dy_{it_1}^i} = 1$ л я некоторого $i \in \{1,$

$2, \dots, p\}$ определяет наборы входных переменных устройства, на которых выходы z_{i1} и z_{i2} могут одновременно иметь неправильные значения за счет логических неисправностей элементов устройства, входящих в подмножество $\{\eta_{11}^i, \dots, \eta_{it_1}^i\}$. В этом случае выходы z_{i1} и z_{i2} зависимы. Если же выражение (5-1) равно нулю на множестве наборов $\{X\}_p$, то это значит, что ни при каких логических неисправностях элементов любого из подмножеств $\{\eta_{11}^i, \dots, \eta_{it_1}^i\}, \dots, \{\eta_{p1}^i, \dots, \eta_{pt}^i\}$ выходы z_{i1} и z_{i2} не имеют одновременно неправильных значений, т. е. выходы z_{i1} и z_{i2} независимы.

Пусть выходы z_1, \dots, z_k комбинационного устройства

по алгоритму А5-2 разбиты на 5 групп графически зависимых выходов, а IV каждой группе на основе равенства (5-1) определены все пары зависимых выходов. Построим граф, вершинам которого поставлены в соответствие выходы устройства, а ребрами соединены пары вершин, представляющих зависимые выходы. Воспользуемся далее каким-либо алгоритмом минимальной или минимизированной раскраски построенного графа, после чего образуем группы вершин, окрашенных одинаковым цветом. Эти группы вершин представляют искомые группы независимых выходов устройства.

Одним из алгоритмов минимизированной раскраски графа является следующий алгоритм А5-3:

1. Проверить наличие в графе несоседних (не соединенных ребром) вершин: а) если такие вершины есть, то перейти к п. 2; б) если таких вершин нет, то перейти к п. 3.

2. Взять две любые несоседние вершины графа и склеить (объединить) их в одну вершину, обозначив ее совокупностью обозначений склеиваемых вершин, после чего вернуться в п. 1.

3. Конец. Число вершин полученного графа есть число групп независимых выходов устройства. Совокупности обозначений вершин графа задают составы выходов соответствующих групп.

Заметим, что при наличии нескольких групп графически зависимых выходов имеется возможность применять алгоритм А5-3 сначала независимо к каждому подграфу, представляющему отдельную группу (эти подграфы в общем графе не связаны между собой), а затем — к графу, составленному из полученных результирующих подграфов. Это может оказаться полезным при очень большом числе выходов устройства.

Пример 5-8. В примере 5-7 для схемы (см рис 4-2) единственная группа графически зависимых выходов содержит выходы z_1 , z_2 , z_3 . Произведем разбиение этих выходов на группы независимых выходов

Для пары выходов z_1 и z_2 графически существенными вершинами являются (см рис 4-3) вершины 1 и 2, содержащие соответственно подмножества элементов $\{2\}$ и $\{4\}$; для пар выходов z_1 и z_3 , а также z_2 и z_3 графически существенной является вершина 1, содержащая подмножество элементов $\{2\}$. Таким образом, в соответствии с (5-1) необходимо проверить на равенство нулю следующие три выражения

$$\frac{dz_1}{dy_2} \frac{dz_2}{dy_2} \vee \frac{dz_1}{dy_4} \frac{dz_2}{dy_4};$$

$$\frac{dz_1}{dy_2} \frac{dz_2}{dy_2}, \frac{dz_2}{dy_2} \frac{dz_2}{dy_2},$$

где y_2 и y_4 — функции выходов элементов 2 и 4. Обращаясь к схеме на рис 4-2, находим

$$\begin{aligned} z_1 &= \overline{(x_1 \vee x_2)} (y_2 \vee x_3) = \overline{(x_1 \vee x_2)} y_4; \\ z_2 &= \overline{(y_2 \vee x_3 \vee x_4)} \overline{(y_2 \vee x_3 \vee x_4)} = \overline{(y_4 \vee x_1)} \overline{(y_4 \vee x_4)}; \\ z_3 &= y_2 x_3 x_4. \end{aligned}$$

По формуле вычисления булевой производной

$$\frac{df(x_1, \dots, x_i, \dots, x_n)}{dx_i} = f(x_i = 1) \oplus f(x_i = 0)$$

получаем:

$$\frac{dz_1}{dy_2} = \overline{(x_1 \vee x_2)} \oplus \overline{(x_1 \vee x_2)} x_3 = x_1 \bar{x}_2 \vee x_2 \bar{x}_3;$$

$$\frac{dz_1}{dy_4} = \overline{(x_1 \vee x_2)} \oplus \bar{0} = x_1 \vee x_2;$$

$$\frac{dz_2}{dy_2} = \bar{x}_1 \bar{x}_3 x_4 \vee x_1 \bar{x}_3 \bar{x}_4;$$

$$\frac{dz_2}{dy_4} = \bar{x}_1 x_4 \vee x_1 \bar{x}_4;$$

$$\frac{dz_3}{dy_2} = x_3 \bar{x}_4.$$

Тогда

$$\frac{dz_1}{dy_2} \frac{dz_2}{dy_2} \vee \frac{dz_1}{dy_4} \frac{dz_2}{dy_4} = x_1 \bar{x}_2 \vee \bar{x}_1 x_2 x_4;$$

$$\frac{dz_1}{dy_2} \frac{dz_3}{dy_2} = 0;$$

$$\frac{dz_2}{dy_2} \frac{dz_3}{dy_2} = 0.$$

Таким образом, зависимыми являются выходы z_1 и z_2 , выходы z_1 и z_3 , а также z_2 и z_3 — независимы. На рис 5-13,а представлен граф с вершинами z_1 , z_2 , z_3 и с единственной дугой, связывающей вершины z_1 и z_2 .

Обращаясь к операциям алгоритма А5-3, находим, что несоседние вершины в графе есть. Склеиваем вершины z_1 и z_3 и получаем показанный на рис 5-13,б граф, не имеющий несоседних вершин. Получены две группы независимых выходов $\{z_1, z_3\}$ и $\{z_2\}$. Если в графе рис 5-13,а склеить вершины z_2 и z_3 , то будет получен второй вариант групп независимых выходов — $\{z_2, z_3\}$ и $\{z_1\}$. Первый



Рис. 5-13 Граф анализа независимости выходов для схемы на рис. 4-2 (а) и граф со склеенными вершинами (б).

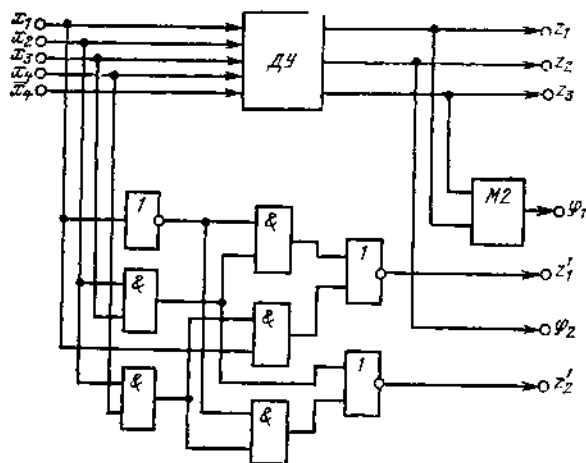


Рис. 5-14. Первый вариант реализации самопроверяемой СВК для схемы рис. 4-2.

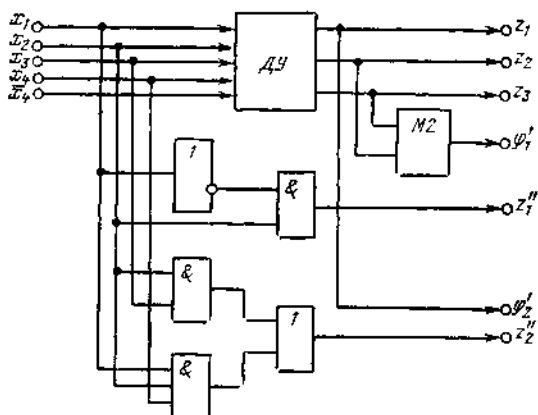


Рис. 5-15. Второй вариант реализации самопроверяемой СВК для схемы рис. 4-2.

вариант реализован в схеме на рис. 5-14, где на входы схемы свертки $M2$ поданы выходы z_1 и z_3 , а второй — в схеме на рис. 5-15, где на элемент $M2$ поданы выходы z_2 и z_3 .

Построение дополнительного дискретного устройства

Перейдем теперь к задаче построения дополнительного дискретного устройства (см. рис. 5-12). Как было сказано выше, выходные функции $ДДУ$ инверсны выходным функциям схем свертки $M2$. Входами $ДДУ$ являются внешние входы контролируемого устройства.

Задание на синтез $ДДУ$ можно получить в форме таблиц истинности. Для этого по значениям выходных функций исправного контролируемого устройства, приведенным в его таблице истинности (группа столбцов z_1, \dots, z_h в табл. 5-6), находят значения h выходных функций схем свертки (группа столбцов $\varphi_1, \dots, \varphi_h$ табл. 5-6), а инвертированием этих функций получают значения искомых выходных функций $ДДУ$ (группа столбцов x_1, \dots, x_n в табл. 5-6). Группа столбцов входных переменных x_1, \dots, x_h и группа столбцов выходных функций z'_1, \dots, z'_h табл. 5-6 образуют таблицу истинности, по которой может быть синтезировано $ДДУ$.

Таблица 5-6

x_1	\dots	x_n	z_1	\dots	z_h	φ_1	\dots	φ_h	z'_1	\dots	z'_h
α_{11}	\dots	α_{1n}	β_{11}	\dots	β_{1h}	0	\dots	1	1	\dots	0
\vdots		\vdots	\vdots		\vdots	\vdots		\vdots	\vdots		\vdots
α_{r1}	\dots	α_{rn}	β_{r1}	\dots	β_{rh}	1	\dots	0	0	\dots	1

Пример 5-9. Построим самопроверяемые схемы встроенного контроля для комбинационного устройства на рис. 4-2. Согласно первому варианту разбиения на группы независимых выходов $\{z_1, z_3\}, \{z_2\}$ (см. пример 5-8) необходимо реализовать следующие функции: $z'_1 = \varphi_1 = z_1 \oplus z_3$, $z'_2 = \varphi_2 = z_2$. Согласно второму варианту разбиения на группы независимых выходов — $\{z_2, z_3\}, \{z_1\}$ необходимо реализовать функции: $z''_1 = \varphi'_1 = z_1 \oplus z_3$, $z''_2 = \varphi'_2 = z_1$. Таблица 5-7 представляет собой аналог рассмотренной ранее табл. 5-6 для обоих вариантов реализации самопроверяемой схемы встроенного контроля.

Реализация ДДУ по первому варианту представлена на рис. 5-14, а по второму — на рис. 5-15. Реализация по рис. 5-15 предпочтительна по числу логических элементов.

Таблица 5-7

№ п/п	$x_1 \ x_2 \ x_3 \ x_4$	$z_1 \ z_2 \ z_3$	Первый вариант				Второй вариант			
			φ_1	φ_2	z'_1	z'_2	φ'_1	φ'_2	z''_1	z''_2
1	0 1 1 0	0 1 1	1	1	0	0	0	0	1	1
2	1 1 1 0	0 1 0	0	1	1	0	1	0	0	1
3	1 0 0 1	1 0 1	0	0	1	1	1	1	0	0
4	0 1 0 1	1 1 1	0	1	1	0	0	1	1	0
5	1 1 0 1	0 0 1	1	0	0	1	1	0	0	1
6	0 0 1 1	1 0 1	0	0	1	1	1	1	0	0

Отметим, что СВК при ее типовой реализации вида представленной на рис. 5-12 будет самопроверяемой даже при произвольной не раздельной реализации выходных функций z'_1, \dots, z'_n ДДУ если в процессе правильного функционирования контролируемого устройства на схемы свертки поступают их проверяющие тесты.

Оба варианта схем рис. 5-14 и 5-15 самопроверяемы, так как на каждый вход схем свертки в процессе правильного функционирования контролируемого устройства поступают пары соседних наборов. Так, для схемы рис. 5-14 на вход схемы свертки с выходом Φ_1 поступают наборы 4 и 5 (см. табл. 5-7) — 11 и 01 (z_1, z_3), проверяющие вход z_1 , а также наборы 1 и 2 — 01 и 00 (z_1, z_3), проверяющие вход z_3 . Выход Φ_2 схемы также проверяется на наборах, например 2 и 3 — 1,0 (z_2). Аналогичны наборы 4 и 5 — 11 и 01 (z_2, z_3), а также наборы 1 и 2 — 11 и 10 (z_2, z_3) проверяют входы схемы свертки с выходом Φ'_1 схемы на рис. 5-15. Выход Φ'_2 проверяется в этой схеме, в частности, на наборах 1 и 3 — 0 и 1 (z_1).

Заметим, что приведенные условия самопроверяемости схем рис. 5-14 и 5-15 удовлетворяются относительно константных одиночных и кратных неисправностей на входах и выходах схем свертки, представляющих собой при таком рассмотрении древовидные структуры в сложном базисе. При реализации двухвходового элемента, реализующего сложение по модулю 2, схемой в простом (например, булевом) базисе и при требовании обнаружения указанных неисправностей не только на внешних входах и выходах схемы, но также на ее внутренних узлах необходим (тривиальный) проверяющий тест, содержащий все четыре входных набора (а не три набора, как в рассмотренных схемах рис. 5-14 и 5-15).

Рассмотрим теперь пример, в котором задание на синтез дополнительного комбинационного устройства будет получено аналитически (без использования таблицы истинности).

Пример 5-10. Построим самопроверяемую схему встроенного контроля для одноразрядного сумматора рис. 5-16, а. Обобщенный граф для рассматриваемой схемы изображен на рис. 5-16, б. Множество

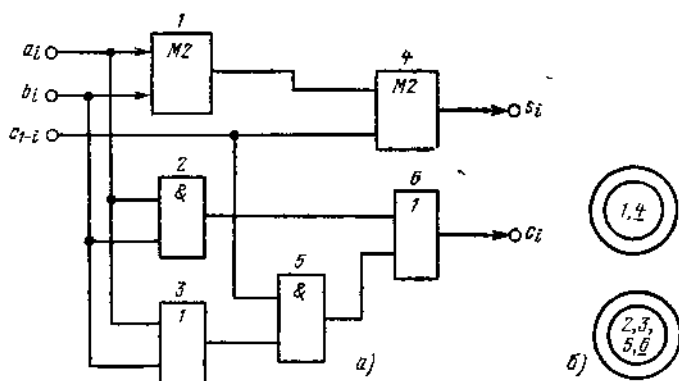


Рис 5-16. Схема одноразрядного сумматора (а) и обобщенный граф (б).

элементов сумматора оказалось разбитым на 2 подмножества $\{1, 4\}$ и $\{2, 3, 5, 6\}$. В соответствии с обобщенным графом выходы s_i и c_i графически независимы и, следовательно, независимы. Объединим выходы s_i и c_i по схеме свертки по модулю 2 и найдем функцию Φ , реализуемую на выходе схемы свертки:

$$\begin{aligned}\varphi &= s_i \oplus c_i = a_i \oplus b_i \oplus c_{i-1} \oplus (a_i b_i \vee (a_i \vee b_i) c_{i-1}) = \\ &= \bar{a}_i b_i c_{i-1} \vee a_i \bar{b}_i c_{i-1} \vee a_i b_i \bar{c}_{i-1} \vee \bar{a}_i \bar{b}_i c_{i-1} \vee \\ &\quad \vee \bar{a}_i b_i \bar{c}_{i-1} \vee a_i \bar{b}_i \bar{c}_{i-1}.\end{aligned}$$

Чтобы получить аналитическое выражение для выходной функции z дополнительного дискретного устройства ДДУ, необходимо взять инверсию функции φ :

$$z = \bar{\varphi} = \bar{a}_i \bar{b}_i \bar{c}_{i-1} \vee a_i b_i c_{i-1}.$$

Выражение для функции z в общем случае при необходимости может быть минимизировано, после чего по нему строится схема ДДУ. На рис. 5-17 представлена реализация самопроверяемой СВК для рассматриваемого примера комбинационного сумматора. При правильном функционировании сумматора на элемент M2 поступают все четыре набора значений его входных переменных s_i , c_i . Поэтому этот элемент будет проверяться полностью на все возможные кратные константные (а также логические) неисправности.

В заключение параграфа опишем класс неисправностей, обнаруживаемых на выходах φ_i и z'_i , $i=1, \dots, h$ (или, что то же, на выходах f_1^{n-1} и f_1^{n-2} при ис-

пользовании схем сжатия), самопроверяемой СВК при ее типовой реализации (см. рис. 5-12). В этот класс входят следующие неисправности:

1. Все одиночные логические неисправности комбинационного устройства.

2. Такие кратные логические неисправности комбинационного устройства, которые не изменяют одновременно значения ни одной пары выходов устройства, входящих

в одну и ту же группу независимых выходов.

В случае, если каждая группа независимых выходов содержит по одному выходу, самопроверяемой СВК будут обнаруживаться все кратные логические и константные неисправности контролируемого комбинационного устройства (при этом число выходов ДДУ равно числу выходов ДУ, а возможной реализацией ДДУ может быть дублирование ДУ).

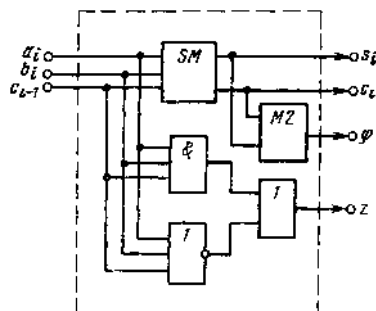


Рис. 5-17. Самопроверяемая СВК для сумматора рис. 5-16,а.

3. Все кратные существенные логические и константные неисправности дополнительного комбинационного устройства.

4. Все одиночные (кратные) константные неисправности схем свертки, если на последние в процессе правильного функционирования контролируемого комбинационного устройства поступает проверяющий (полный проверяющий) тест.

5-4. СТРУКТУРНЫЙ МЕТОД ПОСТРОЕНИЯ САМОПРОВЕРЯЕМЫХ СВК ПО МОДУЛЮ 2 ДЛЯ синхронных АВТОМАТОВ С ПАМЯТЬЮ

Рассмотрим процедуру построения самопроверяемых СВК для автоматов с памятью.

Пусть задан синхронный автомат A в виде структурной схемы, а также таблицы переходов-выходов вида табл. 4-7 из гл. 4. Автомат A имеет входы x_1, \dots, x_n , выходы z_1, \dots, z_k и внутренние переменные y_1, \dots, y_m .

На рис. 5-18 представлена типовая реализация самопроверяемой СВК для автомата с памятью A , где ДУ —

ции автоматов A и A_n , а также отсутствии схем сжатия.

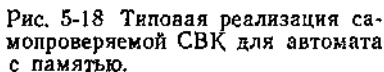


Рис. 5-18 Типовая реализация самопроверяемой СВК для автомата с памятью.

Определение групп независимых выходов

Как и для комбинационных устройств, разбиение выходов контролируемого автомата с памятью на группы независимых выходов можно проводить по обобщенному графу (см. гл. 4) схемы автомата в три этапа:

1. Разбить выходы автомата на группы графически зависимых выходов.
2. Определить для каждой пары выходов каждой группы графически зависимых выходов, являются ли выходы этой пары зависимыми или независимыми.
3. Сформировать группы независимых выходов.

По обобщенному графу структурной схемы автомата Л производится разбиение выходов автомата на группы графически зависимых выходов по алгоритму, аналогичному рассмотренному в § 5-3 алгоритму А5-2 для комбинационных устройств. Поскольку в матрице соединений, построенной по обобщенному графу автомата, может не оказаться столбца, не содержащего единиц (за счет обратных связей), то п. 1 алгоритма А5-2 следует изменить. В измененной формулировке п. 1 алгоритма А5-2 имеет вид:

1. Отметить любой столбец, номер которого не подчеркнут. Напомним, что столбцы с подчеркнутыми номерами соответствуют вершинам обобщенного графа, содержащим внешние выходы. Далее применяются все остальные пункты алгоритма А5-2 в том виде и в той

последовательности, как они изложены в описании алгоритма.

Выполним, как было описано в § 4-2 гл. 4, построение частичных таблиц переходов-выходов (типа табл. 4-9) неисправных автоматов путем моделирования поведения автомата с инвертированием выходов его максимальных подсхем (для каждой из B одиночных логических неисправностей выходов максимальных подсхем).

Рассмотрим два выхода z_{i1} и z_{i2} автомата, принадлежащих i -й группе графически зависимых выходов.

Для определения зависимости выходов z_{i1} и z_{i2} рассмотрим содержимое клеток (X_w, Y_j) табл. 4-8 и 4-9 ($w=1, \dots, r; j=1, \dots, q$). Сравним покомпонентно значения выходов Z_{wj}^0 и Z_{wj}^b , $b=1, 2, \dots, B$. Если среди неправильных значений Z_{wj}^b хотя бы одного $b \in \{1, 2, \dots, B\}$ значения обеих компонент z_{i1} и z_{i2} неправильны, то выходы z_{i1} и z_{i2} зависимы. Построим граф, k вершин которого обозначены символами выходов z_1, \dots, z_k , а пары вершин, соответствующих зависимым выходам, соединены ребрами. Применим к этому графу описанный выше алгоритм А5-3 и найдем группы независимых выходов. Для каждой группы $\{z_{11}, \dots, z_{1p_1}\}, \dots, \{z_{h1}, \dots, z_{hp_h}\}$ независимых выходов вычислим функции

$$\varphi_1 = z_{11} \oplus \dots \oplus z_{1p_1}, \dots, \varphi_h = z_{h1} \oplus \dots \oplus z_{hp_h},$$

реализуемые схемами свертки.

Построение дополнительного дискретного устройства

Теперь можно построить таблицу переходов-выходов автомата A_d . По числу строк и столбцов, а также по переходам эта таблица полностью совпадает с таблицей переходов-выходов правильно функционирующего автомата A . Число выходов автомата A_d равно h , а их функции z'_1, \dots, z'_h инверсны функциям $\varphi_1, \dots, \varphi_h$. В соответствии с этим для заполнения значения выхода Z'_{wj} клетки (X_w, Y_j) таблицы переходов-выходов автомата A_d производятся вычисление и инвертирование значений функций $\varphi_1, \dots, \varphi_h$ по значениям соответствующих компонент $\{z_{11}, \dots, z_{1p_1}\}^0, \{z_{h1}, \dots, z_{hp_h}\}^0$ вектора Z_{wj}^0 , записанного в клетке (X_w, Y_j) таблицы переходов-выходов правильно функционирующего автомата A . Полу-

ченная таблица переходов-выходов для A_d может быть минимизирована, и по ней строится схема, реализующая автомат A_d с выходами z'_1, \dots, z'_h .

Пример 5-11. Построим самопроверяемую СВК для кольцевого регистра сдвига, структурная схема которого приведена на рис. 4-10, обобщенный граф — на рис. 4-11, а таблицей переходов-выходов является табл. 4-11.

Обращаясь к обобщенному графу схемы регистра, находим, что все выходы z_1, z_2, z_3 и z_4 образуют единственную группу графически зависимых выходов.

Для анализа зависимости выходов z_1, z_2, z_3, z_4 между собой воспользуемся результатами моделирования регистра, проведенного ранее при рассмотрении примера 4-5. Результаты моделирования представлены в табл. 4-12, содержащей четыре (по числу максимальных подсем) частичные таблицы переходов-выходов.

Сравним покомпонентно значения векторов выходов каждой клетки (X_w, Y_j) $w=1, 2, j=1, 2, 3, 4$ табл. 4-11 и соответствующих клеток частичных таблиц переходов-выходов из табл. 4-12. Проводим такое сравнение для каждой пары выходов (компонент) $\{z_1, z_2\}, \{z_1, z_3\}, \{z_1, z_4\}, \{z_2, z_3\}, \{z_2, z_4\}, \{z_3, z_4\}$, проверяя каждый раз, принимают ли оба выхода пары одновременно неправильные значения в табл. 4-12. В частности, для клетки (1,1) имеем: $Z^0_{11}=1100$ и $Z^1_{11}=0100, Z^2_{11}=1000, Z^3_{11}=1110, Z^4_{11}=1101$. Значения пары выходов $\{z_1, z_2\}$ в Z^0_{11} — 11, а в «неправильных» векторах — 01, 10, 11, 11 соответственно, т. е. выходы z_1 и z_2 оба одновременно не принимают неправильных значений в клетке (1,1).

Выполнив указанные операции для всех пар выходов и для всех клеток табл. 4-11 и 4-12, найдем, что все выходы z_1, z_2, z_3, z_4 независимы между собой, т. е. составляют одну группу независимых выходов. Поэтому нет необходимости строить граф, отображающий зависимости выходов регистра, и обрабатывать его по алгоритму А5-3.

Построим таблицу переходов-выходов автомата A_d . Заготовим таблицу (табл. 5-8), содержащую две строки для входов $\{X_v\}_p$ и четыре столбца для состояний $\{Y_v\}$. Присвоим столбцам табл. 5-8 те же номера состояний, которые присвоены столбцам табл. 4-11, а во все клетки занесем те же номера состояний, которые содержатся в соответствующих клетках табл. 4-11. Например, в клетку (1,1) табл. 5-8 занесено состояние 2, так как в клетке (1,1) табл. 4-11 содержится состояние 0110, номер которого 2.

Т а б л и ц а 5-8

Y_{v+1}/Z'_v			$\{Y_v\}$			
			1	2	3	4
			1100	0110	0011	1001
$\{X_v\}_p$	1	0	1/11	2/00	3/11	4/00
	2	1	2/11	3/00	4/11	1/00

Определим по табл. 4-11 значения функции $\varphi = z_1 \Phi z_2 \Phi z_3 \oplus z_4'$, подлежащей реализации схемой свертки. Вычисления показывают, что для каждой клетки табл. 4-11 $\Phi=0$ (в каждой клетке два выхода имеют нулевые значения и два — единичные). Поэтому получаем $\Phi \equiv 0$, а это означает, что необходимое условие самопроверяемости СВК (см § 5-1) не выполняется. Это условие состоит в том, что в процессе функционирования на каждом выходе самопроверяемой СВК должны быть как единичные, так и нулевые значения сигналов.

В таких ситуациях группы независимых выходов, для которых оказалось $\Phi \equiv 0$ (или $\Phi \equiv 1$), следует попытаться разбить на две или большее число (вплоть до числа выходов в группе) подгрупп независимых выходов и реализовать функции Φ отдельно для каждой подгруппы.

Разобьем группу $\{z_1, z_2, z_3, z_4\}$ независимых выходов регистра на две подгруппы: $\{z_1, z_2\}$ и $\{z_3, z_4\}$. Вычислим для каждой клетки табл. 4-11 значения функций $\varphi_1 = z_1 \oplus z_2$ и $\varphi_2 = z_3 \oplus z_4$. Теперь φ_1 и φ_2 не являются константами. Поместим в клетки табл. 5-8 значения выходных функций z'_1 и z'_2 автомата A_d , которые инверсны значениям φ_1 и φ_2 , найденным для соответствующих клеток табл. 4-11, $z'_1 = \varphi_1$, $z'_2 = \varphi_2$.

После минимизации табл. 5-8 (состояния 1 и 3, а также 2 и 4 эквивалентны) будет получена таблица переходов-выходов дополнительного автомата A_d , представленная табл. 5-9.

Автомат A_d может быть реализован одним триггером со счетным выходом. На рис. 5-19 представлена полученная типовая реализация СВК для регистра сдвига. Схема самопроверяема, так как на каждую схему свертки по модулю 2 поступают все четыре набора значений их входных переменных z_1, z_2 и z_3, z_4 .

В § 5-6 будет получена более простая реализация самопроверяемой СВК для регистра сдвига рис 4-10.

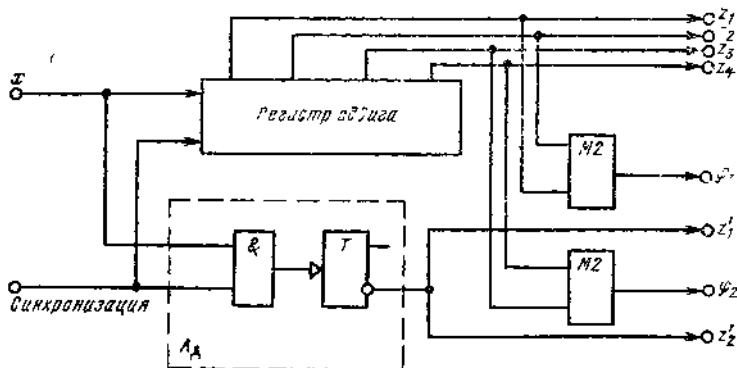


Рис. 5-19. Самопроверяемая СВК для регистра сдвига рис. 4-10.

Таблица 5-9

$Y_{v+1}/z'_1 z'_2$		$\{Y_v\}$	
		1	2
$\{X_v\}_p$	0	1/11	2/00
	1	2/11	1/00

5-5. ТИПОВАЯ РЕАЛИЗАЦИЯ САМОПРОВЕРЯЕМЫХ СВК для кодов t из n

В § 5-2 были рассмотрены методы построения самопроверяемых СВК для кодов t из n . Эти схемы предназначены для обнаружения любого некодового слова, число единиц в котором отлично от t . Это значит, что СВК для кодов t из n обнаруживают все *однонаправленные ошибки*. Однако, если необходимо обнаруживать только одиночные ошибки в кодовых словах, процедура построения самопроверяемых СВК существенно упрощается.

Разобьем n заданных контролируемых выходов на два подмножества $A=\{a_1, \dots, a_m\}$, $B=\{b_1, \dots, b_{n-m}\}$ так, что в первое подмножество входят точно t выходов и во второе — $(n-t)$ выходов.

Построим следующие функции f_1 и f_2 , являющиеся выходными функциями самопроверяемой СВК для кодов t из n :

$$\left. \begin{aligned} a) \quad f_1 &= a_1 \oplus \dots \oplus a_m, \\ f_2 &= b_1 \oplus \dots \oplus b_{n-m}, \end{aligned} \right\} \text{если } t - \text{нечетное число,}$$

или

$$\left. \begin{aligned} б) \quad f_1 &= a_1 \oplus \dots \oplus a_m, \\ f_2 &= b_1 \oplus \dots \oplus b_{n-m}, \end{aligned} \right\} \text{если } t - \text{четное число.}$$

На рис. 5-20, а и б представлены схемы реализации указанных функций при нечетном и четном t .

Рассмотрим реализацию, представленную на рис. 5-20, а. Если t — нечетное число, то при четном (нечетном) числе единиц в подмножестве A в процессе правильного функционирования источника кода t из n в подмножестве B будет нечетное (четное) число еди-

ниц. В результате на выходах f_1, f_2 будут существовать пары значений 01 и 10. Если число единиц в коде уменьшится или увеличится на одну единицу, то на выходах f_1, f_2 появится пара значений 00 или 11. Это значит, что рассматриваемая СВК будет обнаруживать все одиночные ошибки в коде m из n .

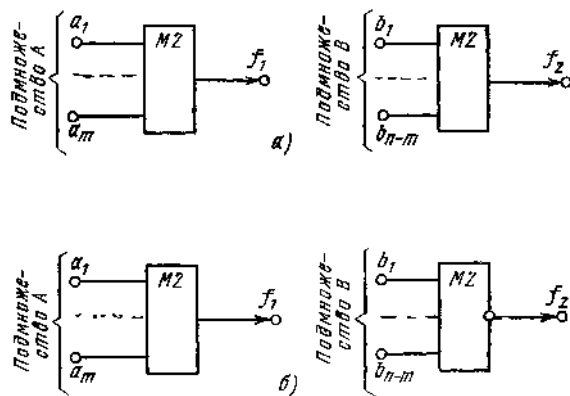


Рис. 5-20. Реализация функций f_1 и f_2 .
а — при нечетном m ; б — при четном m .

Если контролируемые выходы являются выходами дискретного устройства и эти выходы составляют одну группу независимых выходов, то СВК будет обнаруживать все одиночные логические неисправности дискретного устройства.

Рассмотрим реализацию рис. 5-20,б. Если m — четное число, то в процессе правильного функционирования в обоих подмножествах A и B четность числа единиц будет одинакова. В результате снова на выходах f_1, f_2 будут существовать пары значений 01 и 10. Если число единиц в коде уменьшится или увеличится на одну единицу, то на выходах f_1, f_2 появится пара значений 00 или 11. Теперь необходимо показать, что рассматриваемая СВК является самопроверяемой.

Пусть $m \leq (n-1)/2$, если n — нечетное, или $m < n/2$, если n — четное число. Это значит, что число единиц в коде m из n меньше числа нулей. Тогда на входах схемы свертки, реализующей функцию f_1 , будут появляться наборы, содержащие одну единицу во всех возможных сочетаниях с $m-1$ нулем; две единицы во всех возможных сочетаниях с $m-2$ нулями и т. д. Кроме

того, на входах схемы свертки будут также наборы, состоящие из всех нулей и всех единиц. Поскольку $C_m^0 + C_m^1 + \dots + C_m^m = 2^m$, то на входах схемы свертки, реализующей функцию /1, в процессе правильного функционирования появятся все 2^m возможных входных наборов. Это значит, что на выходе f_1 будут обнаружены все возможные одиночные и кратные логические и константные неисправности схемы свертки, реализующей функцию /1.

Рассмотрим схему свертки, реализующую функцию f_2 (рис. 5-20, а б). Покажем, что для каждого входа этой схемы свертки среди входных наборов всегда существуют два набора, соседних по данному входу, и на этих двух наборах функция /2 изменяет свое значение. Действительно, пусть на некоторый вход b_i , $i \in \{1, \dots, n-m\}$ схемы свертки, реализующей функцию /2, поступает набор, в котором $b_i = 1$. Такой набор всегда существует. Так как число единиц в коде меньше числа нулей, то среди наборов, поступающих на схему свертки, всегда существует набор, соседний по входу b_i , т. е. на этом наборе $b_i = 0$, и на этих двух соседних наборах функция f_2 изменяет свое значение (поскольку число единиц в группе B изменяется на единицу). Доказанное для схемы свертки утверждение является достаточным условием обнаружения всех одиночных и кратных константных неисправностей входов и выходов (узлов) элементов СВК, имеющей реализацию в виде древовидной структуры без разветвлений на входах и построенной на элементах простого или сложного базиса (см. § 5-1). Поэтому на выходе f_2 будут обнаружены все одиночные и кратные константные неисправности входов и выходов элементов «сумма по модулю 2», на которых построена схема свертки, реализующая функцию f_2 .

Пусть $m \geq (n+1)/2$, если n нечетное, или $m > n/2$, если n четное число. Это значит, что число единиц в коде m из n больше числа нулей.

В этом случае на выходе f_1 будут обнаружены все одиночные и кратные константные неисправности входов и выходов элементов «сумма по модулю 2», на которых построена схема свертки, реализующая функцию f_1 . На выходе f_2 будут обнаружены все возможные одиночные и кратные логические неисправности схемы свертки, реализующей функцию f_2 .

И, наконец, рассмотрим случай, когда число единиц в коде m из n равно числу нулей. В этом случае мы

имеем код k из $2k$. Тогда $2k$ выходов разбиваются на два подмножества A и B равной мощности: $A=\{a_1, \dots, a_k\}$, $B=\{b_1, \dots, b_k\}$. Реализации самопроверяемой СВК для нечетного и четного k аналогичны реализациям, приведенным на рис. 5-20, а, б при $m=n-m=k$. На входах каждой схемы свертки, реализующих функции f_1 и f_2 , в процессе правильного функционирования появятся все 2^k возможных входных наборов. Поэтому на выходе $f_1(f_2)$ будут обнаружены все возможные одиночные и кратные логические и константные неисправности схемы свертки, реализующей функцию $f_1(f_2)$.

Пример 5-12. Построим самопроверяемую СВК для кода 3 из 6. Поскольку $k=3$ — нечетное число, то типовая реализация строится

в соответствии с рис. 5-20, а. На рис. 5-21 представлена типовая реализация самопроверяемой СВК для кода 3 из 6 древовидными структурами на двухвходовых элементах сложения по модулю 2.

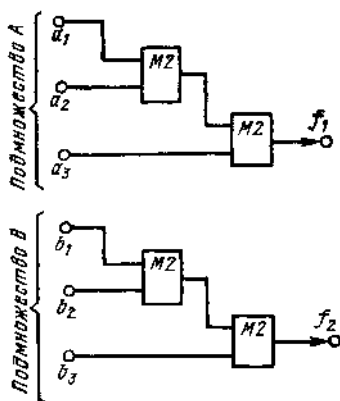


Рис. 5-21. Типовая реализация самопроверяемой СВК для кода 3 из 6.

Пример 5-13. Построим самопроверяемую схему встроенного контроля для дешифратора, схема которого дана ранее на рис. 5-3. Разобьем выходы дешифратора z_1, \dots, z_8 на группы независимых выходов. Согласно обобщенному графу (рис. 5-4) все выходы дешифратора разбиваются на следующие группы графически зависимых выходов: $\{z_1, z_5\}$, $\{z_2, z_6\}$, $\{z_3, z_7\}$ и $\{z_4, z_8\}$.

Рассмотрим группу $\{z_1, z_5\}$ графически зависимых выходов дешифратора. Для пары выходов этой группы графически существенной вершиной является вершина 1 (см. рис. 5-4), содержащая элементы $\{1, 2\}$. Для определения зависимости выходов z_1 и z_5 проверим на тождественное равенство нулю выражения:

$$\frac{dz_1}{dy_2} \frac{dz_5}{dy_2};$$

$$z_1 = y_1 x_2;$$

$$z_5 = y_2 x_1;$$

$$\frac{dz_1}{dy_2} = 1 \cdot \bar{x}_2 \oplus 0 \cdot \bar{x}_1 = \bar{x}_2;$$

$$\frac{dz_5}{dy_2} = 1 \cdot x_2 \oplus 0 \cdot x_1 = x_2;$$

$$\frac{dz_1}{dy_2} \frac{dz_5}{dy_2} = \overline{x_7 x_8} = 0.$$

Отсюда следует, что выходы z_1 и z_5 независимы. Аналогичный анализ зависимости других пар графически зависимых выходов показывает, что все эти пары выходов независимы между собой. Это значит, что все выходы дешифратора независимы между собой, т. е. составляют одну группу независимых выходов — $\{z_1, z_2, z_3, z_4, z_5, z_6, z_7, z_8\}$. На этих выходах в процессе правильного функционирования дешифратора реализуется код 1 из 8. В этом коде $m=1$ и $n=8$. Поскольку $m=1$ — нечетное число, то типовая реализация строится в соответствии с рис. 5-20, а. На рис. 5-22 представлена типовая реализация самопроверяемой СВК для дешифратора, выполненная на двухвходовых элементах сложения по модулю 2. В качестве элемента a_1 подмножества A взят выход z_1 (можно было взять любой другой выход), остальные выходы отнесены к подмножеству B .

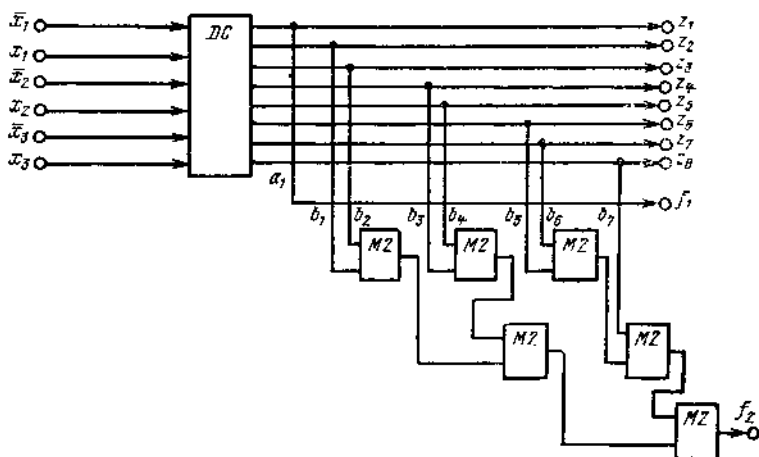


Рис. 5-22. Типовая реализация самопроверяемой СВК для дешифратора рис. 5-3.

Пример 5-14. Построим самопроверяемую СВК для регистра сдвига (см. рис. 4-10). При рассмотрении примера 5-11 было установлено, что выходы z_1, z_2, z_3, z_4 составляют одну группу независимых выходов. Нетрудно заметить, что в процессе правильного функционирования регистра на этих выходах реализуется код 2 из 4. Поскольку $k=2$ — четное число, то типовая реализация строится в соответствии с рис. 5-20, б. На рис. 5-23 представлена искомая реализация самопроверяемой СВК.

Здесь необходимо отметить следующее обстоятельство. Для кода 2 из 4 всего существует $C_4^2=6$ кодовых слов, содержащих точно две единицы и два нуля. В процессе правильного функционирования регистра на его выходах присутствуют только четыре кодовых сло-

ва — 1100, 0110, 0011, 1001. Тем не менее на каждый двухвходовой элемент сложения по модулю 2, с помощью которых реализуются функции f_1 и f_2 , поступают все возможные их входные наборы. Поэтому при правильно функционирующем регистре сдвига на вы-

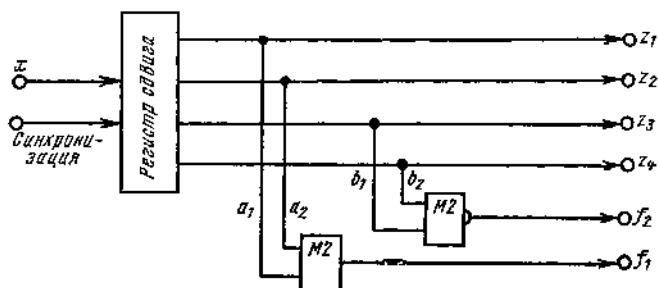


Рис. 5-23. Типовая реализация самопроверяемой СВК для регистра сдвига рис. 4-19.

ходе $f_1(f_2)$ будут обнаружены все возможные одиночные и кратные логические и константные неисправности схемы свертки, реализующей функцию $f_1(f_2)$.

5-6. ПОСТРОЕНИЕ САМОПРОВЕРЯЕМЫХ СВК, ОБЕСПЕЧИВАЮЩИХ ЧАСТИЧНУЮ ПРОВЕРЯЕМОСТЬ КОМБИНАЦИОННЫХ УСТРОЙСТВ

Комбинационное устройство называется частично проверяемым, если любая его одиночная константная неисправность (неисправность из заданного класса) обнаруживается СВК хотя бы на одном заданном наборе значений входных переменных $\{X\}_p$.

Характерной особенностью рассмотренных ранее методов построения самопроверяемых СВК являлось условие, по которому любое появление неправильных значений выходов контролируемого устройства немедленно фиксировалось схемой контроля. Это условие означает, что каждая логическая неисправность устройства обнаруживалась на каждом из тех входных наборов из $\{X\}_p$, которые для нее являются тестовыми. Ослабим это требование и ограничимся тем, чтобы каждая одиночная существенная константная неисправность обнаруживалась хотя бы на одном тестовом для нее входном наборе из $\{X\}_p$. При этом можно во многих случаях получить более простые реализации СВК. Платой за это упрощение является то, что не все неисправности будут обнаруживаться в момент их первого проявления на выходах контролируемого устройства.

Процедура построения СВК, обеспечивающей частичную проверяемость комбинационного устройства, состоит из следующих операций:

а) по заданной схеме устройства выделяются m ее максимальных одновыходных подсхем;

б) строятся m частичных проверяющих тестов для устройства; i -й ($i=1, 2, \dots, m$) частичный тест обнаруживает все неисправности (из заданного класса) i -й максимальной подсхемы при условии, что остальные максимальные подсхемы исправны;

в) выполняется m -кратное моделирование работы неисправного устройства; i -е ($i=1, 2, \dots, m$) моделирование выполняется на входных наборах i -го частичного теста, принадлежащих $\{X\}_p$, при включенном инверторе на выходе i -й максимальной подсхемы;

г) результаты моделирования заносятся в подтаблицы неправильных значений общей таблицы истинности, аналогичной табл. 5-1;

д) выполняется кодирование общей таблицы истинности значениями функций f_1, f_2 . Значения 00 и И в подтаблицах неправильных значений проставляются только в тех строках, для которых выполнялось моделирование, в остальных строках значения f_1, f_2 безразличны;

е) по полученной общей таблице истинности выполняется синтез схемы встроенного контроля, реализующей функции f_1 и f_2 .

Операция п. «а» выполняется по алгоритму А4-1, описанному в гл. 4.

Операция п. «б» по построению частичных проверяющих тестов может выполняться любым известным методом, в том числе методами, изложенными в [1-5, гл. 3]. В большинстве случаев для сложных устройств из-за ограниченных возможностей методов приходится ограничиваться классом одиночных константных неисправностей на входах и выходах элементов максимальных подсхем.

Вариантов частичных тестов для одной и той же максимальной подсхемы может быть много. Задача выбора лучшего из них (например, минимального по числу тестовых наборов, все или большинство которых принадлежат множеству $\{X\}_p$) весьма громоздка. Решение этой задачи не входит в наши цели здесь, а излагаемый метод обеспечивает построение СВК, сложность которой зависит от качества частичных тестов.

Заметим еще, что частичные тесты для максималь-

ных подсхем можно получить другим путем, а именно, построить сначала тест для устройства в целом, а затем выделить из него частичные тесты для максимальных подсхем. Выделение частичных тестов может проводиться либо в процесс построения теста для устройства, либо после построения при помощи моделирования устройства с неисправностями его максимальных подсхем. Наконец, укажем на возможность выделения частичных тестов для максимальных подсхем путем моделирования устройства с неисправностями на множестве $\{X\}_p$ заданных входных наборов.

Операция п. «в» предусматривает моделирование поведения неисправного устройства на наборах частичных тестов при инвертировании выходов максимальных подсхем. Ясно, что при таком моделировании будут обнаруживаться все те логические (а значит, и константные) неисправности, которые обнаруживаются построенными частичными тестами. Операция п. «в» и следующая за ней операция п. «г» могут быть исключены, если информация, необходимая для заполнения подтаблиц неправильных значений общей таблицы истинности, фиксируется при построении частичных тестов.

Пример 5-15. Построим самопроверяемую СВК, обеспечивающую частичную проверяемость комбинационного сумматора, схема которого и ее обобщенный граф представлены на рис. 5-16, а, б.

Имеются две максимальные подсхемы с выходами S_i и C_i . Частичный проверяющий тест, который обнаруживает неисправности подсхемы, реализующей функцию s_i , содержит следующие наборы значений входных переменных a_i, b_i, c_{i-1} : 001, 010, 101, ПО. На этой совокупности наборов каждый из двухвыходных элементов (1 и 4) сложения по модулю 2 проверяется на всех четырех наборах значений своих входов.

Совокупность входных наборов значений тех же переменных a_i, b_i, c_{i-1} , проверяющих подсхему, которая реализует функцию c_i , следующая: 001, 010, 011, 100, 101, 110.

Поскольку функционирование сумматора определено на всех 2^3 наборах значений входных переменных a_i, b_i, c_{i-1} , то найденные наборы проверяющих тестов заведомо содержатся в множестве $\{X_p\}$ заданных входных наборов.

Построим общую таблицу истинности для сумматора (табл. 5-10), в которой поместим значения, реализуемые на выходах S_i и C_i на множествах входных наборов полученных частичных проверяющих тестов при последовательной замене в схеме сумматора функций элементов 4 и 6 на инверсные функции.

Как уже говорилось, при кодировании общей таблицы истинности значениями функций f_1 и f_2 следует стремиться к тому, чтобы каждая из функций f_1 и f_2 зависела от меньшего числа переменных. При выполнении в табл. 5-10 варианте кодирования для функций f_1 (f_2) существенными являются переменные b_i, c_{i-1}, S_i (a_i, b_i, c_{i-1}, c_i), причем значения функции f_1 (f_2) совпадают со значениями

Таблица 5-10

a_i	b_i	c_{i-1}	s_i	c_i	f_1	f_2	Схема
0	0	0	0	0	1	0	Исправная
0	0	1	1	0	1	0	
0	1	0	1	0	1	0	
0	1	1	0	1	0	1	
1	0	0	1	0	1	0	
1	0	1	0	1	0	1	
1	1	0	0	1	0	1	
1	1	1	1	1	1	0	
0	0	1	0	0	0	0	С инвертором на выходе элемента 4
0	1	0	0	0	0	0	
1	0	1	1	1	1	1	
1	1	0	1	1	1	1	
0	0	1	1	1	1	1	С инвертором на выходе элемента 6
0	1	0	1	1	1	1	
0	1	1	0	0	0	0	
1	0	0	1	1	1	1	
1	0	1	0	0	0	0	
1	1	0	0	0	0	0	

переменных s_i (c_i) на всех наборах, за исключением набора 00000 (11111) ($a_i b_i c_{i-1} s_i c_i$). Функции f_1 и f_2 имеют следующие реализации: $f_1 = (b_i \vee c_{i-1}) \vee s_i$, $f_2 = (a_i b_i c_{i-1}) c_i$. На рис. 5-24 представлена СВК сумматора, которая является самопроверяемой, так как функции f_1 и f_2 имеют раздельные реализации в виде древовидных схем без разветвлений на входах, и для каждой существенной входной переменной функции f_1 и f_2 среди наборов, входящих в подтаблицу исправных значений общей таблицы истинности, существуют пары проверяющих наборов. Комбинационный сумматор является частично проверяемым вследствие того, что все его одиночные константные неисправности гарантированно обнаруживаются СВК только на входных наборах, частичных проверяющих тестов сумматора. Выполнив анализ схем, реализующих функции f_1 и f_2 , при инвертировании выходов s_i и c_i , можно установить, что неисправности сумматора, проявляющиеся на выходе s_i , не обнаруживаются на входных наборах

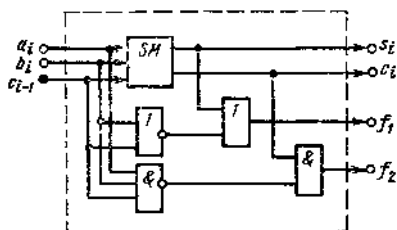


Рис. 5-24. Самопроверяемая СВК для частично проверяемого сумматора рис. 5-16,а.

000 и 100 (a, b, c_{i-1}), a проявляющиеся на выходе c_i — наборе 111 (a, b, c_{i-1}). Причина этого состоит в том, что указанные наборы не вошли в соответствующие частичные тесты, и случайно оказалось, что на этих наборах значения f_1, f_2 равны 10. С другой стороны, также случайно получилось, что на не вошедших в частичный тест для подсхемы с выходом $s_i(c_i)$ наборах 011 и 111 (наборе 000) неисправности, проявляющиеся на выходе $s_i(c_i)$, обнаруживаются, так как на этих наборах значения f_1, f_2 равны соответственно 11 и 00 (11).

Пример 5-16. Построим самопроверяемую СВК для дешифратора, структурная схема и обобщенный граф которого изображены на рис 5-3 и 5-4. Согласно обобщенному графу дешифратора имеется 12 максимальных одновыходных подсхем. Пары элементов, входящих в подсхемы, а также соответствующие подсхемам частичные проверяющие тесты дешифратора для переменных (x_1, x_2, x_3) следующие:

1. {1, 2} — 000, 100, 010; \bar{x}_1
2. {3, 4} — 100, 110, 000; \bar{x}_2
3. {5, 6} — 010, 110, 000; \bar{x}_3
4. {7, 8} — 110, 010, 100; x_1
5. {9, 10} — 000, 100, 001; x_2
6. {11, 12} — 100, 000, 101; x_3
7. {13, 14} — 010, 110, 011; \bar{x}_1
8. {15, 16} — 110, 010, 111; \bar{x}_2
9. {17, 18} — 001, 101, 000; \bar{x}_3
10. {19, 20} — 101, 001, 100; x_1
11. {21, 22} — 011, 111, 010; x_2
12. {23, 24} — 111, 011, 110; x_3

В общей таблице истинности дешифратора (см. табл. 5-3) в подтаблицах неправильных значений определяем значения функций f'_1 и f'_2 только для входных наборов, включенных в соответствующие частичные тесты. Для экономии места в табл. 5-3 представлены только две подтаблицы неправильных значений — для максимальных подсхем 1 и 12.

С целью получения простой реализации СВК при кодировании общей таблицы истинности значениями функций f'_1 и f'_2 разобьем все выходы дешифратора на две группы: в первую группу включим выходы, для которых входные наборы дешифратора содержат четное число единиц, во вторую — нечетное число единиц. Все выходы дешифратора при этом разбиваются на две группы: $\{z_1, z_4, z_6, z_7\}$ и $\{z_2, z_3, z_5, z_8\}$.

Закодируем общую таблицу истинности (табл. 5-3) значениями функций f'_1 и f'_2 следующим образом: а) в подтаблице правильных значений в строках, содержащих единицы в столбцах, принадлежащих первой (второй) группе выходов, помещаем значения $f'_1=1, f'_2=0$ ($f'_1=0, f'_2=1$); б) в подтаблицах неправильных значений в строках, содержащих все нули, помещаем значения $f'_1=0, f'_2=0$, а в строках, содержащих две единицы, — значения $f'_1=1, f'_2=1$.

При таком кодировании получаем $f'_1 = z_1 \vee z_4 \vee z_6 \vee z_7$ и $f'_2 = z_2 \vee z_3 \vee z_5 \vee z_8$. Реализация самопроверяемой схемы встроенного контроля, обеспечивающей частичную проверяемость дешифратора, представлена на рис. 5-25.

На каждую из схем ИЛИ, с помощью которых реализуются функции f'_1 и f'_2 , в процессе правильного функционирования дешифратора поступают наборы минимального теста — 0000, 1000, 0100, 0010, 0001 ($z_1 z_4 z_6 z_7$ или $z_2 z_3 z_5 z_8$). Поэтому на выходе f'_1 (f'_2) будут обнаружены все одиночные и кратные константные неисправности схемы, реализующей функцию f'_1 (f'_2).

Сравнивая рис. 5-2 и 5-24 (а также рис. 5-5 и 5-25), на которых представлены самопроверяемые СВК, обеспечивающие полную и соответственно частичную проверяемость сумматора (дешифратора), легко заметить различие схем по сложности реализации. Более простые схемы частичной проверяемости получают за счет увеличения числа безразличных значений функций f_1 и f_2 (f'_1 и f'_2) при кодировании подтаблиц неправильных значений общих таблиц истинности.

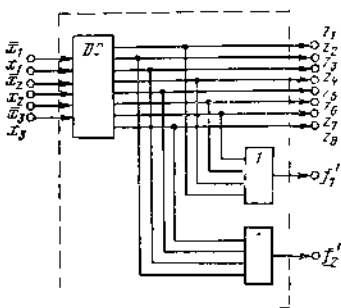


Рис. 5-25 Самопроверяемая СВК для частично проверяемого дешифратора рис. 5-3.

ЗАМЕЧАНИЯ ПО БИБЛИОГРАФИИ

В [5-2, 5-5] введены понятия полностью самопроверяемых дискретных устройств и схем встроенного контроля. Следует отметить, что понятие полностью самопроверяемого ДУ определяет класс избыточных устройств, в которых каждая неисправность обнаруживается на множестве рабочих входных наборов или последовательностей (выполняется свойство самотестируемости) и каждая неисправность проявляется на выходах только в виде некодовых слов (выполняется свойство защищенности от неисправностей). При этом подмножество некодовых слов не должно пересекаться с подмножеством кодовых слов, реализуемых исправным устройством.

Понятие полностью проверяемого ДУ [4-1, 4-10, 5-13] определяет устройства, в которых каждая неисправность обнаруживается СВК в момент ее первого проявления на выходах устройства в виде неправильного слова. При этом подмножества правильных и неправильных выходных слов ДУ могут пересекаться. Блок,

содержащий полностью проверяемое ДУ и самопроверяемую СВК является полностью самопроверяемым.

Изложенные в § 5-2 методы построения самопроверяемых СВК для кодов m из n предложены в [5-2, 5-5]. Схема сжатия на рис. 5-11,*а* описана в [5-5, 5-6]. Схема сжатия на рис. 5-11,*б* предложена В. А. Вашиным.

Исследование условий построения самопроверяемых СВК для комбинационных устройств проведено в [5-13]. Структурные методы построения самопроверяемых СВК по модулю 2 (самопроверяемых СВК типовой реализации) предложены для комбинационных устройств в [5-13] и для автоматов с памятью в [5-1].

Описанию процедур построения полностью самопроверяемых последовательностных устройств посвящены работы [5-4, 5-10]. Методы построения полностью самопроверяемых последовательностных устройств с использованием кодов k из n при кодировании внутренних состояний автомата рассмотрены в [5-8, 5-9, 5-11].

Понятие частично проверяемого ДУ и понятие частично самопроверяемого ДУ введены соответственно в [5-13, 5-16]. В [5-16] частичная самопроверяемость используется для организации контроля арифметических и логических операций.

В [5-3] рассматривается процедура построения самопроверяемых схем встроенного контроля для разделимых кодов. В работе [5-12] предложен метод построения самопроверяемых устройств на основе альтернативной логики, в которой последовательно во времени реализуется как сама функция, так и ее самодвойственная функция.

Построенная по изложенному в § 5-6 методу самопроверяемая СВК, обеспечивающая частичную проверяемость дешифратора, совпадает со схемой, приведенной в [5-7].

В [5-15] на примере комбинационных схем рассмотрена процедура построения одновыходных самопроверяемых схем контроля. Одновыходные СВК гарантируют обнаружение всех одиночных неисправностей в ДУ и СВК и при выполнении найденных в [5-15] условий всех кратных константных неисправностей на выходах ДУ и в самом СВК.