

ЗАМЕЧАНИЯ ПО КОНТРОЛЮ И ДИАГНОСТИКЕ
 ОДНОРОДНОЙ ВЫЧИСЛИТЕЛЬНОЙ СРЕДЫ

И.В. Коган

В работе рассматривается методика построения проверяющих и диагностирующих тестов на кратные константные неисправности для однородных вычислительных сред (ВС).

Введем ряд обозначений. Под сквозными входными наборами будем понимать такие наборы входных переменных, для которых:

- в исправной среде изменение значений переменных (для которых они составлены) в наборе распространяется до выхода среды по определенному пути \mathcal{J} ;

- при одиночной неисправности пути \mathcal{J} значение пути от точки неисправности до выхода равно константе.

Если обозначить (для плоской среды) входы элемента через x_{ij}^r, y_{ij}^s , где $r = 1, 2, \dots, k$ - число входов элемента среды по координате i ($i = 1, 2, \dots, m$), а $s = 1, 2, \dots, l$ - число отрок по координате j ($j = 1, 2, \dots, n$) и выходы элемента через $x_{(i+1)j}^r, y_{ij}^{(s+1)}$, то для сквозных наборов должно выполняться, по крайней мере, одно из следующих соотношений:

$$\begin{aligned} x_{ij}^r &= x_{(i+1)j}^r, \\ y_{ij}^s &= y_{ij}^{(s+1)}, \\ x_{ij}^r &= f(y_{pj}^s) \quad \text{для } i > p, \\ y_{ij}^s &= \varphi(x_{iq}^r) \quad \text{для } j > q. \end{aligned} \tag{I}$$

Определим проверяющее покрытие путей среды S , которое обладает следующими свойствами:

- если Q - путь в ВС, то точки пути Q содержатся в S ;
- если \mathcal{J} - путь, используемый множеством сквозных наборов, то \mathcal{J} содержится в S ;
- если \mathcal{R} - путь от входов к выходам и не используемый сквозными наборами, то \mathcal{R} не содержится в S .

Порядок контроля вычислительной среды следующий. Сначала строятся проверяющий и диагностирующий тесты для элемента ВС. Затем для пропуска тестов элементов строятся сквозные наборы, пути которых образуют проверяющее покрытие путей. Изложение методики проведено на примере построения тестов для ВС, описанной в [1]. На рис.1 приведена схема элемента ВС, а на рис.2 - схема соединения элементов в среде.

Построим тесты для элемента среды. Формулы, эквивалентные схеме (ФЭС) [2], для элемента имеют вид:

$$x' = (x_1 \vee (v z_1)_5)_7, \quad (2)$$

$$z' = (((\bar{y}_1)_4 z_2)_6 \vee |_5)_8, \quad (3)$$

$$y' = y_2, \quad (4)$$

$$u' = u_1, \quad (5)$$

$$v_{y+1} = ((u_2 \vee x_{2y})_2 \vee \overline{(u_{3y} |_4)_3})_1, \quad (6)$$

где символ $|_i^j$ заменяет повторную запись выражения в скобках, $(f)_i$ - ветви схемы, j - номер ветви (верхний индекс $j = 0$ в скобке $(f)_i$ опущен). Нижние индексы обозначают номер вхождения переменной в ФЭС, а в формуле (6) индекс v обозначает момент времени.

Проверяющий тест на кратные константные неисправности представляет (см. [2] и [3]) совокупность парных наборов для букв и ветвей ФЭС. Чтобы воспользоваться этими результатами для устройства с памятью, распространим понятие парных наборов на формулы (2) - (6). Поскольку v_{y+1} возрастает [2] по переменной x (так как $v_{y+1} \geq x_y$ при $u = 1$), как и формула (2), то для проверки x_1 можно построить условно-парные наборы с блокирующей буквой z_1 , которая проверяется парными наборами. Если же зафиксировать последовательность пропуска наборов во времени, то для проверки буквы x_1 можно построить парные наборы. Для это-

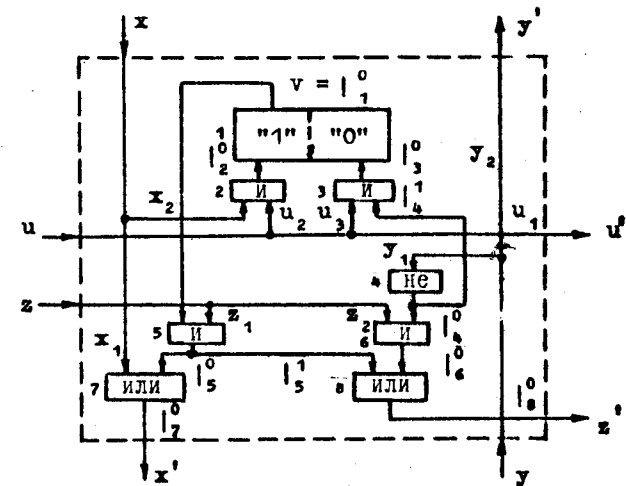


Рис. 1

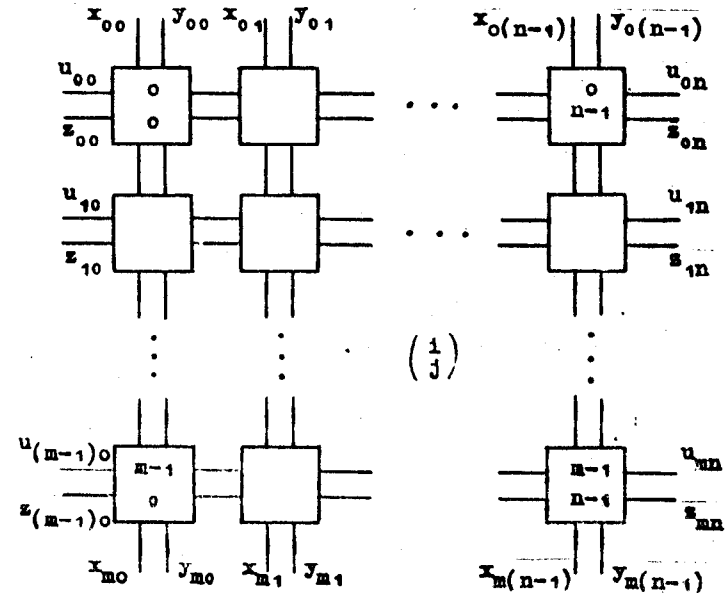


Рис. 2

го необходимо (поскольку формула (2) возрастает по буквам x , и v) сначала пропустить набор с $x_1 = 1$, а затем с $x_1 = 0$. Кроме того, необходимо проверить способность триггера принимать оба состояния, для чего нужно построить парные наборы для переменной v (изменение значения переменной v производится переключением триггера). Переключения триггера $0 \rightarrow 1$ и $1 \rightarrow 0$ проверяются парными наборами (ветвей l_2^0 и l_3^0 [2]).

В табл. 1 приведены полные наборы для букв и ветвей ФЭС (2) - (6). В графе I3 символом \checkmark указывается, что порядок пропуска во времени зафиксирован в направлении стрелки. Символом * отмечены значения, не проверяемые в данном такте, и символом ∇ - значения, которые не задаются. В строках 8 и 9 указаны переключающие и проверяющие переключение наборы. Символом + указаны произвольно задаваемые значения, фиксация которых может быть использована для минимизации теста.

Таблица 1

№ пп	Проверяемая буква (ветвь)	Входы					Выходы					Примечание	Вес наборов
		x	y	z	u	v	x'	y'	z'	u'	v _{v+1}		
I	2	3	4	5	6	7	8	9	10	11	12	I3	I4
1	$x_1^0 _5^1 _5^0 _7^1 _8^0$	0	1	0	0	1	0	1	0	0	1		1
		0	1	1	0	1	1	1	0	0	1		
2	$y_1 y_2 x_1 _7^0 _6^0 _8^0$	+	1	1	0	0	+	1	0	0	0		2
		+	0	1	0	0	+	0	1	0	0		
3	$z_2 _6^0 _8^0$	+	0	0	0	0	+	0	0	0	0		4
		+	0	1	0	0	+	0	1	0	0		
4	$u_2 _1^0 _2^0 _5^0 _7^1 _8^0$	1	1	+	0	0	1	1	0	0	0	0 → 1	8
		1	1	+	1	0	1	1	0	1	1		
5	$x_2 _1^0 _2^0 _5^0 _7^1 _8^0$	0	1	+	1	0	0	1	0	0	0		16
		1	1	+	1	0	1	1	0	1	1	0 → 1	
6	$u_3 _4^0 _3^0 _5^0 _7^0$	0	1	1	1	1	1	1	1	1	1	1 → 0	32
		0	0	1	1	1	1	0	1	1	0		
7	$l_4^1 _1^0 _5^0 _7^0$	0	0	1	0	1	1	0	1	0	1	1 → 0	64
		0	0	1	1	1	1	0	1	1	0		
8	v	1	1	1	1	∇	1	1	∇	1	1	0 → 1	
		0	+	1	0	1	1	+	1	0	1	v = 1	
9	v	0	0	1	1	∇	0	0	1	1	0	1 → 0	
		0	+	1	0	0	0	+	+	0	0	v = 0	

ны значения, не проверяемые в данном такте, и символом ∇ - значения, которые не задаются. В строках 8 и 9 указаны переключающие и проверяющие переключение наборы. Символом + указаны произвольно задаваемые значения, фиксация которых может быть использована для минимизации теста.

На основе табл. 1 построен проверяющий тест ТП [4], который приведен в табл. 2. Всего в тест входит 12 наборов.

Таблица 2

№ пп	Входы					Выходы				
	x	y	z	u	v	x'	y'	z'	u'	v _{v+1}
1	0	0	0	1	1	0	0	0	1	0
2	0	0	0	0	0	0	0	0	0	0
3	1	1	1	0	0	1	1	0	0	0
4	0	1	1	1	0	0	1	0	0	0
5	0	0	1	0	0	0	0	1	0	0
6	1	1	0	1	0	1	1	0	1	1
7	0	1	0	0	1	0	1	0	0	1
8	0	1	1	1	1	1	1	1	1	1
9	0	0	1	0	1	1	0	1	0	1
10	0	1	1	0	1	1	1	1	0	1
11	0	0	0	1	1	0	0	0	1	0
12	0	0	1	0	0	0	0	1	0	0

Диагностирующий тест будем строить по методу, изложенному в [5]. Построим таблицу весов (табл. 3). Учитывая, что схема имеет 2 выхода, символом + обозначена существенность наборов для неисправности по выходу x' и символом * - по выходу z' . Диагностика неисправностей на выходах y' и u' тривиальна и

не рассматривается. Как мы видим, веса неисправностей (сумма весов пар наборов, которые не проходят при данной неисправности) различны, за исключением неразличимых неисправностей. Следовательно, тест ТП, приведенный в табл. 2, является одновременно диагностирующим тестом на кратные неисправности. Для определения наиболее вероятного списка подозреваемых одиночных неисправностей следует построить граф покрытий [5].

Перейдем к построению проверяющего теста для ВС. Определим наборы, проверяющие пути и состояния элементов в среде. Для этого составим табл. 4, в которой строки I - 8 перечисляют подмножество путей, покрывающих все пути в среде. Наборы, проверяющие эти пути, приведены в графе "Входы", где символом + обозначены элементы, задаваемые произвольно, что можно использо-

Таблица 3

Номер пары	Вес пары	Неисправности													0	16	25	123	31																							
		x_1	x_2	z_1	z_2	u_2	u_3	$ _{1'}^0(=0)$	$ _{1'}^0(=1)$	$ _{2'}^0$	$ _{3'}^0$	$ _{4'}^0$	$ _{4'}^1$	$ _{5'}^0$						$ _{5'}^1$	$ _{6'}^0$	$ _{6'}^1$	$ _{7'}^0$	$ _{7'}^1$																		
1	1	+																																								
2	2																																									
3	4																																									
4	8																																									
5	16																																									
6	32																																									
7	64																																									
Вес неисправн.		2	16	1	-	8	32	96	24	24	96	24	24	24	96	64	121	-	-	-	6	-	25	-	123	-	-	6	-	2	-	24	24	-	-	31	-	-				

Таблица 4

№ пп	Вид проверки	Входы
1	$x_{0j} - x_{mj}$	$\begin{cases} 0 & + & + & 0 & 0 \\ 1 & + & + & 0 & 0 \end{cases}$
2	$z_{0j} - x_{mj}$	$\begin{cases} 0 & + & 0 & 0 & 1 \\ 0 & + & 1 & 0 & 1 \end{cases}$
3	$z_{10} \vee z_{in}$	$\begin{cases} + & 1 & 0 & 0 & 1 \\ + & 1 & 1 & 0 & 1 \end{cases}$
4	$z_{10} \wedge z_{in}$	$\begin{cases} + & 0 & 0 & 0 & 0 \\ + & 0 & 1 & 0 & 0 \end{cases}$
5	$y_{0j} - z_{in}$	$\begin{cases} + & 0 & 1 & 0 & 0 \\ + & 1 & 1 & 0 & 0 \end{cases}$
6	$v_{0j} - x_{mj}$	$\begin{cases} 0 & + & 1 & 0 & 0 \\ + & 0 & + & + & + \end{cases}$
7	$y_{0j} - y_{mj}$	$\begin{cases} + & 1 & + & + & + \\ + & + & + & 1 & + \end{cases}$
8	$u_{10} - u_{in}$	$\begin{cases} + & + & + & 0 & + \\ + & + & + & 1 & + \end{cases}$
9	$v = 0$	$\begin{cases} 0 & + & 1 & 0 & 0 \\ + & 1 & 1 & 0 & 1 \end{cases}$
10	$v = 1$	$\begin{cases} 0 & 0 & 0 & 1 & + \\ 1 & 1 & + & 1 & + \end{cases}$
11	$v_{1j} \rightarrow 0$	$\begin{cases} 0 & 0 & 0 & 1 & + \\ 1 & 1 & + & 0 & 0 \end{cases}$
12	$v_{1j} \rightarrow 1$	$\begin{cases} 1 & 1 & + & 1 & + \\ 0 & 1 & + & 1 & 0 \end{cases}$
13	$u_{21j} \equiv 1$	$\begin{cases} 1 & 1 & + & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{cases}$
14	$x_{21j} \equiv 1$	$\begin{cases} 0 & 1 & + & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{cases}$
15	$u_{31j} \equiv 1$	$\begin{cases} 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \end{cases}$
16	$ _{41j} \equiv 1$	$\begin{cases} 0 & 1 & 0 & 1 & 1 \end{cases}$

вать для минимизации теста. Каждый набор необходимо понимать как задание значений $x_{0j}, y_{0j}, z_{10}, u_{10}, v_{1j} (j = 0, 1, \dots, n-1; i = 0, 1, \dots, m-1)$ такими, как указано в наборе. Символ $\alpha_{\rho i}$ следует понимать так, что $\alpha_{\rho i} = 1$ при $\rho = i$ и $\alpha_{\rho i} = ?$ при $\rho \neq i$ ($\rho = 0, 1, \dots, m-1$). Таким образом, набор, включающий символ $\alpha_{\rho i}$, определяет совокупность из m наборов. Аналогично набор с $\alpha_{\rho \rho}$ - это совокупность из n наборов. Строки 9-12 содержат наборы, переключающие триггеры и проверяющие их состояния. Из наборов в строках 1-12 табл. 4 можно получить все наборы проверяющего теста (согласно табл. 1 и 2) для произвольного элемента среды, кроме, может быть, тех частей наборов 3, 4, 8 и 9, которые служат для проверки неправильного переключения триггеров. Эти наборы записаны в строки 13-16 табл. 4. После пропуска наборов 13, 14 необходима проверка $v_{1j} = 0$, а после наборов 15, 16 - $v_{1j} = 1$. С помощью табл. 1, 2 и 4 построен проверяющий тест, который сведен в табл. 5. Всего в тест входит:

$$L = n + m + 12 \quad (7)$$

наборов, пропуск которых нужно проводить в порядке их размещения в табл. 5. В табл. 5 отмечены лишь проверяемые выходы, а значения остальных заменены символом +. Построение диагностирующего теста для среды не проводится вследствие громоздкости этой процедуры.

Таблица 5

№№ ПП	ВХОДЫ						ВЫХОДЫ						Примечание
	X _{0j}	Y _{0j}	Z ₁₀	U ₁₀	V _{1j}	X _{mj}	Y _{mj}	Z _{1n}	U _{1n}	V _{n+1}			
1	0	0	0	1	1	+	+	+	+	0			
2	0	0	0	0	0	0	0	0	0	0			
3	0	0	1	0	0	0	0	0	0	0			
4	1	1	0	1	0	1	0	0	0	0			
5	0	0	1	0	0	0	1	0	0	0			
6	0	0	1	0	0	+	+	+	1	1			
7	1	1	0	1	0	1	0	0	0	1			
8	0	0	0	0	1	1	0	0	0	1			
9	0	0	1	0	1	1	z ^{po}	0	0	1			
10	0	0	0	1	1	0	0	1	1	1			
11	0	0	1	0	1	0	0	0	0	1			
12	0	0	1	0	1	+	+	+	+	0			
13	0	0	0	1	0	+	+	+	+	0			
14	0	0	1	0	0	0	1	0	0	0			

$$Y_{0q} = \begin{cases} 1 & \text{при } j=q \\ 0 & \text{при } j \neq q \end{cases}$$

$$z_{po} = \begin{cases} 1 & \text{при } i=p \\ 0 & \text{при } i \neq p \end{cases} \quad (p = 0, 1, \dots, m-1)$$

В заключение можно отметить, что проведенный анализ показывает возможность построения тестов для однородных вычислительных сред с элементами памяти при достаточно хорошей оценке их длины.

Л и т е р а т у р а

1. ФЕТ Я.И. Вычислительные среды с простыми ячейками. I. Настоящий сборник, стр. 48-66.
2. БЛАЖИВСКАЯ Г.В., КОГАН И.В. Некоторые вопросы построения проверяющих тестов для многовыходных комбинационных устройств. II Всесоюзное совещание по теории релейных устройств и конечных автоматов. Тезисы докладов. Рига, 28 сентября - 1 октября 1971.
3. КОГАН И.В. Построение тестов для одноканальных автоматов. Материалы научных семинаров по теоретическим и прикладным вопросам кибернетики. Киев, 1963.
4. КОГАН И.В. Построение тестов для контактных схем. Материалы научных семинаров по теоретическим и прикладным вопросам кибернетики. Киев, 1963.
5. КОГАН И.В. Диагностика кратных неисправностей в комбинационных устройствах. Информационные методы в системах управления, измерений и контроля (доклады II Всесоюзного семинара), Владивосток, 1972, том. II.

Поступила в ред.-изд.отд.
29 июня 1972 г.