

Т. ЛОХУАРУ, М. ПАЛЛЬ, Р. УБАР

АВТОМАТИЧЕСКИЙ СИНТЕЗ ТЕСТОВ ДЛЯ ДИАГНОСТИКИ ЦИФРОВЫХ УСТРОЙСТВ

(Представил Н. Алумяэ)

1. Введение

Одной из тенденций в производстве цифровых устройств (ЦУ) является все большее применение элементов с высокой степенью интеграции. Происходит резкое усложнение функций, выполняемых сменными интегральными элементами. В результате этого увеличиваются трудности, связанные с диагностикой неисправностей в ЦУ. Раньше эти трудности возникали главным образом при поиске неисправностей в устройстве или системе, теперь же локализация места неисправности даже в пределах одного конструктивного блока, например, платы ЭВМ, представляет собой достаточно трудоемкую задачу. В настоящее время уровень сложности устройств и их специфика часто опережают возможности существующих методов технической диагностики. Применительно к отдельным элементам (комбинационным и схемам с памятью) эти методы оправдывают себя, но в случае применения элементов с повышенной степенью интеграции (СИС и БИС) приводят к существенному увеличению размерности решаемых задач. К тому же место неисправности в ЦУ обычно требуется указать с точностью не до логического, а до сменного элемента, например, интегральной схемы (ИС, СИС или БИС).

Предлагаемый в данной работе подход заключается в организации построения тестов на уровне не элементов, а модулей. Под модулями здесь понимаются сменные элементы (подсхемы) устройства, с точностью до которых желательно производить диагноз.

Задача синтеза тестов разделяется на два этапа: синтез тестов для модулей и синтез тестов для всего устройства. Первый этап здесь не рассматривается, так как из-за небольшой размерности модулей достаточно успешно оправдывают себя известные методы. Кроме того, для ряда специфических модулей (счетчиков, сдвигающих регистров, микропрограммных автоматов и т. д.) часто более естественно и легко разработать тесты для их функциональной проверки, так как локализация неисправностей внутри модуля не нужна. Тесты для модулей будем называть далее элементарными тестами.

Синтез теста для всего устройства заключается в составлении цепочки тестов отдельных модулей при условии наличия активизированных одномерных или многомерных путей от проверяемого модуля к первичным выходам устройства.

Для активизации путей через определенные модули могут быть разработаны соответствующие стандартные процедуры. В общем случае требуется некоторое описание функций модулей, позволяющее формально провести активизацию путей. Недостатком известных функциональных моделей цифровых схем в виде булевых функций является либо большой их объем (при нормальной форме), либо сложность обработки модели (при скобочной форме).

С целью экономии памяти ЭВМ и увеличения производительности системы автоматического синтеза тестов для ЦУ нами разработан метод описания функций проверяемых объектов в виде иерархической системы альтернативных графов (АГ) [1,2] и сформулированы задачи активизации путей на этой системе.

2. Задание объекта диагностики

Рассмотрим объект диагностики (ЦУ) в виде некоторой системы модулей $A = \{A^k\}$, имеющей общее функциональное описание $\Phi(A) = \{Z, F\}$. Здесь F — множество булевых функций на множестве переменных Z :

$$z_k = f_k(Z_k), \quad z_k \in Z, \quad f_k \in F,$$

где $Z_k \subseteq Z \setminus z_k$. Для каждого модуля $A^k \in A$ выделяется из $\Phi(A)$ свое описание $\Phi(A^k) = \{Z^k, F^k\}$, где $Z^k \subset Z, F^k \subset F$.

Множество переменных Z разбивается на подмножества входных X , внутренних Q и выходных Y переменных объекта. Множество функций F , в свою очередь, разбивается на подмножества $F_{\text{вых}}$ и $F_{\text{вн}}$ — выходных функций

$$y_k = f_k(Z_k), \quad y_k \in Y, \quad f_k \in F_{\text{вых}}$$

и внутренних функций

$$q_k = f_k(Z_k), \quad q_k \in Q, \quad f_k \in F_{\text{вн}}.$$

Здесь $Z_k = X_k \cup Q_k$, где $X_k \subseteq X$ и $Q_k \subseteq Q$. При этом допускаются частные случаи $Q_k = \emptyset$ или $X_k = \emptyset$. Для последовательных цифровых схем из Z выделяется подмножество переменных состояния памяти $Z_{\text{п}} \subseteq Q \cup Y$, а из F — подмножество функций перехода $F_{\text{п}} \subseteq F$. Для комбинационных схем $Z_{\text{п}} = \emptyset$ и $F_{\text{п}} = \emptyset$.

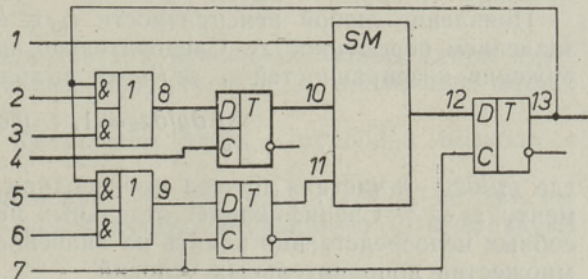
Множество $F_{\text{вых}}$ определяется непосредственно структурными свойствами схемы — множеством выходных переменных Y . Множество $F_{\text{вн}}$ определяется соответствующей структурной декомпозицией исходной схемы и в этом смысле может задаваться по-разному. В предельных случаях может иметь место или чисто функциональное описание объекта (если $F_{\text{вн}} = \emptyset$), или структурно-аналитическое (если $F_{\text{вн}}$ содержит функции структурных элементов, модулей объекта). Уровень декомпозиции схемы зависит от требуемой точности диагноза.

Для каждого модуля A^k может быть также проведена локальная декомпозиция его функций исходя из требований эффективного моделирования: $F^k = F^k_{\text{вых}} \cup F^k_{\text{вн}}$, где $|F^k_{\text{вн}}| \geq 0$. Для простоты изложения далее предполагается, что все модули A^k имеют один выход z_k , т. е. $|F^k_{\text{вых}}| = 1$ для всех $k: A^k \in A$. Обобщение результатов на многовыходные модули не представляет труда. Например, при тривиальном подходе многовыход-

ные модули можно рассматривать состоящими из нескольких фиктивных одновыходных модулей.

В качестве примера рассмотрим цифровую

Рис. 1.



схему (рис. 1), описываемую множествами переменных

$$X = \{z_1, \dots, z_7\}, \quad Q = \{z_8, \dots, z_{12}\}, \quad Y = \{z_{13}\}, \quad Z_{\Pi} = \{z_{10}, z_{11}, z_{13}\}$$

и множествами функций

$$F_{\text{вн}} = \{f_8, \dots, f_{12}\}, \quad F_{\text{вых}} = \{f_{13}\}, \quad F_{\Pi} = \{f_{10}, f_{11}, f_{13}\},$$

где

$$f_8 : z_8 = z_3 \vee z_2 z_{13},$$

$$f_9 : z_9 = z_6 \vee z_5 z_{13},$$

$$f_{12} : z_{12} = z_1 z_{10} z_{11} \vee z_1 \bar{z}_{10} \bar{z}_{11} \vee \bar{z}_1 z_{10} \bar{z}_{11} \vee \bar{z}_1 \bar{z}_{10} z_{11}$$

— модули с комбинационными схемами, а

$$f_{10} : z_{10}(t) = z_4(t) z_8(t) \vee \bar{z}_4(t) z_{10}(t-1),$$

$$f_{11} : z_{11}(t) = z_7(t) z_9(t) \vee \bar{z}_7(t) z_{11}(t-1),$$

$$f_{13} : z_{13}(t) = \bar{z}_7(t) z_{12}(t) \vee z_7(t) z_{13}(t-1)$$

— модули с последовательностными схемами.

3. Постановка задачи

Обозначим множество всех логически описываемых одиночных неисправностей в исследуемом объекте через R . Требования логического описания неисправностей $r \in R$ непосредственно связаны с необходимой степенью точности диагноза, а следовательно, с декомпозицией объекта на модули.

Выделим для каждого модуля A^k подмножество переменных $\hat{Z}_k^F \subseteq Z^k$, соответствующих его входам и выходам. Объединение всех \hat{Z}_k^F , $\hat{Z} = \bigcup_k \hat{Z}_k^F$, $k: A^k \in A$, дает множество переменных, каждая из которых представляет собой некоторую точку в исходной схеме соединений на уровне сменных элементов и, следовательно, может служить базой при логическом описании физических неисправностей $r \in R$, проявляющихся в определенной окрестности этой точки.

Пусть $R_k = R_k^F \cup R_k^M \subset R$ — подмножество неисправностей, местонахождение которых определено переменной $z_k \in \hat{Z}$. Здесь $R_k^F = \{r_{kj}^F\}$, $k: z_k \in \hat{Z} \setminus X$, — подмножество неисправностей в самом модуле A^k (напр., произвольное изменение его функции), а $R_k^M = \{r_{kj}^M\}$, $k: z_k \in \hat{Z}$, — подмножество неисправностей монтажа в сети связей у данной точки (напр., короткие замыкания (КЗ), перепутывание проводов (ПП) и т. д.). Объединяя все определенные таким образом подмножества R_k , $k: z_k \in \hat{Z}$, получим полное множество неисправностей, возможных в заданной схеме.

Появление любой неисправности $r_{kj} \in R_k$ сопровождается ложным значением переменной z_k . Следовательно, необходимым условием обнаружения неисправностей $r_{kj} \in R_k$ на выходе объекта является

$$\bigvee_y \frac{\partial y}{\partial z_k} = 1, \quad y \in Y, \quad (1)$$

где $\frac{\partial y}{\partial z_k}$ — частная булева производная функции $y = f(z)$ по аргументу $z_k \in Z$. Спецификацию отдельных неисправностей $r_{kj} \in R_k$, способных непосредственно влиять на значение z_k , можно задавать в виде множества дополнительных условий

$$W_{hj}(\hat{Z}_h) = 1, \quad j: r_{hj} \in R_h. \quad (2)$$

Здесь $\hat{Z}_h = \hat{Z}_h^F \cup \hat{Z}_h^M \subset \hat{Z}$, где \hat{Z}_h^M — подмножество переменных, необходимых для логического описания неисправностей монтажа $r \in R_h^M$ в точке, соответствующей z_h . Выполнение (2) для конкретного j требуется для того, чтобы неисправность r_{hj} была локально активизирована, т. е. чтобы ее наличие изменило значение z_h .

Так, например, КЗ между проводами z^h , $z_l \in \hat{Z}_h^M$ можно описать условием

$$W(\hat{Z}_h^M) = z_h \bar{z}_l = 1 \quad (W(\hat{Z}_h^M) = \bar{z}_h z_l = 1),$$

а ПП между теми же проводами — условием

$$W(\hat{Z}_h^M) = z_h \oplus z_l = 1.$$

Для описания неисправностей внутри модуля воспользуемся функциональной моделью [3], предполагая, что для всех модулей $A^k \in A$ элементарные тесты известны. Обозначим через $E_{hj} = \{e_{hj}/h : z_h \in \hat{Z}^h\}$, $e_{hj} \in \{0, 1\}$, j -й входной-выходной набор, а через E_h — упорядоченное множество всех таких наборов E_{hj} , $j = 1, 2, \dots$, составляющих элементарный тест модуля A^h . Тогда ложное значение $z_h \in \hat{Z}_h^F$, $z_h \neq e_{hj}$, при проверке модуля A^h набором $E_{hj} \in E_h$ является признаком наличия в A^h некоторой «функциональной неисправности» $r_{hj}^F \in R_h^F$. Спецификацию последней зададим в виде условия

$$W_{hj}(\hat{Z}_h^F) = \bigwedge_h \overline{(z_h \oplus e_{hj})} = 1, \quad h: z_h \in \hat{Z}_h^F. \quad (3)$$

Итак, объединяя условия (1) и (2) и вводя время (текущий такт t проверки), получим уравнение

$$W_{hj}(\hat{Z}_h(t-\tau)) \bigvee_y \partial y(t) / \partial z_h(t-\tau) = 1, \quad y \in Y, \quad (4)$$

решение которого необходимо для синтеза теста на проверку неисправности $r_{hj} \in R$. Здесь τ — количество тактов, которое требуется для транспортировки проверяемого сигнала от выхода модуля z_h до выхода устройства y .

4. Описание подхода к синтезу тестов

Решать уравнение (4) для каждой неисправности $r \in R$ в отдельности нецелесообразно по следующим причинам:

- 1) возрастает трудоемкость вычислений, поскольку $|R|$ для реальных цифровых объектов слишком велико;
- 2) теряется возможность отразить регулярность объекта в регулярности теста;
- 3) уменьшается достоверность теста в отношении обнаружения кратных неисправностей в различных модулях или в различных местах сети связей между модулями.

Учитывая все это, мы предлагаем здесь «групповой подход» к решению уравнения (4).

Разобьем множество уравнений (4) для всех $r_{hj} \in R_h$ на две системы: систему уравнений, описывающую регулярную (стандартную) часть теста

$$\bigvee_y \partial y(t_3)/\partial z_k(t_2) = 1, \quad y \in Y, \quad t_2 \leq t_3, \quad (5a)$$

$$\partial z_k(t_2)/\partial z_k(t_1) = 1, \quad t_1 \leq t_2, \quad (5b)$$

$$\partial z_i(t_1)/\partial z_{ih}(t_0) = 1, \quad z_i \in \hat{Z}_k, \quad z_{ih} \in X, \quad t_0 \leq t_1, \quad (5b)$$

и систему уравнений, описывающую нерегулярную (нестандартную) часть теста

$$W_{hj}(\hat{Z}_k(t_1)) = 1, \quad j: r_{hj} \in R_k, \quad (6a)$$

$$z_{ih}(t_0) = z_i(t_1) \oplus d_{ih}, \quad i: z_i \in \begin{cases} \hat{Z}_k^F \setminus z_k & \text{при } r_{kj} \in R_k^F, \\ \hat{Z}_k^M & \text{при } r_{kj} \in R_k^M. \end{cases} \quad (6b)$$

Решение системы уравнений (5) можно рассматривать как общее решение уравнения (4). Оно определяет те части последовательности входных наборов, которые остаются постоянными в течение всей проверки модуля A^k , т. е. неисправностей $r \in R_k$, и образуют, таким образом, регулярную часть теста. Предпосылкой получения общего решения является независимость его от значений $z_i \in \hat{Z}_k$.

Уравнением (5b) решается задача активизации путей транспортировки необходимых стимулов от первичных входов объекта до места проверки (напр., до входов проверяемого модуля) и находится для каждой стимулирующей переменной $z_i \in \hat{Z}_k$ некоторая входная переменная $z_{ih} \in X$, отличная для всех $i: z_i \in \hat{Z}_k^F \setminus z_k$. Уравнением (5a) решается задача активизации путей транспортировки реакций проверяемого модуля на заданные стимулы от этого модуля до одного из первичных выходов объекта. Решение (5b) интерпретируется как запоминание реакций модуля (при проверке неисправностей $r \in R_k^F$) или стимулов (при проверке неисправностей $r \in R_k^M$) в течение $t_2 - t_1$. Последнее требуется в общем случае, когда при решении систем уравнений (5) и (6) возникают противоречия.

Полученное для системы (4) общее решение доопределяется частными решениями системы (6) для каждой конкретной неисправности $r_{hj} \in R_k$. Решением уравнения (6a) находятся необходимые значения переменных $z_j \in \hat{Z}_k$ для этой неисправности, а решением (6b) доопределяются входные наборы объекта. Здесь d_{ih} учитывает количество инаверсий на пути транспортировки соответствующих стимулов, т. е.

$$d_{ih} = \begin{cases} 1, & \text{если } z_i(t_1) = \bar{z}_{ih}(t_0), \\ 0, & \text{если } z_i(t_1) = z_{ih}(t_0), \end{cases}$$

при условии, что $\partial z_i(t_1)/\partial z_{ih}(t_0) = 1$.

Графическое представление решений уравнений (5) и (6) показано на рис. 2. Оно изображает общую временную структуру тестовой последовательности. По вертикальной оси расположены входные переменные $z_i \in X$, по горизонтальной — время. Каждой точке (z_i, t_j) дискретного пространства соответствует некоторое значение z_i в такте t_j . Выделены пять областей, соответствующие решениям пяти задач (уравнения от (5a) до (6b)). Пустые области характеризуют стандартные части теста, повторяющиеся при проверке всех неисправностей $r_{hj} \in R_k$, а заштрихованные — нестандартные части, доопределяющие тест для проверки каждой конкретной неисправности $r_{hj} \in R_k$. Начало цикла T проверки очередной неисправности определяется значением $\min\{t'_1, t_0, t'_3\}$, конец цикла — значением t_3 . В такте t_3 сигналы на

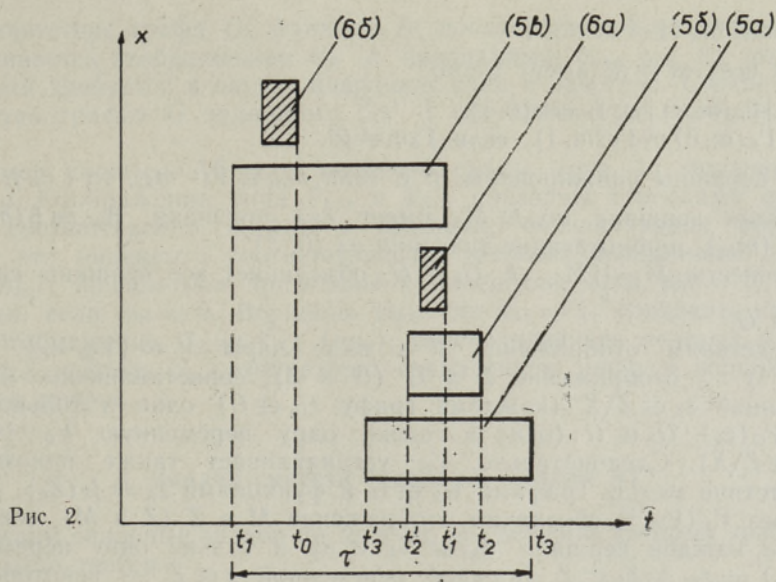


Рис. 2.

выходе сравниваются с эталонными. На следующем такте начинается цикл проверки следующей неисправности.

Полученная регулярная структура теста (стандартные и нестандартные области) позволяет эффективно решать проблему сохранения теста в памяти ЭВМ. Использование однотипных сигналов при активизации путей через объект уменьшает вероятность взаимной компенсации неисправностей в различных модулях и, следовательно, увеличивает достоверность теста при наличии кратных неисправностей. Уменьшение трудоемкости при синтезе тестов по описанному подходу очевидно.

Если обеспечить независимость общих решений (5) от значений $z_i \in \hat{Z}_h$ не удастся, следует проверить непротиворечивость общего и частных решений уравнений (6) для каждого $r_{hj} \in R_h$. В случае противоречивости требуется локальное корректирование общего решения (для $r_{hj} \in R_h$). В такой ситуации, очевидно, роль перечисленных преимуществ подхода уменьшается.

Далее мы предлагаем метод описания ЦУ системой АГ и метод решения систем уравнений (5) и (6) на этой модели.

5. Задание объекта в виде системы АГ

Рассмотрим ЦУ A , заданное функциональным описанием $\Phi(A) = \{Z, F\}$, в виде системы АГ $\Phi_G = \{G, V\}$. Здесь $G = \{G_h\}$ — некоторое множество ориентированных графов, а V — отображение G в Z .

Зададим $G_h \in G$ в виде пары $G_h = \{M_h, \{\Gamma_{h,e}\}\}$, $e \in \{0, 1\}$, где $M_h = \{m_{kj}\}$, $j = 0, 1, 2, \dots$, — множество вершин, а $\Gamma_{h,e}$ — отображение M_h в M_h , где e — тип (альтернатива) отображения. Через $\Gamma_h(m, e) \subset M_h$ ($\Gamma_h^{-1}(m, e) \subset M_h$) обозначим подмножество последователей (предшественников) вершины m в направлении e , через $\Gamma_h m$ ($\Gamma_h^{-1} m$) — объединение этих подмножеств при обоих значениях e , а через $\hat{\Gamma}_h m$ ($\hat{\Gamma}_h^{-1} m$) — подмножество всех потомков (предков) вершины m .

Постулируем следующие свойства G_h :

- 1) G_h является связным и не содержит циклов;

- 2) $\exists! m \in M_h : \Gamma_h^{-1}m = \emptyset$;
- 3) для всех $m \in M$ имеет место
- $|\Gamma_h(m, e)| \leq 1, e \in \{0, 1\}$;
 - $\Gamma_h(m, 0) \neq \Gamma_h(m, 1)$, если $\Gamma_h m \neq \emptyset$;
- 4) все вершины ранжированы, т. е. если $m_{hi} \in \Gamma_h^{-1}m_{hj}$, то $i < j$;
- 5) каждая вершина $m_{hi} \in M_h$ имеет два признака: $\beta_{hi} \equiv \beta(m_{hi})$ и $\tau_{hi} \equiv \tau(m_{hi})$, принимающие значения из $\{0, 1\}$.

Множество $M = \bigcup_h M_h, k : G_h \in G$, объединяет все вершины системы графов G .

Представим отображение V в виде пары $V = \{V_G, V_Z\}$. Здесь $V_G(V_G^{-1})$ — отображение Z в G (G в Z), сопоставляющее каждой переменной $z_h \in Z \setminus X$ (каждому графу $G_h \in G$) один и только один граф $V_G(z_h) : G_h \in G$ (одну и только одну переменную $V_G^{-1}(G_h) = z_h \in Z \setminus X$). Следовательно, V_G устанавливает также однозначное соответствие между графами $G_h \in G$ и функциями $z_h = f_h(Z_h), f_h \in F$.

Через $V_Z(V_Z^{-1})$ обозначим отображение M в Z (Z в M), сопоставляющее каждой вершине $m_{hi} \in M_h$ одну и только одну переменную $V_Z(m_{hi}) = z_{hi} \in Z_h \subset Z$ (каждой переменной $z \in Z$ — некоторое непустое множество вершин $V_Z^{-1}(z) \subset M$), взятую с инверсией, если $\beta_{hi} = 1$, и с запаздыванием на такт, если $\tau_{hi} = 1$.

Согласно описанию модели Φ_G , граф G_h можно интерпретировать как блок-схему алгоритма вычисления функции $z_h = f_h(Z_h)$, а вершины $m_{hi} \in M_h$ — как операторы вычисления некоторого параметра $e(m_{hi}) = e_{hi}$:

$$e_{hi}(t) = \beta_{hi} \oplus z_{hi}(t - \tau_{hi}), \quad (7)$$

где t — текущий такт, на котором вычисляется значение функции $z_h = f_h(Z_h)$. В случае, когда $z_{hi} \notin X$, требуется рекурсивное обращение к некоторому другому графу $G_l = V_{GZ}(m_{hi}) = V_G(V_Z(m_{hi}))$ с целью вычисления для (7) значения z_{hi} .

Выполнение алгоритма по графу G_h начинается из вершины $m_{h0} \in M_h$. Продвижение через любую вершину $m_{hi} \in M_h$ графа производится согласно отображению $\Gamma_h(m_{hi}, e_{hi})$ в одном из двух возможных направлений в зависимости от значения e_{hi} , вычисляемого согласно (7). Выполнение алгоритма заканчивается в вершине m_{hi} , если $\Gamma_h(m_{hi}, e_{hi}) = \emptyset$. Искомое значение функции $z_h = f_h(Z_h)$ равно значению параметра e_{hi} в конечной вершине.

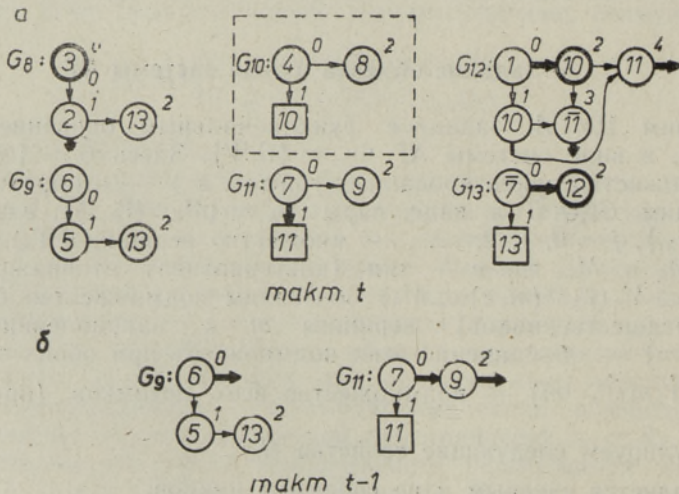


Рис. 3.

Соответствие графа G_h функции f_h , согласно такой интерпретации, обеспечивается отображением V_Z и признаками β_{hi} , τ_{hi} , из которых последний требуется в случае цифровых схем с памятью. Соответствие множества графов G заданному ЦУ A обеспечивается отображением V_G .

Пример системы АГ для заданной схемы (рис. 1) приведен на рис. 3, а. Отображения типа $\Gamma_{h,0}$ и $\Gamma_{h,1}$ показаны стрелками вниз и вправо соответственно. Цифры в кружках, обозначающих вершины m_{hi} , — это индексы j соответствующих весовых переменных $j: z_j = V_Z(m_{hi})$, причем они приведены с инверсией, если $\beta_{hi} = 1$, и без инверсии, если $\beta_{hi} = 0$. Вершины m_{hi} , где $\tau_{hi} = 1$, показаны квадратами. Отображение $G_p = V_{GZ}(m_{hi})$, обеспечивающее переход из вершины m_{hi} в граф G_p , изображено совпадением цифр в вершинах и индексов графов.

6. Задачи синтеза тестов на модели АГ

Рассмотрим основные задачи на модели АГ, решение которых требуется при синтезе тестов.

1. Задача моделирования. Заданы значения переменных $z(\tau)$, $z \in Z$, в моменты времени $\tau = t, t-1, \dots$. Требуется найти значение функции $z_h = f_h(Z_h)$ в такте t . Обозначим оператор, решающий эту задачу, через $G(m_{h0}; t)$, где $k: G_k = V_G(z_h)$. Решение обеспечивает алгоритм $\hat{G}(m_{h0}; t)$:

Шаг 1. $i := 0$.

Шаг 2. Определяется $z_{hi} = V_Z(m_{hi})$.

Шаг 3.

$$e_{hi} := \beta_{hi} \oplus \begin{cases} z_{hi}(t - \tau_{hi}), & \text{если } z_{hi} \in X, \\ G(m_{p0}, t - \tau_{hi}), & \text{если } z_{hi} \notin X, \end{cases}$$

где $p: G_p = V_G(z_{hi})$.

Шаг 4. Если $\Gamma_h(m_{hi}, e_{hi}) = \emptyset$, то переход к 6-му шагу.

Шаг 5. Определяется $j: m_{hj} = \Gamma_h(m_{hi}, e_{hi})$; $i := j$, переход ко 2-му шагу.

Шаг 6. $z_h(t) := e_{hi}$, где $z_h = V_G^{-1}(G_h)$.

Конец.

Формально любой вершине $m_{hi} \in M_h$ можно сопоставить оператор $G(m_{hi})$, значение которого $G(m_{hi}) = q$, $q \in \{m_{hj}, 0, 1\}$, где $m_{hj} \in \hat{\Gamma}_h m_{hi}$, соответствует конечной точке пути, исходящей из m_{hi} и определяемой, согласно приведенному алгоритму, заданными значениями $z \in Z$ при заданных $\tau = t, t-1, \dots$.

2. Установочная задача. Задано необходимое значение $\alpha \in \{0, 1\}$ функции $z_h(t)$. Требуется найти значения $z(\tau)$, $z \in Z$, $\tau = t, t-1, \dots$, обеспечивающие передвижение по графу G_h из m_{h0} до выхода из него в направлении α по алгоритму $G(m_{h0}; t)$. Задача, очевидно, обратна предыдущей.

Для ее решения введем оператор $G^{-1}(m_{h0}, \alpha; t)$, $k: G_k = V_G(z_h)$, которому соответствует алгоритм $G^{-1}(m_{h0}, \alpha; t)$:

Шаг 1. $i = 0$.

Шаг 2. Определяется $z_{hi} = V_Z(m_{hi})$.

Шаг 3. Если значение $z_{hi}(t - \tau_{hi})$ фиксировано, то $e_{hi} := \beta_{hi} \oplus z_{hi}(t - \tau_{hi})$, переход к 9-му шагу.

Шаг 4. Если $z_{hi} \in X$, то переход к 7-му шагу.

Шаг 5. Вычисление по $G^{-1}(m_{p0}, \alpha \oplus \beta_{hi}; t - \tau_{hi})$, где $p: G_p = V_G(z_{hi})$; если решение имеется, то переход к 7-му шагу.

Шаг 6. $e_{hi} := 1 \oplus \alpha$, переход к 8-му шагу.

Шаг 7. $e_{hi} := \alpha$.

Шаг 8. $z_{hi}(t - \tau_{hi}) := e_{hi} \oplus \beta_{hi}$.

Шаг 9. Если $\Gamma_h(m_{hi}, e_{hi}) = \emptyset$, то переход к 11-му шагу.

Шаг 10. Определяется $j: m_{hj} = \Gamma_h(m_{hi}, e_{hi})$; $i := j$; переход к 3-му шагу.

Шаг 11. Если $e_{hi} = \alpha$, то решение найдено, в противном случае решения нет.

Конец.

В результате реализации оператора $G^{-1}(m_{h0}, \alpha; t)$ образуется некоторое упорядоченное подмножество $l(m_{h0}, \alpha) \subset M$, представляющее собой пройденный путь через систему АГ от вершины $m_{h0} \in M_h$ до выхода из графа G_h в направлении α .

Выделим на этом пути некоторое подмножество $M_\alpha = \{m_h | m_h \in l, l(m_h) = \alpha\}$, включающее лишь те вершины m_h , где произошло присваивание значения переменной $V_z(m_h)$ в направлении движения (по 7-му шагу алгоритма). В случае, если при реализации оператора G^{-1} решения не нашлось, требуется провести корригирование пройденного пути l изменением значения переменных, зафиксированных в вершинах $m \in M_\alpha$.

3. Задача активизации пути. Заданы две вершины m_{hi} и m_{hj} в графе G_h и требуется найти значения $z(\tau)$, $z = Z$, $\tau = t, t-1, \dots$, обеспечивающие движение из m_{hi} в m_{hj} по алгоритму $G(m_{hi}, t)$. Сформулированная задача, очевидно, включает как частный случай предыдущую — движение из m_{h0} до выхода из графа в направлении α . Учитывая это, зададим оператор G^{-1} в общем виде $G^{-1}(m_{hi}, q; t)$, где $q \in \{0, 1, m_{hj}\}$, $j \geq i$. Значением оператора $G^{-1}(m_{hi}, q; t)$ будет 1, если решение в виде активизированного пути нашлось, и 0 — в противном случае. Соответствующее изменение алгоритма G^{-1} сводится к уточнению критерия завершения процедуры, что в случае ранжированности вершин тривиально. Через $G^{-1}(m_{hi}, \alpha; t) | m_{hi}$, $\alpha \in \{0, 1\}$ обозначим оператор условной активизации пути, при котором выполняются последовательно операторы $G^{-1}(m_{hi}, m; t)$ и $G^{-1}(m^\alpha, \alpha; t)$ при условии $z_{hi}(t - \tau_{hi}) = \beta_{hi} \oplus \alpha$.

4. Вычисление булевых производных. Требуется найти решение уравнения $\partial z_h(t) / \partial z_i(t - \tau) = 1$, $\tau = 0, 1, \dots$, в случае, когда $m_{hi} \in V_z^{-1}(z_i)$ однозначно определено так, что $k: G_k \in G_G(z_h)$. Для того чтобы, согласно определению булевых производных, изменение значения весовой переменной для m_{hi} приводило к изменению значения функции $z_h = f_h(Z_h)$, представляемой графом G_h , очевидно, требуется решать на модели АГ уравнение

$$\bigwedge_{\alpha \in \{0, 1\}} G^{-1}(m_{h0}, \alpha; t) |_{m_{ht}} = 1, \quad (8)$$

что эквивалентно двойной реализации оператора G^{-1} при условиях $z_{hi}(t - \tau_{hi}) = \beta_{hi} \oplus \alpha$ для $\alpha = 0$ и $\alpha = 1$.

5. Задача синтеза теста. Требуется найти тест на проверку неисправности $r_{hj} \in R_h$, заданной дополнительным условием $W_{hj} = 1$. Согласно уравнениям (4) и (8), находим тест для r_{hj} решением уравнения

$$W_{hj} \bigwedge_{\alpha \in \{0, 1\}} \hat{G}^{-1}(m_{h0}, \alpha; t) |_{m_{ht}} = 1. \quad (9)$$

Рассмотренные выше задачи и введенные операторы G и G^{-1} , работающие на модели $\Phi_G(G, V)$, являются основными при синтезе тестов для заданного ЦУ согласно описанному в разделе 4 подходу.

7. Пример

Рассмотрим цифровую схему (рис. 1) и ее модель в виде системы АГ (рис. 3, а). Построим групповой тест для проверки модуля A^{10} с функцией $z_{10} = f_{10}(z_4, z_8)$.

Согласно (5а), для активизации пути от A^{10} к выходу надо найти решение уравнения

$$\frac{\partial z_{13}(t)}{\partial z_{10}(t-\tau)} = \frac{\partial z_{13}(t)}{\partial z_{12}(t-\tau_1)} \frac{\partial z_{12}(t-\tau_1)}{\partial z_{10}(t-\tau_2)} = 1, \quad (10)$$

где $\tau_1 \leq \tau_2 = \tau$. На модели АГ получим соответственно

$$\bigwedge_{\alpha \in \{0, 1\}} [G_{13}^{-1}(m_{13,0}, \alpha; t) |_{m_{13,2}} \bigwedge G_{12}^{-1}(m_{12,0}, \alpha; t) |_{m_{12,2}}] = 1. \quad (11)$$

Реализацией оператора G_{13}^{-1} активизируется путь $\{m_{13,0}, m_{13,2}\}_t$, реализацией оператора G_{12}^{-1} — пути $\{m_{12,0}, m_{12,2}\}_t$, $\{m_{12,4}\{m_{11,0}, m_{11,1}\{m_{11,0}, m_{11,2}\{m_{9,0}, 1\}, 1\}_{t-1}, 1\}, 1\}_t$ и $\{m_{12,3}, 0\}_t$. Они показаны на рис. 3, а жирными стрелками для такта t и на рис. 3, б для такта $t-1$. При этом $\tau_1=0$, $\tau_2=\tau=1$.

Согласно (5в), для активизации путей от входа к A^{10} находим решение уравнения

$$\frac{\partial z_8(t)}{\partial z_3(t-\tau)} = 1.$$

Результат показан жирной стрелкой на графе G_8 для такта t (рис. 3, а). При этом $\tau=0$.

Решение уравнения (5б) в данном случае не требуется, так как элементарные тесты для A^{10} вполне совместимы с полученными решениями (5а) и (5в). Конечный вид стандартной части теста длиной в два такта приведен в таблице. Место нестандартной части выделено жирными линиями. Активизированные этим тестом пути в реальной схеме показаны на рис. 1 также жирными линиями.

Такт	z_1	z_2	z_3	z_4	z_5	z_6	z_7	z_{12}
$t-1$						1	1	
t	1	0					0	

8. Заключение

Синтез тестов ЦУ на уровне модулей заключается в декомпозиции ЦУ на функционально единые подсхемы (модули), для которых локальные подтесты или берутся из архива (для стандартных подсхем), или синтезируются по известным методам, а активизация локальных подтестов через другие подсхемы производится на модели АГ ЦУ. В отличие от традиционного модульного подхода, по которому активизация подтестов производится путем склеивания подтестов других подсхем, предложенный метод обеспечивает большую результативность (т. е. если тест существует, его нахождение гарантировано).

ЛИТЕРАТУРА

1. Шнейдер Б. Н. В кн.: II Всесоюзное совещание по теории релейных устройств и конечных автоматов. Тезисы докладов. Рига, «Зинатне», 1971, с. 71—72.
2. Плакк М., Убар Р. Тр. Таллин. политехн. ин-та, № 432, 3—13 (1977).
3. Ubar, R., Nachrichtentechn. Elektron., 27, № 4, 149—150 (1977).

Институт кибернетики
Академии наук Эстонской ССР

Таллинский политехнический
институт

Поступила в редакцию
26 апреля 1982

T. LOHUARU, M. PALL, R. UBAR

DIGITAALSKEEMIDE TESTIDE AUTOMAATNE SÜNTEES

Artikkel käsitleb digitaalskeemide testide sünteesi süsteemi tootlikkuse suurendamise ja arvuti mälu säästmise võimalusi traditsiooniliste meetoditega võrreldes. On esitatud testide sünteesi meetod, mille aluseks on digitaalskeemide kirjeldamine hierarhilise alternatiivsete graafide süsteemina. Testide sünteesi ülesanne on taandatud graafidel põhinevale teede genereerimise ülesandele.

T. LOHUARU, M. PALL, R. UBAR

AUTOMATIC TEST GENERATING FOR DIGITAL DEVICES

The proposed approach to troubleshooting in digital circuits is based on test program generation on modular level. The test programs of modules are obtained from the computers memory in completed form, or they are generated using some known method. The generation of a test capable to test the whole circuit involves the development of a system of tests of modules so that a single-dimensional or a multidimensional activated path consisting of modules under test, is provided between the input and output of the digital circuit. To develop the algorithms implementing the process of activation of paths, one needs functional descriptions of the modules. A functional description written as a Boolean expression has the disadvantage of complexity or an excess memory capacity requirement. To economized on the computer memory and to increase the performance of the automatic test generating system, a method is proposed to describe the functions of the digital circuit by the system of hierarchical alternative graphs and to form the activated paths in it.