

МОДЕЛИ ДИАГНОСТИРОВАНИЯ ИЕРАРХИЧЕСКИХ СИСТЕМ НА ОСНОВЕ ТЕХНОЛОГИЙ ГРАНИЧНОГО СКАНИРОВАНИЯ

Предлагаются модели тестирования сложных программно-технических комплексов, ориентированные на решение задач диагностирования в реальном масштабе времени. Модели используют технологии граничного сканирования IEEE стандартов для наблюдения невыходных линий и методы оценки тестопригодности для определения критических мест в цифровых объектах. Модели и методы ориентированы на тестирование рассредоточенных систем управления критическими технологиями.

1. Введение

Основными требованиями, предъявляемыми к современным информационным и управляющим системам для сложных объектов и критических технологий [1-7], являются: 1) обеспечение высокой надежности их функционирования [5]; 2) оперативный контроль и управление всеми параметрами критической системы или объекта; 3) тестирование, диагностика и восстановление работоспособности подсистем за допустимое техническими условиями и стандартами время; 4) обеспечение требуемой глубины диагностирования системы или ее компонентов автоматически и в реальном масштабе времени.

Новое поколение современных технологий и маршрутов проектирования вводит также дополнительные критерии, связанные с проектированием, производством и эксплуатацией цифровых изделий: time-to-market, Design for - Manufacturability, Testability, Diagnosis, Verification [8-19]. Важнейшим этапом разработки компонентов цифровой системы является процесс верификации, связанный с устранением всех ошибок проектирования как можно на более ранней стадии, что приводит к значительной экономии временных и материальных ресурсов. Здесь интерес представляет внесение допустимой тестопригодной избыточности (ассерции) в состав проекта на ранней (системной) стадии разработки, которая, совместно с применением методов верификации и тестирования, существенно уменьшает основной параметр – time-to-market, что является актуальной и рыночно привлекательной моделью проектирования [8-14].

Речь идет об использовании теста верификации, полученного на стадии системного проектирования, для проверки изделия с минимальными дополнительными аппаратными и программными затратами с применением технологии граничного сканирования [8,10]. При этом механизм программно-аппаратной избыточности должен включать дополнительные точки контроля, которые необходимо ввести в проект с помощью Boundary Scan регистра или специальных (ad hoc) технологий на стадии синтеза. В результате однажды созданная избыточность проекта может быть многократно использована для проверки компонентов цифровой системы на всех стадиях его жизненного цикла.

Сложные цифровые изделия в настоящее время рассматриваются как объекты, имеющие несколько уровней иерархии. На первом самом низком уровне система представлена современными интегральными схемами (PLD, ASIC), в которые имплементированы SoC, NoC, память, процессоры. Второй уровень формируется цифровыми устройствами, представленными на платах (system on board), где в качестве примитивов выступают интегральные схемы первого уровня. Третий уровень представляет собой совокупность плат, которые объединяются в понятие крейта (system on crate). Четвертый уровень объединяет совокупность крейтов или шкафов в сложную распределенную систему управления технологическими процессами, производствами или критическими технологиями (авиация, космонавтика, атомная энергетика, метеорология, оборона, экология) [1-3]. Пятый уровень можно рассматривать как рассредоточенную в пространстве систему, примером которой может быть Internet. Далее рассматривается иерархия системы от первого до четвертого уровней в целях создания моделей и методов их тестирования с заданной глубиной диагностирования.

Цель разработки – существенное уменьшение времени тестирования сложных цифровых систем в процессе эксплуатации на основе создания общей модели организации и проведения диагностического эксперимента, включающего безусловные алгоритмы поиска дефектов с использованием IEEE стандартов тестопригодного проектирования [8-14].

Для достижения поставленной цели решаются *задачи*: 1) Выбор приемлемых методов и средств тестирования сложных цифровых систем всех, обозначенных выше, уровней иерархии. 2) Разработка иерархической модели организации и проведения диагностического эксперимента, включающей условные и безусловные алгоритмы поиска дефектов, ориентированные на использование стандартов тестопригодного проектирования. 3) Практическая реализация моделей тестирования сложных цифровых систем и экспериментальная оценка их состоятельности.

2. Модель тестирования цифровой системы

Объект исследования – цифровая система управления, представленная несколькими уровнями иерархии, разрабатываемая на основе IEEE стандартов тестопригодного проектирования. В частности, рассматривается программно-технический комплекс F , как объект тестирования, который может быть представлен в виде кортежа: $F = \langle C, V, P, M \rangle$, где $C = \{c_1, c_2, \dots, c_n\}$ – конечное непустое множество крейтов в цифровой системе, $V = \{b_1, b_2, \dots, b_m\}$ – множество печатных плат в крейте, $P = \{p_1, p_2, \dots, p_k\}$ – совокупность интегральных схем на печатной плате, $M = \{m_1, m_2, \dots, m_g\}$ – множество функциональных IP-модулей на чипе (в интегральной схеме). При этом выполняются следующие условия вложения или иерархии:

$$m_{ijrs} \in p_{ijr} \in b_{ij} \in c_i \in F, (i = \overline{1, n}; j = \overline{1, m}; r = \overline{1, k}; s = \overline{1, g}). \quad (1)$$

Основное уравнение диагноза для рассматриваемого объекта исследования может быть представлено в следующем виде:

$$D = R^* \wedge L = (R^+ \oplus R^-) \wedge L = [(T \oplus F) \oplus (T \oplus F^*)] \wedge L, \quad (2)$$

где параметры формулы определяются как: D – множество обнаруженных в устройстве дефектов в процессе тестирования; $L(T_i)$ – список неисправностей, обнаруживаемых тест-вектором

$$T_i \in T = (T_1, T_2, \dots, T_i, \dots, T_h),$$

представленный в форме таблицы

$$L(t_i) = [L_{ij}], i = \overline{1, p}; j = \overline{1, q}.$$

здесь каждому тесту ставится в соответствие подмножество проверяемых дефектов, p – число наблюдаемых выходов, q – общее количество линий в устройстве;

$R^-(T_i) = (R_1^-, R_2^-, \dots, R_i^-, \dots, R_p^-)$ – двоичный вектор элементарной экспериментальной проверки, равный числу наблюдаемых выходов системы; $R^+(T_i) = (R_1^+, R_2^+, \dots, R_i^+, \dots, R_p^+)$ – эталонный вектор состояния выходов; F – эталонная модель устройства; F^* – реальное изделие.

Для организации процесса диагностирования формируются координаты вектора

$$R^*(T_i) = (R_1^*, R_2^*, \dots, R_i^*, \dots, R_p^*)$$

в соответствии со следующим правилом:

$$R_i^* = R_i^+ \oplus R_i^-. \quad (3)$$

Идеальная модель для диагностирования дефектов цифровой системы представляется в виде следующей структуры: $F = \{T \times L \times R\}$. Однако мощность такой трехмерной таблицы определяется выражением $|F| = h \times 2q \times p$. Если учесть, что размерность теста, в лучшем случае, есть квадрат от числа линий устройства, то данное выражение приводится к следующему виду: $|F| = \{q^2 \times 2q \times p\} = 2q^3 p \approx q^4$. Если число линий в реальных объектах от

100 000 до 1000 000, то размерность трехмерной таблицы диагностирования будет расположена в интервале значений $10^{20} \geq |F| \geq 10^{24}$. Естественно, что такой объем диагностической информации практически невозможно обработать в целях поиска дефекта в цифровой системе. Возникает проблема сжатия данных до приемлемых размеров. Практически сжимать можно все три компонента $\{T, L, R\}$ [11]: тест, дефекты и реакции на входные наборы, но при этом теряется или точность установления диагноза, или время проведения тестового эксперимента. Далее предлагается метод с максимально возможной глубиной диагностирования, основанный на выполнении тестового эксперимента над моделью и объектом в реальном масштабе времени, когда исключается из рассмотрения параметр T , а объем информации переходит плоскость – теряет одно измерение, связанное с длиной теста. В связи с этим предлагается следующая модель диагностирования:

$$\begin{aligned}
 & 1) \left\{ \begin{aligned} & L(T_i) = F \oplus T_i; \\ & R^*(T_i) = f(F, T_i, D^*); \\ & D(T_i) = g[L(T_i), R^*(T_i)]. \end{aligned} \right. \\
 & 2) \left\{ \begin{aligned} & D^s(T_i) = \bigwedge_{\forall j(R_j^*=1)} L_j \& \overline{\bigvee_{\forall j(R_j^*=0)} L_j}; \\ & D^m(T_i) = \bigvee_{\forall j(R_j^*=1)} L_j \& \overline{\bigvee_{\forall j(R_j^*=0)} L_j}. \end{aligned} \right. \\
 & 3) \left\{ \begin{aligned} & D_0^s(T) = D_0^s(T) \vee D^s(T_i) \& S^+(T_i); \\ & D_1^s(T) = D_1^s(T) \vee D^s(T_i) \& \bar{S}^+(T_i). \end{aligned} \right. \tag{4} \\
 & 4) \left\{ \begin{aligned} & D_0^m(T) = D_0^m(T) \vee D^m(T_i) \& S^+(T_i); \\ & D_1^m(T) = D_1^m(T) \vee D^m(T_i) \& \bar{S}^+(T_i). \end{aligned} \right.
 \end{aligned}$$

Здесь первые три равенства представляют процедуры: формирования списков проверяемых дефектов по наблюдаемым выходам объекта; вычисления фактической реакции устройства на заданную входную последовательность; определения дефектов, присутствующих в изделии на момент тестирования. Вторая система уравнений позволяет определить наличие одиночных или кратных константных дефектов в устройстве путем выполнения двух процедур над содержимым таблицы L проверяемых на входном наборе T_i дефектов и вектора экспериментальной проверки R^* . Третья система уравнений позволяет вычислить вектор одиночных дефектов, присутствующих в цифровой системе на момент тестирования, как функцию от вектора $S^+(T_i)$ исправного состояния линий и полного теста T . Четвертая система уравнений позволяет накапливать вектор кратных дефектов, присутствующих в цифровой системе на момент тестирования, так же как функцию от вектора $S^+(T_i)$ исправного состояния линий. Результатом выполнения диагностического эксперимента, проводимого в реальном масштабе времени, является множество дефектов, одиночных или кратных, которые имеются в цифровой системе. При этом затраты памяти для хранения информации и проведения диагностического эксперимента определяются выражением:

$$Q^M = (p^x \times q^2) + (q \times p) + 7q + 3p. \tag{5}$$

Если использовать встроенный генератор тестов, для которого нет необходимости хранить информацию, то объем памяти будет минимальным:

$$Q_A^M = (q \times p) + 7q + 3p. \tag{6}$$

Естественно, что и вычислительная сложность проведения диагностического эксперимента является функцией от объема обрабатываемой памяти, представленной выражениями (5) и (6).

На рис. 1 представлена структура вычислителя для проведения диагностического эксперимента, которая построена на основе уравнений (4). Достоинством устройства является выполнение операций над таблицами и векторами, что легко реализуется как в компьютере, так и в кристалле PLD.

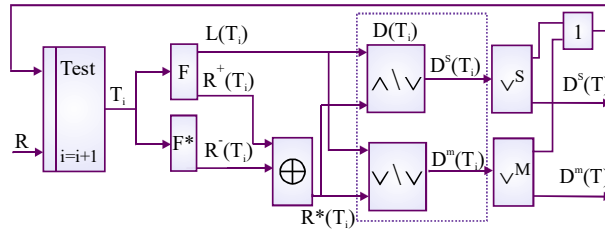


Рис. 1. Структурная модель диагностирования цифровой системы

В качестве примера структур данных, необходимых для проведения диагностического эксперимента, ниже предлагаются таблицы и векторы, ориентированные на иллюстрацию выполнения процедур, определенных упомянутым вычислителем (см. рис. 1) и уравнениями (4):

$$L(T_i) = [L_{ij}] \quad q = 8 \quad R^*(t) = R^-(t) \oplus R^+(t)$$

$$p = 4 \quad \begin{array}{cccccccc|ccc} 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \end{array}$$

$$D^s(T_i) = \begin{array}{|c|c|c|c|c|c|c|c|} \hline 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline \end{array}$$

$$D^m(T_i) = \begin{array}{|c|c|c|c|c|c|c|c|} \hline 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ \hline \end{array}$$

$$S^+(T_i) = \begin{array}{|c|c|c|c|c|c|c|c|} \hline 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ \hline \end{array}$$

$$D_0^s(T) = \begin{array}{|c|c|c|c|c|c|c|c|} \hline 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline \end{array}$$

$$D_1^s(T) = \begin{array}{|c|c|c|c|c|c|c|c|} \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline \end{array}$$

$$D_0^m(T) = \begin{array}{|c|c|c|c|c|c|c|c|} \hline 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline \end{array}$$

$$D_1^m(T) = \begin{array}{|c|c|c|c|c|c|c|c|} \hline 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \hline \end{array}$$

Здесь рассматривается объект диагностирования, имеющий 4 выхода ($p=4$), где каждому из них ставится в соответствие вектор проверяемых дефектов, равный общему числу линий в объекте, в данном случае $q=8$. Проверимость дефекта в ячейке рассматриваемой таблицы $L(T_i)$ идентифицируется символом 1, непроверяемость – нулем. Тип константного дефекта определяется инверсией от исправного состояния соответствующей линии вектора $S^+(T_i)$. Состояния координат вектора экспериментальной проверки $R^*(T_i)$ вычисляются как сумма по модулю 2 векторов: исправного состояния модели $R^+(T_i)$ и фактических реакций $R^-(T_i)$ выходов реального объекта. Таким образом, единичное (нулевое) значение координаты вектора $R^*(T_i)$ идентифицирует несовпадение (совпадение) по выходам эталонной и экспериментальной реакций и позволяет точно определять дефекты в цифровой системе. При этом используются две стратегии: поиск одиночных и кратных дефектов на каждом тест-векторе, отображаемых векторами $D^s(T_i), D^m(T_i)$ соответственно. Накопление информации, по мере подачи входных последовательностей, происходит уже отдельно в векторах $D_0^s(T), D_1^s(T), D_0^m(T), D_1^m(T)$, которые формируют списки одиночных и кратных, нулевых и единичных константных дефектов. При этом используются системы уравнений 3 и 4 из (4).

Таким образом, для иерархической модели сложной цифровой системы управления экономичнее проводить диагностический эксперимент в реальном масштабе времени путем анализа модели и реального объекта на каждом входном наборе в целях определения и сравнения их реакций на каждую входную последовательность. В этом случае необходимо хранить информацию только о состоянии выходов и результаты моделирования неисправностей на одном тест-векторе.

Выбор формата данных определяется приоритетами разработчика. Что целесообразнее: уменьшение объема информации при использовании списков неисправностей, но с одновременным усложнением процедур диагностирования и операций над списками, или технологичные алгоритмы на основе регистровых операций при существенном увеличении затрат памяти? Поскольку две упомянутые реализации имеют только формальные отличия, то можно предоставить квалифицированному пользователю два режима, которые существенно улучшат функциональность продукта и расширят круг пользователей, занимающихся созданием средств диагностирования.

Предложенная модель тестирования может быть также использована при ее реализации на дереве контроллеров, управляющих BS-регистрами, которые позволяют сканировать только очередную точку контроля, необходимую для установления точного диагноза.

3. Тестирование с использованием IEEE стандартов

Основной недостаток существующих методов и маршрутов проектирования ведущих фирм мира [21-23] заключается в том, что реализация нефункциональных свойств изделия, каковыми являются избыточные средства тестирования и диагностирования, не всегда поддерживается. К тому же избыточность требует определенных временных и материальных затрат, которые окупаются лишь в процессе эксплуатации, когда возникают нештатные ситуации, связанные с повреждениями или отказами. Тем не менее разработчики средств проектирования все более стараются учитывать требования современного рынка и IEEE стандарты тестопригодности и надежности. Что же дают упомянутые стандарты в плане улучшения процесса и результата диагностирования сложных иерархических систем? Естественно, рассматривается критерий эффективности: время диагностирования с наперед заданным качеством, определяемым глубиной поиска дефекта.

Существующие методы тестирования не свободны от элементарных проверок, позволяющих уточнить место и вид дефекта в системе. При этом временные затраты элементарной проверки при выполнении условного диагностического эксперимента определяются выражением

$$T_d = T_s + T_k + T_t + T_r, \quad (7)$$

где T_s – время поиска координаты микросхемы, платы, крейта, шкафа для установки (сигнаурного) зонда; T_k – время проверки наличия связи между зондом и выводом для наблюдения реакции; T_t – время подачи теста на объект диагностирования, снятия экспериментальной реакции в точке контроля и сравнения ее с эталонной; T_r – время восстановления работоспособности. Материальные затраты на реализацию элементарной зондовой проверки пропорциональны времени T_d . Естественно, что наиболее весомыми компонентами являются T_s и T_r . Второй компонент трудно исключить из процедуры диагностирования, поскольку он связан с человеческим фактором. Первый же можно устранить из диагностического эксперимента путем исключения зондирований как элементарных проверок. Для этого следует перейти к безусловному диагностическому эксперименту, убирающему неоднозначность тестирования и установления диагноза путем введения дополнительных линий наблюдения на основе использования существующих IEEE-стандартов.

Какие же линии нуждаются в дополнительном наблюдении? Если коротко, те, которые позволяют разделить эквивалентные неисправности [24] и привести их к собственным подмножествам на существующем тесте диагностирования. Например, если число наблюдаемых выходов – p , а длина теста – h , то q , как число неисправностей, которое можно обнаружить и различить, определяется формулой

$$q = 2^{p \times h} \quad (p \neq 0, h \geq 2). \quad (8)$$

Однако эквивалентные неисправности – это хорошо для проектов, насчитывающих миллионы вентиляей. Они дают возможность существенно уменьшить длину теста (пара-

метр h), но при этом ухудшить его диагностические свойства. Для компенсации или восстановления упомянутых свойств необходимо увеличивать параметр p в формуле (8). Таким образом, дополнительная наблюдаемость внутренних линий дает возможность существенно повысить глубину диагностирования цифрового изделия и в пределе получить возможность различать каждый дефект в объекте. С другой стороны, уменьшение (увеличение) одного из параметров в (8) должно быть скомпенсировано увеличением (уменьшением) другого. В пределе на паре входных наборов можно различить все неисправности, если этому способствует логика цифрового устройства (сумматор, имеющий 3 наблюдаемых входа и два выхода). Но при этом в пределе все линии должны быть наблюдаемы.

Для формализации процесса поиска дополнительных линий наблюдения вводится модель, зависящая от следующих параметров:

$$Z = f(S, T, C, O, T^*), \quad (9)$$

где S – структура объекта, T – тест; C и O – управляемость и наблюдаемость линий; T^* – тестируемость изделия.

Общее количество наблюдаемых выходов (априорных и дополнительных) определяется следующим выражением:

$$Y = \{Z, R, y\}, \quad (10)$$

здесь Z – количество дополнительных линий в объекте, подлежащих определению по методу CAMELOT [24]; R – множество сходящихся разветвлений, которые не проверяются базовым тестом; y – априорные выходы схемы.

Дополнительные выходные линии определяются путем вычисления наблюдаемости $O(X_i)$ всех переменных схемы. После этого необходимо выбрать совокупность таких линий X_i , которые удовлетворяют наперед заданному пороговому значению наблюдаемости $O(X_i)$ или тестируемости $T^*(X_i)$:

$$X_i \in Y \leftarrow [O(X_i) \leq \{O_{\max}, T_{\max}^*\} \vee X_i \in R]. \quad (11)$$

Из множества Y следует исключать все входные линии схемы. Кроме того, при прочих равных характеристиках тестопригодности предпочтение следует отдавать сходящимся разветвлениям R . Применяя данное выражение необходимо вычислять значения управляемости, наблюдаемости и тестопригодности для каждой линии схемы. Для этого необходимо использовать следующие определения.

1. Управляемость – количественная мера способности устройства генерировать на рассматриваемой линии значение 0 или 1, которая зависит от логической функции устройства и уменьшается по мере отдаления линии от внешних входов схемы:

$$C(Y) = C_i \cdot C^*(X_i), \quad (12)$$

где $C_i = \left| N_i^0 - N_i^1 \right| / (N_i^0 + N_i^1)$, – коэффициент передачи управляемости; $N_i^0, (N_i^1)$ – число

всех способов установки логического нуля (единицы) на линии; $C^*(X_i) = \frac{\sum_{i=1}^n C(X_i)}{\sum_{i=1}^n X_i}$ –

среднее арифметическое управляемостей входов примитива.

2. Наблюдаемость – количественная мера способности устройства транспортировать состояние рассматриваемой линии на внешние выходы схемы, которая зависит от логической функции устройства и уменьшается по мере отдаления линии от внешних выходов схемы:

$$O(Y) = O_i \cdot O(X_i) \cdot C^*(X_i), \quad (13)$$

где $O_i = \left| T^a - T^b \right| / (T^a + T^b)$ – коэффициент передачи наблюдаемости; T^a – число наборов,

активирующих X_i на выход примитива, T^b – число наборов, блокирующих активизацию X_i на выход примитива; $O(Y)$ – наблюдаемость внешних выходов схемы $O(X_i) = 1$.

3. Тестируемость – мультипликативная количественная оценка способности устройства транспортировать управляемость (наблюдаемость) рассматриваемой линии на внешние

входы (выходы) схемы, которая зависит от логической функции устройства и уменьшается по мере отдаления линии от внешних входов (выходов) схемы:

$$T^* = C(Y) \cdot O(Y). \quad (14)$$

Таким образом, представленные выше критерии тестопригодности дают возможность не только оценить проектируемое изделие, но и определить узкие места в проекте с точки зрения управляемости, наблюдаемости и тестируемости. Далее стандарты Boundary Scan IEEE 1149.1 и IEEE P1500 [8,12-14] делают наблюдаемыми (управляемыми) линии объекта, благодаря технологии их сканирования, незначительно увеличивающей время проведения диагностического эксперимента.

Что касается иерархического проекта, то его структура может быть представлена в виде мультипликативного дерева, где каждая вершина – компонент по отношению к верхнему уровню иерархии – имеет одну входящую дугу и несколько исходящих (рис. 2).

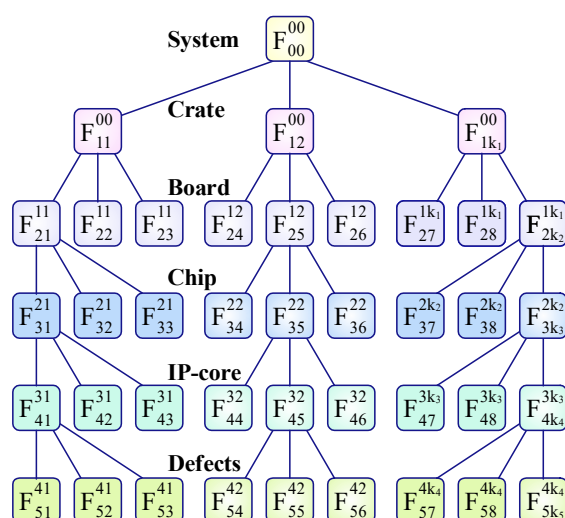


Рис. 2. Иерархическая модель сложной цифровой системы

Математическая модель такой нисходящей иерархии представлена в виде системы уравнений:

$$\begin{cases} F = [F_{ij}^{rs}], \quad (i = \overline{0,5}; j = \overline{1,k}; r = \overline{0,4}; s = \overline{1,k_{j-1}}); \\ T = [T_{ij}^{rs}]; \quad L = [L_{ij}^{rs}] \\ F_{ij}^{rs} \oplus T_{ij}^{rs} = L_{ij}^{rs}. \end{cases} \quad (15)$$

Полный идентификатор компонента системы $F = [F_{ij}^{rs}]$ имеет четыре индекса. Нижние индексы (ij) служат для обозначения уровня иерархии и порядкового номера компонента в нем. В целях установления связи с его родительской вершиной имеются два верхних индекса – rs, которые задают предшествующий уровень иерархии и номер элемента в нем. Каждому компоненту сложной цифровой системы ставится в соответствие тест, дефекты и правила установления диагноза.

Применяются две стратегии тестирования. Первая использует тест проверки функциональности, который позволяет определить исправность блока (системы) или зафиксировать непрохождение теста. Лишь на последнем уровне иерархии тест дает список неисправностей, присутствующих в системе. Процедура диагностирования в худшем случае предполагает тестирование всех компонентов объекта. Вторая стратегия – более избирательная за счет использования сложных и объемных тестов диагностирования. Тестирование текущего блока дает точный адрес компонента, в котором далее следует искать неисправность.

Тест более высокого уровня должен указывать на неисправные компоненты на один уровень ниже:

$$\begin{aligned}
 F_{1j}^{rs} \oplus T_{1j}^{rs} &= L_{1j}^{rs} = \{L_{21}^{1j}, L_{22}^{1j}, \dots, L_{2k_j}^{1j}\}; \\
 F_{2j}^{rs} \oplus T_{2j}^{rs} &= L_{2j}^{rs} = \{L_{31}^{2j}, L_{32}^{2j}, \dots, L_{3k_j}^{2j}\}; \\
 F_{3j}^{rs} \oplus T_{3j}^{rs} &= L_{3j}^{rs} = \{L_{41}^{3j}, L_{42}^{3j}, \dots, L_{4k_j}^{3j}\}; \\
 F_{4j}^{rs} \oplus T_{4j}^{rs} &= L_{4j}^{rs} = \{L_{51}^{4j}, L_{52}^{4j}, \dots, L_{5k_j}^{4j}\}.
 \end{aligned}
 \tag{16}$$

Это дает возможность существенно уменьшить длину теста для верхнего компонента дерева, а также определить локальные тесты для вершин более низкого уровня. Плата за такие удобства – доступность каждого модуля иерархической системы по входам и выходам, благодаря технологии граничного сканирования. Кроме того, иерархия позволяет значительно уменьшить количество рассматриваемых (моделируемых и диагностируемых) дефектов за счет разделения целостного объекта на части. Уменьшается также общая длина теста и время его построения. Если учесть, что вычислительная сложность решения задачи синтеза теста есть функция от количества эквивалентных вентилях ($Q = n^2$), то для устройства, содержащего миллион линий, данная оценка будет иметь значение $Q = 10^{12}$. Если же разбить устройство на 100 компонентов, то вычислительная сложность синтеза тестов будет определяться оценкой $Q = 100 \times (10^4)^2 = 10^{10}$. Аналогично может быть существенно уменьшен объем таблиц неисправностей при иерархическом описании модели объекта. Суммарный объем всех таблиц дефектов компонентов всегда будет значительно меньше, чем наличие одной для всей цифровой системы.

Еще одна стратегия подготовки теста для системы связана с рассмотрением моделей нижнего уровня иерархии. Для каждого IP-cores строится тест проверки неисправностей константного типа. Затем, поднимаясь по иерархии вверх, уже построенные тесты объединяются в соответствии с их принадлежностью к чипу, плате, крейту.

Следующая структура взаимодействия технологии граничного сканирования и иерархической модели объекта иллюстрирует возможность доступа ко входам и выходам каждого компонента цифровой системы в процессе выполнения процедуры диагностирования (рис. 3).

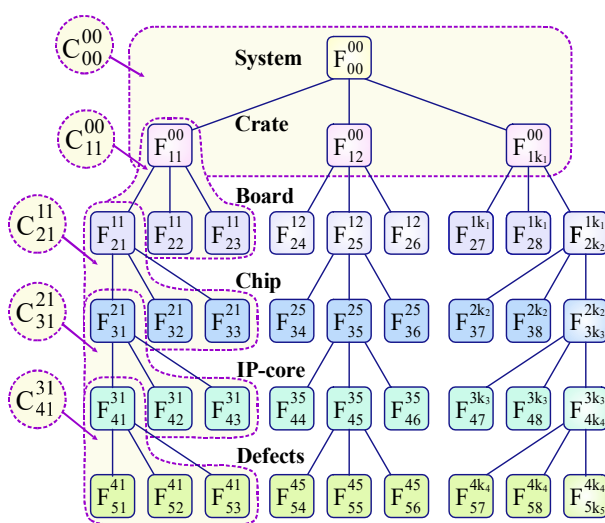


Рис. 3. BS-wrapper иерархической модели цифровой системы

Модель В/S-контроллера является универсальной по отношению к одной родительской вершине и ближайшим ее компонентам-преемникам:

$$F_{ij}^{rs} = \{F_{i+1,1}^{ij}, F_{i+1,2}^{ij}, \dots, F_{i+1,k_j}^{ij}\}. \quad (17)$$

Например, контроллер: $C_{31}^{21} \approx F_{31}^{21} = \{F_{41}^{31}, F_{42}^{31}, F_{43}^{31}\}$ (см. рис. 3), осуществляет сканирование данных в пределах одного чипа, содержащего три функциональных блока – IP-cores: $\{F_{41}^{31}, F_{42}^{31}, F_{43}^{31}\}$. Распространив влияние формулы (17) на модель системы (рис. 4), можно получить иерархическую структуру В/S- контроллеров.

Несмотря на кажущуюся сложность такой архитектуры, аппаратурная избыточность В/S-системы тестирования и диагностирования составляет порядка 2-5 % от общей функциональности, что является допустимым для современных chip set таких фирм как Xilinx, Altera, и привлекательным для заказчиков сложных систем управления критическими технологиями. Что касается ухудшения характеристик быстродействия функциональных модулей, то он составляет не более 0,5% от расчетных характеристик, без использования технологий граничного сканирования. Однако существуют определенные временные затраты при выполнении диагностического эксперимента, связанные со сканированием входных и выходных данных к конкретному блоку. Процесс замедления тестирования зависит от количества наблюдаемых точек (разрядов регистров граничного сканирования) и связан с ним следующим выражением:

$$Q(F) = \sum_{j=1,5} k_j \times p_j \times t^*. \quad (18)$$

Здесь k_j – число регистров граничного сканирования на j -м уровне иерархии; p_j – количество разрядов в регистрах j -го уровня иерархии; t^* – время сдвига информации в регистре. При наличии в проекте 1 000 000 эквивалентных вентилях число наблюдаемых линий может достигать 1 000. В данном случае аппаратурная избыточность будет равна 2,1%, скорость тестирования системы уменьшится в 1 000 раз и составит не более 100 мс/ тест-вектор.

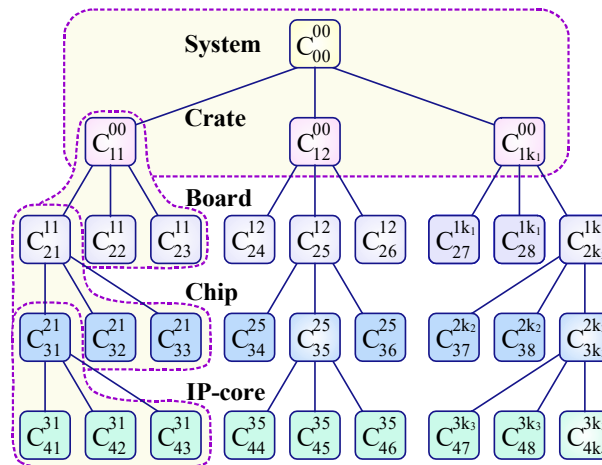


Рис. 4. Иерархическая структура В/S-контроллеров сложной системы

Основная идея стандартов граничного сканирования – легко наблюдать и управлять внутренними линиями изделия за счет введения избыточности в виде Boundary Scan регистра и Test Access Port контроллера. Естественно, что первоначально ориентированная на диагностирование межсоединений на платах стратегия граничного сканирования далее получила распространение как вверх по иерархии, до сложных цифровых систем на крейтах, так и вниз – до функциональных модулей микросхемы, обозначенных IP-cores. Архитектуры сканирования информации (входные тесты и реакции линий на них) в процессе тестиро-

вания и диагностирования близки по идеологии, но отличаются лишь точкой их приложения – различным объектом тестирования. Типовая В/С- структура представлена на рис. 5. Она ориентирована на решение задач тестирования и диагностики внутри одного чипа размерностью более 1 000 000 вентиляей. Функциональность микросхемы разбивается на отдельные модули, для которых строятся тесты и алгоритмы диагностирования. Другое возможное решение не связано с разбиением – наблюдаемыми (управляемыми) становятся критические внутренние линии, которые находятся с помощью известных методов оценки тестопригодности (управляемость, наблюдаемость) [11,25]. Данная структура может быть использована для проверки межсоединений на плате, тестирования и поиска дефектов внутри каждой микросхемы, для диагностики плат, объединенных понятием крейта или шкафа, для тестирования разнесенных в пространстве компонентов системы (компьютеры и крейты), объединенных понятием информационной сети.

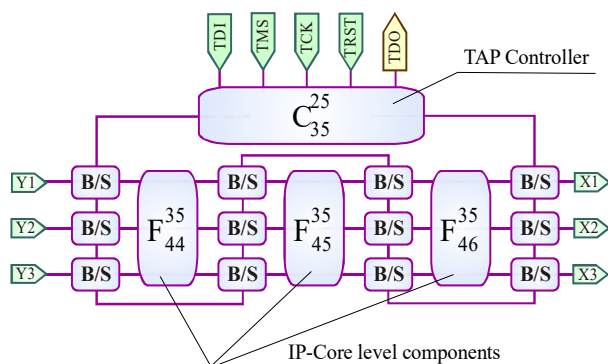


Рис. 5. Структурные схемы IEEE стандартов сканирования

Общая модель процесса тестирования сложной иерархической цифровой системы в целях проверки работоспособности и установления места и вида дефекта имеет вид, представленный на рис.6. Используются тесты проверки неисправностей и функциональности, которые не дают возможности точно установить диагноз относительно компонентов нижнего уровня. Процедура проверки начинается с тестирования функциональности системы. В случае фиксации ошибки тест-процессор опускается на уровень ниже и последовательно проверяет все крейты. При определении неисправности в одном из них процессор переходит к тестированию и поиску неисправной платы. После нахождения таковой выполняется проверка каждой микросхемы на ней. При нахождении неисправного чипа осуществляется поиск дефектного IP-core. Последний этап поиска в нем ориентирован на определение неисправной линии или элемента. На любом уровне иерархии, если глубина диагностирования удовлетворяет пользователя (D+), возможна замена дефектного компонента или конструктива – восстановление работоспособности (BP) – с последующим возвратом к тестированию компонента и объекта в целом.

Принципы организации системы тестирования:

- 1) Использование наиболее подходящих IEEE стандартов [8-14] для верификации компонентов рассматриваемого уровня.
- 2) Установление диагноза заданной глубины в автоматическом режиме, без использования условных алгоритмов поиска дефектов.
- 3) Диагноз выполняется при условии, что только один компонент может быть неисправным.
- 4) После восстановления работоспособности процедура тестирования начинается всегда с верхнего уровня иерархии к нижнему – нисходящее диагностирование.
- 5) Процедура тестирования может начинаться с любого уровня иерархии и заканчиваться на уровне, желаемом разработчиком.

Предложенная модель тестирования дает возможность обнаруживать одиночный дефект и, по меньшей мере, один из кратных, присутствующий в системе [11,24].

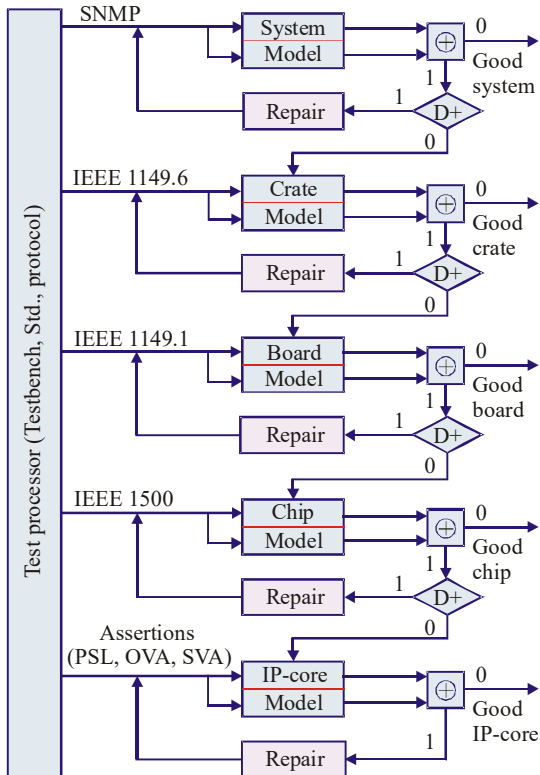


Рис. 6. Структура тестирования цифровой системы

4. Заключение

Рассмотрены международные стандарты общества IEEE, пригодные для тестирования и диагностирования программно-технических комплексов и электронных цифровых систем широкого профиля: начиная от функциональных компонентов внутри интегральной схемы и заканчивая сложными системами, состоящими из печатных плат, крейтов.

Рассмотренные стандарты IEEE Std 1149.1, 1149.4, 1149.6 и 1500 и технические предложения SJTAG предоставляют широкие возможности для решения задач: тестирование функциональности компонента; тестирование соединений на печатной плате; тестирование функционирования системы на печатной плате; тестирование системы, состоящей из печатных плат, крейтов.

Объектами тестирования являются: аналоговые и цифровые модули; компоненты смешанного типа; дискретные элементы; печатные платы и крейты, состоящие из перечисленных выше компонентов.

Научная новизна полученных результатов заключается в разработке ad hoc технологий, моделей и методов тестирования, объединяющих существующие IEEE стандарты со специальными схемотехническими решениями, позволяющими

в несколько раз уменьшить время диагностирования сложных цифровых систем управления критическими технологиями [5-7,21-23]:

1) Представлены аналитическая и структурная модели диагностирования сложных цифровых систем, ориентированные на выполнение тестового эксперимента в реальном масштабе времени с использованием средств моделирования неисправностей, которые дают возможность точного установления диагноза за несколько секунд.

2) Разработан блочно-иерархический метод тестирования и диагностирования программно-технических комплексов на основе представления сложной системы в виде иерархии более простых блоков, для которой разработана сеть контроллеров граничного сканирования, дающая возможность управлять процессом тестирования сложных объектов в реальном масштабе времени и существенно (на 50-70%) уменьшать время проектирования и верификации цифровых изделий.

3) Предложен метод диагностирования дефектов в сложных цифровых системах на основе использования совокупности таблиц неисправностей и процедуры ее теоретико-множественного анализа, который предполагает разбиение объекта на четыре уровня иерархии, что существенно (в десятки и сотни раз) уменьшает объем диагностической информации.

Таким образом, технологии граничного сканирования позволяют разработчикам быть на уровне современных требований и тенденций при проектировании, верификации, отладке, изготовлении, тестировании, внутрисхемном программировании и эксплуатации различных систем. Соответствие конечного программного и/или аппаратного продукта международным стандартам IEEE – залог его успешного продвижения на мировом рынке.

Практическая значимость заключается в существенном (на 50-70%) уменьшении времени поиска дефектов в сложной цифровой системе на основе использования предложенной иерархической модели процесса диагностирования и специальных технологий граничного сканирования IEEE стандартов проектирования [8-14,21-23].

Список литературы: 1. *Grant Martin*. The Reuse of Complex Architectures. IEEE Design and Test of Computers. November-December 2002. P. 4-6. 2. *Rajesh K. Gupta and Yervant Zorian*. Introducing Core-Based System Design. IEEE Design & Test of Computers. November-December 1997. P. 15-25. 3. *Yervant Zorian*. What is Infrastructure IP? IEEE Design & Test of Computers. May-June 2002. P. 5-7. 4. *L. Benini and G. D. Micheli*. Networks on chips: A new soc paradigm. IEEE Computer. 2002. Vol. 35, No. 1. P. 70-78. 5. *Ястребенецкий М.А., Васильченко В.М. и др.* Безопасность атомных станций. Информационные и управляющие системы /Под ред. М.А. Ястребенецкого. К.:Техніка. 2004. 472 с. 6. *Горелик А.Х., Елисеев В.В., Орловский В.А.* Опыт разработки новых и поэтапной реконструкции действующих информационно-вычислительной системы энергоблоков с реактором ВВЭР-1000 // Ядерная и радиационная безопасность. 2005. №1. С. 91-96. 7. *Елисеев В.В., Ларгин В.А., Пивоваров Г.Ю.* Программно-технические комплексы АСУ ТП. Учебное пособие. Киев: Издательско-полиграфический центр «Киевский университет». 2003. 429 с. 8. IEEE P1500/D11. January 2005. Draft Standard Testability Method for Embedded Core-based Integrated Circuits. New York. 2005. 138 p. 9. *Yervant Zorian*. Advances in Infrastructure IP. IEEE Design & Test of Computers. May-June 2003. P. 49-56. 10. *Yervant Zorian*. Test Requirements for Embedded Core-Based Systems and IEEE P1500. In Proceedings IEEE International Test Conference (ITC). Washington, DC. November 1997. IEEE Computer Society Press. P. 191-199. 11. *Abramovici M., Breuer M.A. and Friedman A.D.* Digital systems testing and testable design. Computer Science Press. 1998. 652 p. 12. IEEE Std 1149.1-2001. Standard Test Access Port and Boundary-Scan Architecture. New York. 2001. 208 p. 13. IEEE Std 1149.4-1999. IEEE Standard for a Mixed-Signal Test Bus. New York., 2000. 84 p. 14. IEEE Std 1149.6-2003. Standard for Boundary-Scan Testing of Advanced Digital Networks. New York. 2003. 139 p. 15. *S.Park, T.Kim*. A new IEEE 1149.1 boundary scan design for the detection of delay defects.—*Proc. DATE'2000*. P. 458-462. 16. *Stephen Pateras*. IP for Embedded Diagnosis. IEEE Design and Test of Computers. 2002. P. 46-56. 17. *Hemmady S.G., Anderson T.L., Zorian Y.* Verification and Testing of Embedded Cores. Proc. Design SuperCon. On-Chip Design. 1997. P 119-122. 18. *Yervant Zorian, Erik Jan Marinissen, and Sujit Dey*. Testing Embedded Core-Based System Chips. In Proceedings IEEE International Test Conference (ITC). Washington. 1998. IEEE Computer Society Press. P. 130-143. 19. *Yervant Zorian*. Advances in Infrastructure IP. IEEE Design & Test of Computers. 2003. P. 49. 20. *Harry Bleeker, Peter van den Eijnden, and Frans de Jong*. Boundary-Scan Test – A Practical Approach. Kluwer Academic Publishers. Dordrecht. The Netherlands. 1993. 264 p. 21. *Alexandr Yegorov, Irina Hahanova, Olga Melnikova, Vladimir Yeliseyev*. Eda-Market and Soc Design Verification Technologies. Proceedings of the 8th International Conference CADSM 2005. Lviv, Ukraine. 2005. P. 352-355. 22. *V. Hahanov, V. Obrizan, V. Yeliseev, V. Ghribi*. Hierarchical testing of complex digital systems. Proceedings of the International Conference TCSET 2006.- Lviv, Ukraine. 2006. P. 426-429. 23. *Хаханов В.И., Елисеев В.В., Обризан В.И., Ваде Гриби, Хасан Ктиаман*. Ассерт-метод верификации цифровых систем на основе стандарта IEEE 1500 SECT // АСУ и приборы автоматки. 2005. Вып. 132. С. 93-105. 24. *Автоматизация диагностирования электронных устройств / Ю.В.Малышенко и др./ Под ред. В.П. Чипулиса. М.: Энергоатомиздат. 1986. 216с. 25. Беннеттс Р. Дж. Проектирование тестопригодных логических схем. М.: Радио и связь. 1990. 175 с.*

Поступила в редколлегию 11.12.2006

Елисеев Владимир Васильевич, директор ЗАО "Северодонецкое НПО "Импульс", доцент Северодонецкого технологического института, канд. техн. наук. Научные интересы: компьютерная инженерия, программно-технические комплексы систем контроля и управления. Адрес: Украина, 93405, Северодонецк Луганской обл., пл. Победы, 2, тел. (06452) 2-95-87.

Хаханов Владимир Иванович, д-р техн. наук, профессор кафедры АПВТ ХНУРЭ. Научные интересы: техническая диагностика вычислительных устройств, систем, сетей и программных продуктов. Увлечения: баскетбол, футбол, горные лыжи. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-326. E-mail: hahanov@kture.kharkov.ua

Парфентий Александр Николаевич, аспирант кафедры АПВТ ХНУРЭ. Научные интересы: техническая диагностика цифровых устройств. Увлечения: английский язык, автопутешествия. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-326.

Хаханова Анна Владимировна, аспирантка кафедры АПВТ ХНУРЭ. Научные интересы: техническая диагностика цифровых устройств. Увлечения: аэробика, музыка, автопутешествия. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-326.