

СРЕДСТВА ДИАГНОСТИЧЕСКОГО ОБСЛУЖИВАНИЯ SoC

Предлагается инфраструктура сервисного обслуживания функциональностей SoC, которая отличается минимальным набором процессов встроенного диагностирования в реальном масштабе времени и дает возможность осуществлять сервисы: тестирование функциональностей на основе генерируемых входных последовательностей и анализ выходных реакций; диагностирование с заданной глубиной поиска дефектов в SoC; моделирование неисправностей в целях выполнения первых двух процедур на основе таблицы неисправностей. Описывается структурно-алгебраический метод и алгоритм встроенного диагностирования дефектов в функциональных блоках SoC, использующий предварительный анализ таблицы неисправностей в целях уменьшения ее объема и последующих вычислений, связанных с построением ДНФ, которая формирует все решения по установлению диагноза функциональностей SoC в реальном масштабе времени.

1. Инфраструктура I-IP

Высокая плотность вентиля в кристалле SoC и сложность реализуемых алгоритмов создает определенные проблемы в их проектировании и тестировании на уровне RTL. Поиск новых методов и средств [1-16] требует повышения уровней абстракции создаваемых моделей встраиваемых процессов сервисного обслуживания. Рынок программных продуктов EDA предлагает разнообразие инструментов для самого неавтоматического системного уровня проектирования устройств, начиная со специальных языков программирования (C++, SystemC, SystemVerilog, UML, SDL) [12] и заканчивая графическими оболочками (Simulink, LabView, Xilinx EDK), которые позволяют создавать проекты из библиотечных компонентов путем использования технологий ESL, TLM [13, 14]. Кроме того, выбор в качестве технологической и элементной базы цифровой системы на кристалле FPGA позволяет избежать применения дорогостоящего универсального процессора, существенно уменьшить потребляемую мощность, габаритные размеры, стоимость с одновременным повышением быстродействия при качественном выполнении основных функций, что является актуальным в век мобильных вычислительных устройств.

Современные технологии проектирования цифровых систем на кристаллах предполагают, наряду с созданием функциональных блоков, разработку сервисных модулей ориентированных на комплексное решение проблемы качества проекта и повышение выхода годной продукции в процессе производства, которое определяется внедрением в кристалл следующих сервисов [13]:

1) Наблюдение за состоянием внутренних и выходных линий в процессе функционирования, верификации и тестирования штатных блоков на основе использования стандарта граничного сканирования IEEE 1500 [15].

2) Тестирование функциональных модулей путем подачи проверяющих наборов от различных тестовых генераторов, ориентированных на проверку дефектов или исправного поведения.

3) Диагностирование отказов и дефектов путем анализа информации, полученной на стадии тестирования и использования специальных методов встроенного поиска неисправностей на основе стандарта IEEE 1500.

4) Восстановление работоспособности функциональных модулей после фиксации отрицательного результата тестирования и определения места и вида дефекта при выполнении фазы диагностирования.

5) Измерение основных характеристик и определение параметров функционирования изделия на основе встроенных средств, позволяющих производить временные и вольт-амперные измерения.

6) Надежность и отказоустойчивость функционирования изделия в процессе эксплуатации, которая достигается диверсификацией функциональных блоков, их дублированием и восстановлением работоспособности SoC в реальном масштабе времени.

Используя стандартизированные компоненты сервисного обслуживания, далее предлагается усеченная структурно-функциональная модель, которая ориентирована на выполнение следующих трех задач (рис. 1): 1) тестирование функциональностей на основе генерируемых входных последовательностей (Automated Test Pattern Generator) и анализ выходных реакций; 2) диагностирование дефектов с заданной глубиной путем использования мультizonда стандарта IEEE 1500; 3) моделирование (Fault Simulator) неисправностей в целях обеспечения выполнения первых двух процедур на основе таблицы неисправностей (Fault Detection Table).

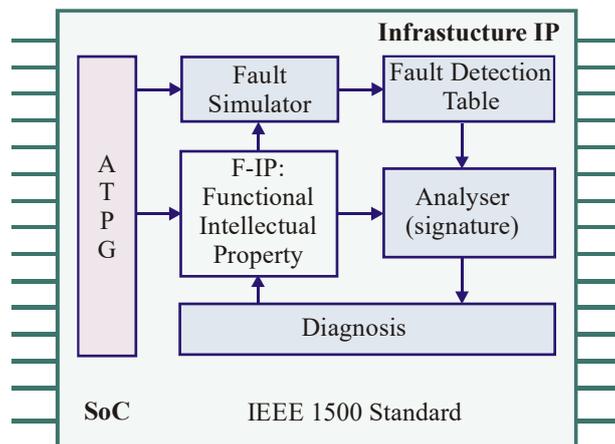


Рис. 1. Инфраструктура сервисов SoC DSP

1. Модуль синтеза тестов для проверки функциональностей и одиночных неисправностей состоит из набора генераторов входных последовательностей, которые обеспечивают синтез следующих тестов: PRTG – псевдослучайный генератор входных стимулов с равномерным законом распределения нулевых и единичных сигналов по входным переменным; SATG – тестовый генератор шестнадцатеричных кодов на основе сигнатурного анализа; SPTG – алгоритмический генератор тестовых векторов, активизирующих одномерные логические пути, ориентированные на проверку заданных одиночных неисправностей; ADTG – тестовый генератор, предназначенный для проверки сумматорных схем АЛУ; BSTG – тестовый генератор для шинных структур приема и передачи данных; METG – генератор тестов, ориентированный на проверку матричной памяти; DFTG – синтезатор тестов для автоматов, заданных в виде граф-схем алгоритмов; RCTG – тестовый генератор для последовательностных счетно-регистровых структур и триггерных схем.

Модуль-генератор анализирует структурно-функциональную модель блока, подлежащего тестированию, и назначает подмножество таких синтезаторов, которые обеспечивают заданное качество покрытия дефектов (F^c) и функциональных режимов (P^c):

$$F^c \left(\bigcup_{i=1}^{n_{\min}} T_i \right) \geq F_{\min}^c ; P^c \left(\bigcup_{i=1}^{n_{\min}} T_i \right) \geq P_{\min}^c , \quad (1)$$

$$T = \{ T_1^{PR}, T_2^{SA}, T_3^{SP}, T_4^{AD}, T_5^{BS}, T_6^{ME}, T_7^{DF}, T_8^{RC} \}.$$

Обобщенная структура синтеза Testbench, представленная на рис. 2, включает также генератор HDL-кода, который предназначен для тестирования и верификации функциональностей на стадии разработки проекта.

Количество тестовых генераторов на стадии проектирования SoC может быть существенно большим, чем то подмножество, которое далее встраивается в кристалл. Поэтому на стадии моделирования и верификации выполняется анализ покрывающих свойств каждого тест-генератора в целях поиска их минимальной совокупной конфигурации, которая удовлетворяет выражению (1). Важно отметить, что в ближайшие 5 лет идеология синтеза тестов для цифровых систем на кристаллах будет заимствовать лучшие традиции, идущие от ESL-, TLM-проектирования [16].

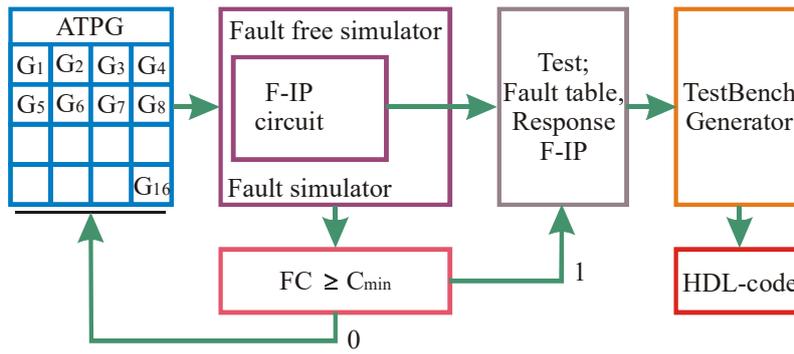


Рис. 2. Структура процесса синтеза Testbench для F-IP

Это означает: 1) Использование библиотек тестов (Testbench) ведущих компаний планеты для тестирования и верификации стандартизованных функциональностей, обозначенных в качестве F-IP. 2) Применение стандартных решений сервисного обслуживания I-IP для встроенного тестирования компонентов системы на кристалле. 3) Создание собственных библиотек тестов для вновь разрабатываемых функциональностей. 4) Внедрение новой технологии синтеза тестов для цифровой системы, основанной на дискретном мэппинге (рис. 3) покрытия функциональностей и дефектов исходной спецификации с помощью минимальной совокупности Testbench, из библиотеки тестов. 5) Применение встроенных средств тестопригодности, таких как IEEE boundary scan – средства граничного сканирования и шесть компонентов I-IP для повышения технологичности процедур синтеза тестов.

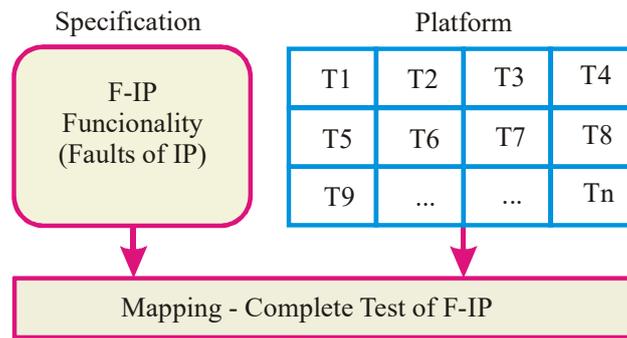


Рис. 3. Mapping модели синтеза тестов для F-IP

2. Модуль анализа неисправностей. Использует дедуктивный алгоритм, который ориентирован на проверку одиночных дефектов, генерируемых на основе аналитического или табличного описания функциональностей SoC. Это означает, что дедуктивное моделирование может обрабатывать проекты, представленные как на вентиляльном, так и на любом другом, более высоком уровне абстракции (регистровом, системном). Основная идея метода заключается в создании дедуктивной модели функциональности на основе использования следующей формулы:

$$F = f[(X_1 \oplus T_1), (X_2 \oplus T_2), \dots, (X_j \oplus T_j), \dots, (X_{n_i} \oplus T_{n_i})] \oplus T_i, \quad (2)$$

где дедуктивная функция F на тест-векторе T есть модифицированная функция исправного поведения, позволяющая вычислять списки входных неисправностей, которые транспортируются на выход компонента схемы под воздействием входных сигналов. На примере функции Хог можно продемонстрировать синтез дедуктивной функции по карте Карно:

$$F = \begin{array}{c|cccc} (xy) \backslash (ab) & 00 & 01 & 11 & 10 \\ \hline 00 & 0 & 1 & 0 & 1 \\ 01 & 0 & 1 & 0 & 1 \\ 11 & 0 & 1 & 0 & 1 \\ 10 & 0 & 1 & 0 & 1 \end{array} \quad (3)$$

Переменные x, y – булевы, а сигналы ab – (регистровые) списки дефектов:

$$L = f(x, y, a, b) = \bar{a}b \vee a\bar{b}. \quad (4)$$

Аппаратная реализация дедуктивной функции, представленной формулой (2), имеет вид, представленный на рис. 4.

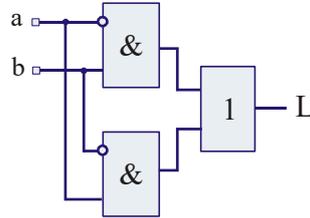


Рис. 4. Дедуктивный примитив функции Хор

Схемный примитив является универсальным по отношению к различным тестовым последовательностям. Стратегия, предложенная в работе относительно синтеза моделей, основывается на создании библиотеки дедуктивных элементов, покрывающих все стандартизованные конструктивы функциональностей, которыми оперирует разработчик, создавая в автоматизированном режиме проект в виде цифровой системы на кристалле. В данном случае речь идет о синтезе дедуктивной структуры на основе мэппинга [16], суть которого представлена на рис. 5.

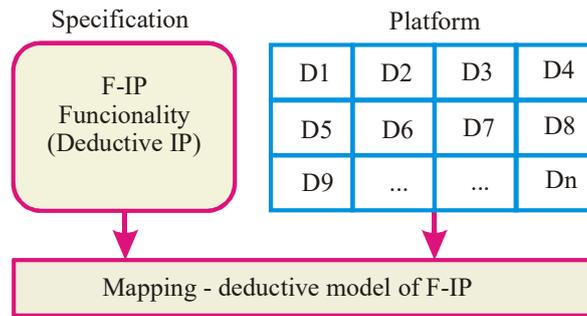


Рис. 5. Mapping дедуктивной модели для F-IP

Предложенный подход к дедуктивному анализу предполагает создание на кристалле еще одной встроенной модели, которая должна обеспечивать практически все шесть сервисов, предусмотренных стандартом инфраструктуры I-IP [13]. Платой за качество диагностического и тестового обслуживания является достаточно высокая стоимость дополнительных аппаратных затрат, которые превышают штатную функциональность в 10-15 раз. При этом выигрыш в быстродействии по сравнению с внешней программной реализацией дедуктивного анализа составляет 2-3 порядка, что практически обеспечивает сервисное обслуживание в реальном масштабе времени. Другое, более экономичное, решение проблемы связано с интерактивной модификацией схемной структуры дедуктивной модели для каждого тест-вектора. Для этого используется внутренняя память кристалла, где формируется модель по правилам, определенным в (2). Мэппинг (см. рис. 5) дает дедуктивную функцию, где аппаратные затраты равны стоимости функциональности F-IP.

2. Структурно-алгебраический метод диагностирования неисправностей

Основная роль отводится технологии граничного сканирования [15], которая в настоящее время, внедренная в кристалл, призвана облегчать решение практически всех задач сервисного обслуживания функциональных модулей системы на кристалле. Контроллер доступа к внутренним линиям и портам регистра граничного сканирования использует ячейку или разряд регистра. В совокупности число таких ячеек, обеспечивающих в данном случае мониторинг, должно быть равно количеству проблемных наблюдаемых линий, которые необходимы для точного установления диагноза. Основанная на регистре граничного сканирования процедура диагностирования использует также информацию из таблицы неисправностей (ТН), которая представляет собой множество дефектов, покрытых тесто-

выми наборами. Используя информацию о результате проведения диагностического эксперимента, которая представлена в виде вектора экспериментальной проверки V , а также таблицу [9] неисправностей F , выполняют процедуру установления диагноза по выражению, записанному в форме произведения дизъюнкций всех дефектов [10], которые могут дать экспериментальную реакцию, полученную в виде вектора экспериментальной проверки V , определенного единичными и нулевыми значениями:

$$F = \bigwedge_{\forall V_i=1}^{i=1, \overline{n}} \left(\bigvee_{\forall M_{ij}=1}^{j=1, \overline{m}} F_j \right). \quad (5)$$

Полученная из таблицы неисправностей конъюнктивная нормальная форма трансформируется к ДНФ с помощью эквивалентных преобразований (логическое умножение, склеивание и поглощение) [11]. В результате получается булева функция, где термы – логические произведения представляют полное множество решений в виде сочетания дефектов (дающих по выходам SoC или ее компоненту вектор экспериментальной проверки V):

$$F = \bigwedge_{\forall V_i=1}^{i=1, \overline{n}} \left(\bigvee_{\forall M_{ij}=1}^{j=1, \overline{m}} F_j \right) = \left[\begin{array}{l} a \vee ab = b \\ a \vee a = a \end{array} \right] = \bigvee_{i=1}^{2^m} \left(\bigwedge_{j=1}^m k_j F_j \right), k_j = \{0,1\}. \quad (6)$$

Представленная процедура в общем случае диагностирует некоторое подмножество дефектов, которое в дальнейшем нуждается в уточнении путем применения дополнительного зондирования внутренних точек с помощью регистра граничного сканирования. Пример поиска дефектов рассматривается на основе заданной таблицы неисправностей (столбцы – дефекты, строки – тестовые последовательности), которая является продуктом дедуктивного анализа дефектов и вектора экспериментальной проверки:

$$M = \begin{array}{c|cccccc|c} T_i \backslash F_j & F_1 & F_2 & F_3 & F_4 & F_5 & F_6 & V \\ \hline T_1 & 1 & & & 1 & & & 1 \\ T_2 & & 1 & & & 1 & & 1 \\ T_3 & & & 1 & 1 & 1 & & 0 \\ T_4 & 1 & & 1 & & & & 1 \\ T_5 & & 1 & & & 1 & 1 & 1 \end{array}$$

Количество единиц (6) в векторе экспериментальной проверки V формирует число дизъюнктивных термов КНФ. Каждый терм – построчная запись дефектов (через логическую операцию ИЛИ), оказывающих влияние на выходы функциональности. Уже само представление таблицы в виде аналитической записи – конъюнктивной нормальной формы дает потенциальную возможность существенно сократить объем диагностической информации для поиска дефектов. Тем более, последующее преобразование КНФ к ДНФ на основе тождеств алгебры логики позволяет существенно уменьшить булеву функцию, что иллюстрируется следующим результатом:

$$\begin{aligned} F &= (F_1 \vee F_4)(F_2 \vee F_5)(F_3 \vee F_4 \vee F_5)(F_1 \vee F_3)(F_2 \vee F_5 \vee F_6) = \\ &= (F_1 \vee F_4)(F_2 \vee F_5)(F_3 \vee F_4 \vee F_5)(F_1 \vee F_3) = \\ &= (F_1 F_2 \vee F_2 F_4 \vee F_1 F_5 \vee F_4 F_5)(F_1 F_3 \vee F_1 F_4 \vee F_1 F_5 \vee F_3 F_3 \vee F_3 F_4 \vee F_3 F_5) = \\ &= (F_1 F_2 \vee F_2 F_4 \vee F_1 F_5 \vee F_4 F_5)(F_1 F_4 \vee F_1 F_5 \vee F_3) = \\ &= (F_1 F_2 F_1 F_4 \vee F_2 F_4 F_1 F_4 \vee F_1 F_5 F_1 F_4 \vee F_4 F_5 F_1 F_4) \\ &\quad (F_1 F_2 F_1 F_5 \vee F_2 F_4 F_1 F_5 \vee F_1 F_5 F_1 F_5 \vee F_4 F_5 F_1 F_5) \\ &\quad (F_1 F_2 F_3 \vee F_2 F_4 F_3 \vee F_1 F_5 F_3 \vee F_4 F_5 F_3) = \\ &= F_1 F_2 F_3 \vee F_2 F_3 F_4 \vee F_1 F_3 F_5 \vee F_3 F_4 F_5 \vee F_1 F_2 F_4 \vee F_1 F_4 F_5 \vee F_1 F_5 = \\ &= F_1 F_5 \vee F_1 F_2 F_3 \vee F_2 F_3 F_4 \vee F_3 F_4 F_5 \vee F_1 F_2 F_4. \end{aligned} \quad (7)$$

Для уменьшения количества вычислений при выполнении логического умножения в первой строке (7), согласно законам алгебры логики, выполнено преобразование:

$$(F_2 \vee F_5) \wedge (F_2 \vee F_5 \vee F_6) = (F_2 \vee F_5).$$

Полученный результат $F = F_1F_5 \vee F_1F_2F_3 \vee F_2F_3F_4 \vee F_3F_4F_5 \vee F_1F_2F_4$ предоставляет все возможные решения – покрытия дефектами всех строк таблицы неисправностей в функциональности SoC, при условии, что вектор экспериментальной проверки имеет все единичные координаты $V = (11111)$. Принимая во внимание фактическое значение вектора экспериментальной проверки, равное $V = (11011)$, выполняется моделирование функции F путем подстановки нулевых значений дефектов, которые теоретически проверяются, но дают в векторе V нулевую координату. Такими являются дефекты: $\bar{F} = F_3 \vee F_4 \vee F_5$. Окончательный результат определяется следующей функцией:

$$F = F_1F_5 \vee F_1F_2F_3 \vee F_2F_3F_4 \vee F_3F_4F_5 \vee F_1F_2F_4 \Big|_{F_3 \vee F_4 \vee F_5 = 0} = 0.$$

Предложенный метод диагностирования отличается от известных компактной записью в виде ДНФ, достаточно объемной таблицы неисправностей. Кроме того, процесс установления диагноза сводится к технологичной и быстродействующей, встроенной в кристалл, процедуре моделирования термов при заданных начальных условиях – нулевых значениях переменных – невозможных дефектов цифровой системы на кристаллах. Метод позволяет находить дефекты любой кратности или фиксировать факт невозможности наличия в схеме любого сочетания, из приведенных в ТН.

Любое сочетание – конъюнктивный терм ДНФ, приведенный в решении

$$F = F_1F_5 \vee F_1F_2F_3 \vee F_2F_3F_4 \vee F_3F_4F_5 \vee F_1F_2F_4,$$

покрывает все строки таблицы неисправностей по определению, поэтому введение любой нулевой строки обязательно обращает в ноль функцию F . Поэтому корректное решение, соответствующее вектору экспериментальной проверки, должно изначально учитывать нулевые координаты вектора V . С учетом сказанного из выражения (7) на стадии записи КНФ необходимо исключить терм $(F_3 \vee F_4 \vee F_5)$:

$$\begin{aligned} F &= (F_1 \vee F_4)(F_2 \vee F_5)(F_1 \vee F_3)(F_2 \vee F_5 \vee F_6) = \\ &= (F_1 \vee F_4)(F_2 \vee F_5)(F_1 \vee F_3) = \\ &= (F_1F_2 \vee F_2F_4 \vee F_1F_5 \vee F_4F_5)(F_1 \vee F_3) = \\ &= F_1F_2 \vee F_1F_2F_4 \vee F_1F_5 \vee F_1F_4F_5 \vee F_1F_2F_3 \vee F_2F_3F_4 \vee F_1F_3F_5 \vee F_3F_4F_5 = \\ &= F_1F_2 \vee F_1F_5 \vee F_2F_3F_4 \vee F_3F_4F_5. \end{aligned} \quad (8)$$

Результат есть все возможные решения, которые приводят к реакции изделия, определенного заданным вектором экспериментальной проверки:

$$F = F_1F_2 \vee F_1F_5 \vee F_2F_3F_4 \vee F_3F_4F_5.$$

Дополнительное моделирование последней булевой функции дает окончательный результат в виде сочетания двух дефектов:

$$F = F_1F_2 \vee F_1F_5 \vee F_2F_3F_4 \vee F_3F_4F_5 \Big|_{F_3 \vee F_4 \vee F_5 = 0} = F_1F_2.$$

Алгоритм предварительного анализа таблицы неисправностей M_1 может быть рассмотрен на примере следующей таблицы неисправностей и представлен в виде пяти пунктов:

$$M_1 =$$

T_i / F_j	F_1	F_2	F_3	F_4	F_5	F_6	V
T_1	1			1			1
T_2		1			1		1
T_3			1	1		1	1
T_4	1		1				1
T_5		1			1	1	0

1. Определяются все строки, соответствующие нулевому значению вектора экспериментальной проверки в целях обнуления всех единичных координат. В данном случае – это одна строка T_5 .

2. Находятся все столбцы, которые имеют нулевые значения координат строки с нулевым состоянием ВЭП. Единичные значения найденных столбцов обнуляются. В данном случае: F_2, F_5, F_6 .

3. Строки и столбцы, имеющие только нулевые значения координат (найденные в пунктах 1 и 2), удаляются из таблицы неисправностей:

$$M_1 = \begin{array}{c|ccccccc} T_i \backslash F_j & F_1 & F_2 & F_3 & F_4 & F_5 & F_6 & V \\ \hline T_1 & 1 & & & 1 & & & 1 \\ T_2 & & 0 & & & 0 & & 1 \\ T_3 & & & 1 & 1 & & 0 & 1 \\ T_4 & 1 & & 1 & & & & 1 \\ T_5 & & 0 & & & 0 & 0 & 0 \end{array} = \begin{array}{c|cccc} T_i \backslash F_j & F_1 & F_3 & F_4 & V \\ \hline T_1 & 1 & & 1 & 1 \\ T_3 & & 1 & 1 & 1 \\ T_4 & 1 & 1 & & 1 \end{array}$$

4. Выполняется построение КНФ по единичным значениям ВЭП:

$$\begin{aligned} F &= (F_1 \vee F_4) \wedge (F_3 \vee F_4) \wedge (F_1 \vee F_3) = \\ &= (F_1 F_3 \vee F_3 F_4 \vee F_1 F_4 \vee F_4 F_4) \wedge (F_1 \vee F_3) = \\ &= F_1 F_1 F_3 \vee F_1 F_3 F_4 \vee F_1 F_1 F_4 \vee F_1 F_4 F_4 \vee F_1 F_3 F_3 \vee F_3 F_3 F_4 \vee F_1 F_3 F_4 \vee F_3 F_4 F_4 = \\ &= F_1 F_3 \vee F_1 F_3 F_4 \vee F_1 F_4 \vee F_3 F_4 \vee F_1 F_3 F_4 \vee F_3 F_4 = \\ &= F_1 F_3 \vee F_1 F_4 \vee F_3 F_4. \end{aligned}$$

5. Преобразование КНФ к ДНФ с последующей минимизацией термов приводит к получению искомого результата в виде сочетания неисправностей:

$$F = F_1 F_3 \vee F_1 F_4 \vee F_3 F_4.$$

Предложенный алгоритм ориентирован на предварительный анализ таблицы неисправностей в целях уменьшения ее объема и последующих вычислений, связанных с построением ДНФ, которая формирует все решения по установлению диагноза функциональностей SoC. Дальнейшее уточнение диагноза возможно только с применением мультитонда на основе регистра граничного сканирования данных [15].

3. Выводы

Научная новизна и практическая значимость: 1) Предложен структурно-алгебраический метод и алгоритм встроенного диагностирования дефектов в функциональных блоках SoC, использующий предварительный анализ таблицы неисправностей в целях уменьшения ее объема и последующих вычислений, связанных с построением ДНФ, которая формирует все решения по установлению диагноза функциональностей SoC в реальном масштабе времени. Дальнейшее уточнение диагноза возможно только с применением мультитонда на основе регистра граничного сканирования. 2) Предложена усовершенствованная инфраструктура сервисного обслуживания функциональностей SoC, которая отличается минимальным набором процессов встроенного диагностирования в реальном масштабе времени и дает возможность осуществлять сервисы: тестирование штатных функций на основе генерируемых входных последовательностей (Automated Test Pattern Generator) и анализ выходных реакций; диагностирование с заданной глубиной поиска дефектов путем использования мультитонда стандарта IEEE 1500; моделирование (Fault Simulator) неисправностей в целях обеспечения выполнения первых двух процедур на основе таблицы неисправностей (Fault Detection Table). 3) Предложена мэппинг-модель процесса синтеза дедуктивной структуры, которая отличается использованием библиотеки дедуктивных компонентов, покрывающих все стандартизированные конструктивы функциональностей, которыми оперирует разработчик, что дает возможность создавать в автоматизированном режиме дедуктивную модель функциональностей цифровой системы на кристалле.

4) Предложена мэппинг-модель процесса синтеза тестов, которая отличается использованием библиотеки встроенных тестовых генераторов для функциональностей DSP SoC, что дает возможность существенно уменьшить время построения тестов, предназначенных для верификации функциональных модулей и проверки дефектов.

Список литературы: 1. *Bergeron J.* Writing testbenches: functional verification of HDL models. Springer, 2003, 512p. 2. *Rashinkar P., Paterson P., Singh L.* System-on-chip Verification: Methodology and Techniques, Kluwer Academic Publishers, 2002. 393p. 3. *IEEE-1800* IEEE Standard for System Verilog Language, 2005. 586p. 4. *S. Hamdioui, G. N. Gaydadjiev, A. J. Van de Goor.* The State-of-the-art and Future Trends in Testing Embedded Memories // Records IEEE International Workshop on Memory Technology, Design, and Testing, San Jose, CA, August 2004. P. 54-59. 5. *Zhong Y., Dropsho S.G., Shen X., Studer A., Ding C.* Miss Rate Prediction Across Program Inputs and Cache Configurations // IEEE Transactions on Computers. 2007. P.328-343. 6. *Shoukourian S., Vardanian V., Zorian Y.* SoC Yield Optimization via an Embedded-Memory Test and Repair Infrastructure // IEEE Design and Test of Computers. 2004. P.200-207. 7. *YoungsL., Paramanandam S.* Mapping and Repairing Embedded-Memory Defects // IEEE Design and Test of Computers. – 1997. P. 18-24. 8. *Zorian Y., Shoukourian S.* Embedded-Memory Test and Repair: Infrastructure IP for SoC Yield // IEEE Design and Test of Computers. 2003. P.58-66. 9. *Hahanov V., Kteaman H., Ghribi W., Fomina E.* HEDEFS – Hardware embedded deductive fault simulation. Proceedings volume from the 3rd IFAC Workshop, Rydzyna, Poland, 2006. P. 25-29. 10. *Rossen K.* Discrete Mathematics and its Applications, McGraw Hill, 2003. 824p. 11. *Парфентий А.Н., Хаханов В.И., Лутвинова Е.И.* Модели инфраструктуры сервисного обслуживания цифровых систем на кристаллах // АСУ и приборы автоматики. 2007. Вып. 138.С.83-99. 12. *Хаханов В.И., Хаханова И.В.* VHDL + Verilog = Синтез за минуты. Харьков: СМИТ. 2007. 264 с. 13. *Zorian Yervant* What is Infrastructure IP? // IEEE Design & Test of Computers. May-June 2002. P. 5-7. 14. *Zorian Yervant, Gizopoulos Dmytris.* Gest editors' introduction: Design for Yield and reliability // IEEE Design & Test of Computers. May-June 2004. P. 177-182. 15. *IEEE 1500* Web Site. <http://grouper.ieee.org/groups/1500>. 16. *Douglas Densmore, Roberto Passerone, Alberto Sangiovanni-Vincentelli.* A Platform-Based taxonomy for ESL design, Design&Test of computers, september-october 2006. P. 359-373.

Поступила в редколлегию 16.08.2007

Хаханов Владимир Иванович, декан факультета КИУ ХНУРЭ, д-р техн. наук, профессор кафедры АПВТ ХНУРЭ. Научные интересы: техническая диагностика цифровых 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.

Хаханова Ирина Витальевна, доцент кафедры АПВТ ХНУРЭ. Научные интересы: проектирование и диагностика цифровых систем на кристаллах. Увлечения: музыка, иностранные языки. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-326. E-mail: hahanova@mail.ru

Каминская Марина Александровна, аспирант кафедры АПВТ ХНУРЭ. Научные интересы: техническая и медицинская диагностика, тестопригодное проектирование. Увлечения: история, литература, музыка, психология, педагогика. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 702-13-26. E-mail: maryna4329@kture.kharkov.ua

УДК 519.713

О.А. КРИВОДУБСКИЙ, С.А. КОСИЛОВ

КЛАССИФИКАЦИЯ ПЕРЕМЕННЫХ И РАСЧЁТ ПОКАЗАТЕЛЕЙ ПОРТФЕЛЯ ЗАКАЗОВ

Рассматривается задача планирования листопркатного производства, связанная с определением оптимальной очередности выполнения отдельных позиций портфеля заказов. Исследуется обобщённая структура объекта управления – листопркатного цеха ОАО «ДМЗ», выполняется классификация входящей информации и её формализация в виде теоретико-множественных конструкций. Устанавливаются и формализуются взаимосвязи между характеристиками объекта, что позволяет заменить исследование характеристик объекта управления изучением отношений между множествами. Формализованная модель объекта управления предназначена для алгоритмов оптимального планирования выпуска проката.