

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ

ХАРЬКОВСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ
РАДИОЭЛЕКТРОНИКИ

ISSN 1563-0064

РАДИОЭЛЕКТРОНИКА И ИНФОРМАТИКА

Научно-технический журнал

Основан в 1997 г.

№ 3(62), июль – сентябрь 2013

Выходит 4 раза в год

© Харьковский национальный
университет радиоэлектроники, 2013

Свидетельство о государственной регистрации КВ № 12097-968 ПР 14.12.2006

РИ, 2013, № 3

СОДЕРЖАНИЕ

РАДИОТЕХНИКА

ПОЗДНЯКОВ Е.К., ТКАЧЕНКО В.Н., КОРОТКОВ В.В. УВЕЛИЧЕНИЕ ТОЧНОСТИ ОПРЕДЕЛЕНИЯ КООРДИНАТ ПАССИВНЫМИ ТРЕХПОЗИЦИОННЫМИ КОМПЛЕКСАМИ В УСЛОВИЯХ ИЗБЫТОЧНОСТИ	3
ГАВРИШ О.С., ЗАБОЛОТНИЙ С.В., БУРДУКОВА О.В. МЕТОД АНАЛИЗА АСИМПТОТИЧНЫХ ВЛАСТИВОСТЕЙ АДАПТИВНЫХ АЛГОРИТМОВ ВИМІРЮВАННЯ ПАРАМЕТРІВ РАДІОСИГНАЛУ ПРИ ЕКСЦЕСІЙНІЙ ЗАВАДІ З ВИКОРИСТАННЯМ НАВЧАЛЬНОЇ ВИБІРКИ	7
ЧЕРНЫШОВ Н.Н. КОМПЬЮТЕРНЫЕ ПРОГРАММЫ ДЛЯ НЕЙТРОННО-ФИЗИЧЕСКОГО РАСЧЕТА ЯДЕРНЫХ РЕАКТОРОВ АЭС	11

ЭЛЕКТРОНИКА

ГУСЕВ В.А., СТАРКОВ В.В., ШОФЕРИСТОВ С.Е., МУРЗИН Д.Г. МЕТОДЫ РЕАЛИЗАЦИИ СТРУКТУРЫ ФОТОЭЛЕМЕНТА С ЗАРЯДОВЫМИ НАСОСАМИ	16
---	----

ТЕЛЕКОММУНИКАЦИИ

БАРАННИК В.В., ХАРЧЕНКО Н.А., БЕКИРОВ А.Э. МЕТОДОЛОГИЧЕСКАЯ БАЗА УПРАВЛЕНИЯ БИТОВОЙ СКОРОСТЬЮ ПРИ ФОРМИРОВАНИИ ПРЕДСКАЗАННЫХ КАДРОВ	23
---	----

СИСТЕМЫ И ПРОЦЕССЫ УПРАВЛЕНИЯ

КОЛОСОВА С.В., СИДОРОВ М.В. ПРИМЕНЕНИЕ ИТЕРАЦИОННЫХ МЕТОДОВ К РЕШЕНИЮ ЭЛЛИПТИЧЕСКИХ КРАЕВЫХ ЗАДАЧ С ЭКСПОНЕНЦИАЛЬНОЙ НЕЛИНЕЙНОСТЬЮ	28
ВОРОНИН А.А., ГАРЯЧЕВСКАЯ И.В. АВТОМАТИЗИРОВАННАЯ СИСТЕМА ОБУЧЕНИЯ РАБОТЕ С СИСТЕМОЙ КОНТРОЛЯ ВЕРСИЙ «GIT»	32

КОМПЬЮТЕРНАЯ ИНЖЕНЕРИЯ И ТЕХНИЧЕСКАЯ ДИАГНОСТИКА

ХАХАНОВ В.И., BAGHDADI AMMAR AWNIABBAS, ЧУМАЧЕНКО С.В., ШКИЛЬ А.С., МЕЛИКЯН ВАЗГЕН. КУБИТНЫЕ ТЕХНОЛОГИИ АНАЛИЗА И ДИАГНОСТИРОВАНИЯ ЦИФРОВЫХ УСТРОЙСТВ	36
---	----

КОМПЬЮТЕРНЫЕ НАУКИ

БАРАННИК В.В., РЯБУХА Ю.Н., КРАСНОРУЦКИЙ А.А. МЕТОД РЕКОНСТРУКЦИИ ИЗОБРАЖЕНИЙ В НЕРАВНОМЕРНОМ БАЗИСЕ СПЕКТРАЛЬНЫХ КОЭФФИЦИЕНТОВ	46
ТУРЕНКО С.В. КОДИРОВАНИЕ ВЕКТОРА ДВУХКОМПОНЕНТНЫХ КОРТЕЖЕЙ ДЛЯ ТЕХНОЛОГИЙ КОМПРЕССИИ С ТРАНСФОРМИРОВАНИЕМ КАДРОВ В ИНФОКОММУНИКАЦИОННЫХ СИСТЕМАХ	52
БОНДАРЕНКО М.А., МАКАРЕНКО А.Ю. СИСТЕМЫ ОБ'ЄКТНО-РЕЛЯЦІЙНОЇ ПРОЕКЦІЇ ДЛЯ C++	55

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

КОЗУЛЯ Т.В., БЛОВА М.О. МЕТОД КОМПАРАТОРНОЇ ІДЕНТИФІКАЦІЇ В СИСТЕМІ МЕТОДИЧНОГО ЗАБЕЗПЕЧЕННЯ ОЦІНКИ ЯКОСТІ НАВКОЛИШНЬОГО СЕРЕДОВИЩА	59
РЕФЕРАТИ	66
ПРАВИЛА ОФОРМЛЕНИЯ СТАТЕЙ ДЛЯ АВТОРОВ НАУЧНО-ТЕХНИЧЕСКОГО ЖУРНАЛА	69

КОМПЬЮТЕРНАЯ ИНЖЕНЕРИЯ И ТЕХНИЧЕСКАЯ ДИАГНОСТИКА



УДК681.326:519.713

КУБИТНЫЕ ТЕХНОЛОГИИ АНАЛИЗА И ДИАГНОСТИРОВАНИЯ ЦИФРОВЫХ УСТРОЙСТВ

*ХАХАНОВ В.И., BAGHDADI AMMAR AWNI
ABBAS, ЧУМАЧЕНКО С.В., ШКИЛЬ А.С.,
МЕЛИКЯН ВАЗГЕН*

Предлагаются теория и примеры реализации кубитных моделей, методов и алгоритмов для повышения быстродействия существующих программных и аппаратных средств анализа цифровых вычислительных устройств за счет увеличения размерности структур данных и памяти для одновременного хранения обрабатываемых состояний. Описываются результаты исследований, касающихся моделей и методов диагностирования цифровых систем, моделирования исправного поведения, восстановления работоспособности отказавших примитивов.

1. Введение

Эволюция киберпространства планеты условно делится на следующие периоды: 1) 1980-е годы – формирование парка персональных компьютеров; 2) 1990-е годы – внедрение Интернет-технологий в производственные процессы и быт человека; 3) 2000-е годы – повышение качества жизни за счет внедрения мобильных устройств и облачных сервисов; 4) 2010-е годы – создание цифровой инфраструктуры мониторинга, управления и взаимодействия между собой стационарных и движущихся объектов (воздушный, морской, наземный транспорт и роботы); 5) 2015-е годы – создание глобальной цифровой инфраструктуры киберпространства, где все процессы, явления идентифицируются во времени и в трехмерном пространстве и становятся интеллектуальными. В связи с необходимостью развития параллельных вычислений неупорядоченных данных в последние годы становятся все более значимыми для анализа эволюционирующего кибернетического пространства создания облачных Интернет сервисов, благодаря их позитивной альтернативности существующим затратным повремени классическим моделям последовательной обработки теоретико-множественных структур, за счет существенного расширения памяти [1]. Однако такая плата в настоящее время вполне допустима, поскольку рынок nano-электронных технологий предоставляет сегодня разработчикам цифровых систем до 1 миллиарда вентилях на кристалле размерностью 2x2 см при толщине пластины в 5 микрон. При этом

современные технологии допускают создание пакета или «сэндвича», содержащего до 7 кристаллов. Практически «беспроводное» соединение таких пластин основывается на технологической возможности сверления порядка 10 тысяч сквозных отверстий (vias) на 1 квадратном сантиметре. Кроме того, появление трехмерных FinFETs транзисторов и основанных на них 3D-технологий реализации объемных цифровых систем предоставляют новые возможности для создания более быстродействующих за счет уменьшения задержек параллельных вычислительных устройств [2-7]. Поэтому можно и нужно использовать «жадные» к аппаратуре модели и методы для создания быстродействующих средств параллельного решения практических задач. Имея в виду дискретность и многозначность алфавитов описания информационных процессов, свойство параллелизма (одновременности процессов), заложенное в квантовых вычислениях, является востребованным при создании эффективных и интеллектуальных «движков» для киберпространства, облачных структур и сервисов Интернета; повышения надежности цифровых устройств; тестирования и моделирования дискретных систем на кристаллах. Здесь не рассматриваются физические основы квантовой механики, касающиеся недетерминированного взаимодействия атомных частиц, но используется понятие кубитной структуры как векторной формы совместного или одновременного задания булеана состояний в конечной и дискретной области киберпространства, ориентированного на параллелизм и суперпозицию обработки предлагаемых кубитных моделей и методов.

Квантовые эмуляторы на классических компьютерах достаточно эффективно применяются для решения оптимизационных задач, связанных полным перебором вариантов решений на основе использования теории множеств [1,8]. Особенность в том, что множество элементов в компьютере всегда является упорядоченным, поскольку каждый бит, байт или другой компонент имеет свой адрес. Поэтому все теоретико-множественные операции, так или иначе, сводятся к полному перебору адресов примитивных элементов. Адресный порядок структур данных хорош для задач, где компоненты моделей можно строго ранжировать, что дает возможность выполнять их анализ за один проход или одну итерацию. Там, где нет порядка в структуре, например, множество всех подмножеств, классическая модель памяти и вычислительных процессов наносит вред времени анализа ассоциации равных по рангу примитивов, или, в лучшем случае, обработка ассоциативных групп является неэффективной. Что можно предложить для неупорядоченных данных вместо строгого порядка? Процессор, где элементарной ячейкой служит образ или шаблон универсума из n примитивов, который генерирует $Q = 2^n$ всех возможных состояний такой ячейки в виде булеана или множества всех подмножеств. Прямое решение, ориентированное на создание такой ячейки, использует унитарное позиционное кодирование состо-

яний примитивов, которое с помощью суперпозиции последних образует универсум примитивов, формирующий в пределе булеан или множество всех подмножеств [8,9].

n-Кубит есть векторная форма унитарного кодирования универсума из примитивов для задания булеана

состояний 2^{2^n} спомощью 2^n двоичных переменных. Например, если $n=2$, то 2-кубит задает 16 состояний с помощью 4-х переменных. Если $n=1$, то кубит задает 4 состояния на универсуме из двух примитивов с помощью 2-х двоичных переменных (00,01,10,11) [1]. При этом допускается суперпозиция (одновременное существование) в векторе 2^n состояний. Кубит (n-кубит) дает возможность использовать логические операции вместо теоретико-множественных для существенного ускорения процессов синтеза и анализа дискретных систем. Далее кубит отождествляется с n-кубитом или вектором, если это не мешает пониманиюизлагаемогоматериала. Поскольку квантовыевычисления связаны с анализом кубитных структур данных, то далее частичноэксплуатируется определение«квантовый» дляидентификации технологий, использующих два свойства квантовой механики: параллелизм обработки и суперпозицию состояний.

2. Кубитный метод диагностирования цифровых систем

Предлагается методдиагностированияфункциональныхнарушений и константныхнеисправностей в программных или аппаратных блоках, которые используют «кубитные» или многозначные структуры данных для задания диагностической информации, что дает возможность существенно уменьшить вычислительную сложность процессов моделирования и диагностирования за счет введенияпараллельных логических операций надматричными данными. Представлен кубитный метод исправного моделирования цифровых устройств с восстановлением работоспособности компонентов цифровой системы в режиме online, который имеет существенно более высокое быстродействие за счет адресной реализации процедуры обработки функциональных примитивов, заданных кубитными векторами состояний выходов.

Модель объекта диагностирования представлена в форме графа цифровой системы, которая имеет функциональные элементы, соединенные линиями связей. Среди нихимеютсяассерции – точки наблюдения или мониторинга, необходимые для верификации, тестирования и диагностированиянеисправностей [2]. Диагностическая информация представлена компонентами: 1) Тест проверки или диагностирования неисправностей заданного класса, в данном случае рассматриваются одиночные константные дефекты линий схемы. 2) Таблица неисправностей [6], строки которой задают векторы проверяемых на каждом тестовом наборе дефектов, привязанных к линиям схемы. 3) Матрица достижимостей, которая определяет достижимость каждой ассерционной точки со

стороны множества предшествующих линий [8]. 4) Матрица состояния ассерционного механизма, или матрица экспериментальной проверки, задающая состояние каждой ассерции на тестовых наборах путем сравнения эталонной реакции вданной точке с реальным сигналом в процессе выполнения диагностического эксперимента [2,7].

Базовая модель диагностирования цифрового изделия, дискретного процесса или явления представлена компонентами, которые создают 4 измерения в пространстве признаков:

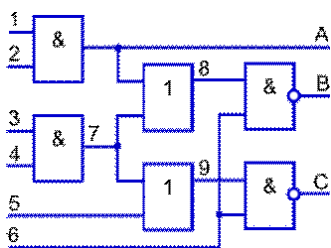
$$\begin{aligned} \spadesuit D_b &= \langle S, A, F, T \rangle \\ \heartsuit D &= \{ \langle S, A \rangle, \langle F, T \rangle \}; \\ \clubsuit V_b &= (\spadesuit | \heartsuit | \clubsuit | \spadesuit); \\ \spadesuit V &= (\spadesuit | \heartsuit | \clubsuit | \spadesuit) + (\heartsuit | \clubsuit | \spadesuit | \heartsuit); \\ \heartsuit V_b &\gg V; \\ \clubsuit S^* &= f(S, A, T); \\ \spadesuit A^* &= g(T, A); \\ \heartsuit F^* &= h(S, A, F, T); \end{aligned}$$

При этом в модели объем диагностической информации Vформируется декартовым произведением(мощностей) четырех компонентов в порядке следования, указанном выше: 1) структура объекта; 2) механизм ассерций или мониторинга; 3) совокупность неисправностей или модулей, подверженных функциональным нарушениям; 4) тестовые наборы или сегменты для диагностирования неисправностей или совокупности упомянутых модулей. Актуально и существенно уменьшить объем диагностической информации можно путем понижения размерности пространства признаков за счет разделения базовой модели на два непересекающихсяподмножества $\langle S, A \rangle, \langle F, T \rangle$. Вэтом случае оценка объема диагностической информации становится не мультипликативной, а аддитивной по отношению к мощности полученных в результате разбиенияподмножеств без какого-либо уменьшения глубины диагностирования. Здесь первый компонент моделидиагностирования представлен матрицей достижимостей, которая позволяетминимизировать маску возможных дефектов на основе анализа структуры схемы путем сравненияистинных и реальных результатов моделирования выходных сигналов на каждом тестовом наборе или сегменте. Число строк такой матрицы равно количеству наблюдаемых выходов или ассерций.

В процессе выполнения метода диагностирования создается двоичная матрица структурной активизации неисправностей, которая служит маской для существенного уменьшения множества подозреваемых дефектов при совместном анализетаблицы неисправностей. При этом символы одиночных константных дефектов $\{0, 1, X, \emptyset\}$, $X = \{0, 1\}$ в ячейках таблицынеисправностей [6] кодируются соответствующими состояниями кубита (10,01,11,00) многозначного алфавита Кантора $A^k = \{0, 1, X, \emptyset\}$, что дает возможность исклю-

чить из вычислительных процессов теоретико-множественные процедуры, заменив их на векторные логические операции.

Для рассмотрения сущности предлагаемого метода используется фрагмент цифровой схемы, представленной на рис. 1. Здесь имеются три ассерционных точки А, В, С для наблюдения за состоянием всех линий схемы в процессе тестирования (выполнения диагностического эксперимента) путем подачи пяти тестовых воздействий, заданных в таблице неисправностей F(T). Координаты данной таблицы задают проверяемые на тест-векторах неисправности 0 и 1, а также имеются состояния координат: $\emptyset(\cdot)$ – отсутствие проверяемых дефектов и X – проверка на линии константы 0 и 1 одновременно. Правая часть таблицы есть матрица состояний ассерционного механизма в виде результатов сравнения эталонной и реальной реакций цифрового устройства на тестовые наборы. Значение 1 означает несравнение, 0 – совпадение упомянутых реакций.



F(T)	1	2	3	4	5	6	7	8	9	A	B	C	V _a	V _b	V _c
111101	0	0	0	0	.	0	0	0	0	0	1	1	1	0	0
010101	1	.	2	3	4	5	6	7	8	9	0	0	0	1	1
101001	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0
000011	0	0	0	1	1	0	1	1	0	1	0	1	0	0	0
111110	0	0	.	1	0	.	1	1	0	0	0	0	1	1	1
F ₁ = T ₁ ∧ S ₁	.	.	0	0	.	.	0	0	0	.	1	1			

Рис. 1. Фрагмент цифровой схемы и таблица неисправностей

В таблице неисправностей не учитывается структура схемы для повышения глубины диагностирования на основе вычисления реальной матрицы состояний ассерционного механизма, которая совместно с матрицей достижимостей создает структурную маску, минимизирующую множество подозреваемых дефектов.

Для фрагмента цифровой схемы, представленной на рис. 1, матрица достижимостей имеет следующий вид:

S = S _{ij}	1	2	3	4	5	6	7	8	9	A	B	C
1	1	1	1	.	.
2	1	1	1	1	.	1	1	1	.	.	1	.
3	.	.	1	1	1	1	1	1

Здесь выходы-ассерции А, В, С являются мониторами технического состояния объекта диагностирования. Каждый из них может иметь два значения: A_{ij} = {0,1}, которые определяют матрицу экспериментальной проверки A = |A_{ij}| путем сравнения эталонных T = |T_{ij}| и

реальных U = |U_{ij}| состояний наблюдаемых или выходных линий: A_{ij} = T_{ij} ⊕ U_{ij}, которые формируют маску возможных дефектов с помощью следующего

$$\text{выражения: } S_i = S(T_i) = \left(\bigvee_{A_{ij}=1} S_{ij} \right) \wedge \left(\overline{\bigvee_{A_{ij}=0} S_{ij}} \right).$$

Каждый тест-вектор (-сегмент) активизирует собственную структуру возможных дефектов, которая функционально зависит от маски, ассерций (состояния наблюдаемых выходов) и тестовых наборов: S = f(S, A, T_i). Если предположить, что в матрице S = |S_{ij}| состояния ассерционных выходов на первом тест-векторе равны A_{1A} = 0; A_{1B} = 1; A_{1C} = 1, где значение 1 идентифицирует проявление дефекта в устройстве, то маска возможных дефектов, согласно

$$\text{функционалу } S_1 = S(T_1) = \left(\bigvee_{A_{1j}=1} S_{1j} \right) \wedge \left(\overline{\bigvee_{A_{1j}=0} S_{1j}} \right),$$

будет иметь следующий вид:

$$S_1 = S(T_1) = (S_2 \vee S_3) \wedge (\overline{S_1}) =$$

$$= (111101110010 \vee 0011111101001) \wedge (\overline{110000000100}) =$$

$$= (111111111011) \wedge (001111111011) = (001111111011).$$

Полученная маска накладывается на первую строку таблицы неисправностей, что определяет множество подозреваемых дефектов:

$$F_i = T_i \wedge S_i \rightarrow F_1 = T_1 \wedge S_1,$$

формирующих ассерционную выходную реакцию A_{1(A,B,C)} = (011) устройства на первый тест-вектор:

В соответствии с предложенной процедурой получения маски одной строки выполняется построение матрицы структурной активизации неисправностей S(T) на основе использования таблицы экспериментальной проверки A = |A_{ij}| задающей состояния ассерционного механизма в процессе выполнения тестирования S(T) = S ⊗ A:

$$S = \begin{matrix} S_{ij} \\ \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & A & B & C \end{matrix} \\ \begin{matrix} 1 & 1 & 1 & . & . & . & . & . & . & 1 & . & . \\ 2 & 1 & 1 & 1 & 1 & . & 1 & 1 & 1 & . & . & 1 & . \\ 3 & . & . & 1 & 1 & 1 & 1 & 1 & . & . & . & . & 1 \end{matrix} \end{matrix} \otimes \begin{matrix} A = |A_{ij}| \\ \begin{matrix} A & B & C \end{matrix} \\ \begin{matrix} T_1 & 1 & 0 & 0 \\ T_2 & 0 & 1 & 1 \\ T_3 & 0 & 0 & 0 \\ T_4 & 0 & 0 & 0 \\ T_5 & 1 & 1 & 1 \end{matrix} \end{matrix} =$$

S(T)	1	2	3	4	5	6	7	8	9	A	B	C
T1	1	1	0	0	0	0	0	0	0	1	0	0
T2	0	0	1	1	1	1	1	1	1	0	1	1
T3	0	0	0	0	0	0	0	0	0	0	0	0
T4	0	0	0	0	0	0	0	0	0	0	0	0
T5	1	1	1	1	1	1	1	1	1	1	1	1

В целях формирования структур данных, удобных для компьютерной обработки, необходимо перевести символы таблицы неисправностей в двухразрядные коды в соответствии с правилами >-кодирования: $\Rightarrow \{0=10, 1=01, X=11, \emptyset=00\}$, применение которых к таблице неисправностей F(T) дает следующий результат:

F(T)	1	2	3	4	5	6	7	8	9	A	B	C
T1	0	0	0	0	.	0	0	0	0	1	1	
T2	1	.	1	.	1	.	1	1	1	0	0	
T3	.	1	.	1	.	1	1	1	0	0		
T4	0	0	1	1	0	1		
T5	0	0	.	.	1	.	.	.	0	0	0	

 \Downarrow

F(T)	1	2	3	4	5	6	7	8	9	A	B	C
T1	10	10	10	10	00	10	10	10	10	10	01	01
T2	01	00	01	00	01	00	01	01	01	01	10	10
T3	00	01	00	01	01	00	01	01	01	01	10	10
T4	00	00	00	00	10	10	01	01	10	01	10	01
T5	10	10	00	00	00	01	00	00	00	10	10	10

После получения структурной матрицы S(T), предназначенной маскировать реальные дефекты в таблице неисправностей и ее кодированной форме, необходимо выполнить #-суперпозицию двух матриц: $F(T) = S(T) \# F(T)$, которая сводится к выполнению #-операции над одноименными координатами $F_{ij} = \bar{F}_j \leftarrow (F_j = 00) \vee (S_{ij} = 0)$, что означает модификацию кодов координат таблицы F(T) при выполнении заданных условий. Иначе, данная операция сводится к инверсии ячеек матрицы кодов неисправностей, маскируемых нулевыми сигналами структурной матрицы активизации, а также всех нулевых кодов таблицы неисправностей. Таблица истинности данной #-операции в символьном и кодированном виде представлена ниже:

$\# = S_{ij} \setminus F_{ij}$	\emptyset	1	0	X
0	X	0	1	\emptyset
1	X	1	0	X

$\# = S_{ij} \setminus F_{ij}$	00	01	10	11
0	11	10	01	00
1	11	01	10	11

Таблица истинности скорректирована относительно инверсии состояния 00 в 11 при единичном значении сигнала активизации неисправности, потому что такой код (00) означает присутствие в схеме на линии пустого множества проверяемых дефектов, что невозможно. Но код 00 еще блокирует все вычисления конъюнкции по столбцу, превращая результат в 00. Инверсия кода дает возможность не маскировать при логическом умножении действительно присутствующих дефектов, любых знаков. При этом предполагается, что невозможно тест-вектором проверить на одной линии схемы дефекты разных знаков.

Выполнение процедуры суперпозиции структурной матрицы с кодированной таблицей неисправностей $F(T) = S(T) \# F(T)$ дает следующий результат:

S(T)	1	2	3	4	5	6	7	8	9	A	B	C
T1	1	1	0	0	0	0	0	0	1	0	0	
T2	0	0	1	1	1	1	1	1	0	1	1	
T3	0	0	0	0	0	0	0	0	0	0	0	
T4	0	0	0	0	0	0	0	0	0	0	0	
T5	1	1	1	1	1	1	1	1	1	1	1	

 \rightarrow

F(T)	1	2	3	4	5	6	7	8	9	A	B	C
T1	10	10	10	10	00	10	10	10	10	10	01	01
T2	01	00	01	00	01	00	01	01	01	01	10	10
T3	00	01	00	01	01	00	01	01	01	01	10	10
T4	00	00	00	00	10	10	01	01	10	01	10	01
T5	10	10	00	00	00	01	00	00	00	10	10	10

F(T)	1	2	3	4	5	6	7	8	9	A	B	C
\wedge	10	10	01	01	11	01	01	01	01	10	10	10
\wedge	10	11	01	11	01	11	01	01	01	10	10	10
\wedge	11	10	11	10	10	11	10	10	10	10	01	01
\wedge	11	11	11	11	01	01	10	10	01	10	01	10
\wedge	10	10	11	11	01	11	11	11	10	10	10	10
$F(T) = \bigwedge_{i=1}^n F_i$	10	10	01	00	00	01	00	00	00	10	00	00
F=	0	0	1	.	.	1	.	.	.	0	.	.

На заключительной стадии диагностирования выполняется единственная векторная операция логического умножения всех строк кодированной модифицированной таблицы истинности F(T):

$$F(T) = \left(\bigvee_{A_i=1} F_i \right) \wedge \left(\bigvee_{A_i=0} \bar{F}_i \right) = \left(\bigwedge_{A_i=1} F_i \right) \wedge \left(\bigvee_{A_i=0} \bar{F}_i \right) = \left(\bigwedge_{A_i=1} F_i \right) \wedge \left(\bigwedge_{A_i=0} \bar{F}_i \right) = \left(\bigwedge_{i=1}^n F_i \right).$$

Это дает возможность точно определить все дефекты, присутствующие в объекте диагностирования, которые представлены в двух нижних строках приведенной выше кодированной таблицы неисправностей F(T):

$$F = \{1^0, 2^0, 3^1, 6^1, A^0\}.$$

Теоретическое доказательство матричного диагностирования одиночных и кратных дефектов представлено в виде двух теорем.

Теорема 1. *Одиночные константные дефекты цифровой схемы, заданные кубитами (двухразрядными векторами) на тестовых наборах многозначной таблицы неисправностей, определяются с помощью векторной and-операции, маскируемой строкам матрицей (вектором) экспериментальной проверки $A < |A_{ij}|$ всех ассерционных точек:*

$$F(T) = \left(\bigvee_{A_i=1} F_i \right) \wedge \left(\bigvee_{A_i=0} \bar{F}_i \right) = \left(\bigwedge_{A_i=1} F_i \right) \wedge \left(\bigvee_{A_i=0} \bar{F}_i \right) = \left(\bigwedge_{A_i=1} F_i \right) \wedge \left(\bigwedge_{A_i=0} \bar{F}_i \right) = \left(\bigwedge_{i=1}^n F_i \right).$$

Выражение является верным, потому что: 1) Второй сомножитель – чистая математика – отрицание дизъюнкции есть конъюнкция отрицаний, что означает умножение кодов таблицы с их предварительным отрицанием. 2) Первый сомножитель ориентирован на поиск непротиворечивых дефектов, поэтому он заменяется на $(\bigwedge_{A_i=1} F_i)$. Действительно, на одной линии

или переменной не могут присутствовать одновременно две противоположных по знаку проверяемых неисправности. Поэтому в базовой формуле дизъюнкция дефектов $(\bigvee_{A_i=1} F_i)$ в большей степени ориентирована на поиск кратных неисправностей, но не связанных с одной линией. Кратность противоречивых дефектов на одной линии, равно как и инверсия

пустого множества неисправностей, теоретически создает условия беспрепятственного умножения других ячеек столбца в целях формирования на каждой линии результата в виде дефекта одного знака или пустого множества неисправностей.

Теорема 2. *Кратные константные дефекты цифровой схемы, заданные кубитами на тестовых наборах многозначной таблицы неисправностей, определяются с помощью векторных or- and- операций, маскируемых по строкам вектором экспериментальной проверки A(T) всех ассерционных точек:*

$$F(T) = (\bigvee_{A_i=1} F_i) \wedge (\overline{\bigvee_{A_i=0} F_i}) = (\bigvee_{A_i=1} F_i) \wedge (\bigwedge_{A_i=0} \bar{F}_i).$$

Выражение является верным, потому что: 1) Второй сомножитель есть отрицание дизъюнкции или конъюнкция отрицаний, что означает умножение кодов таблицы с их предварительным отрицанием. 2) Первый сомножитель ориентирован на поиск кратных дефектов в предположении, что на одной линии или переменной могут присутствовать одновременно две противоположных по знаку проверяемых неисправности. Данная формула в большей степени ориентирована на поиск кратных дефектов в блоках цифровых систем, не связанных с одной линией. Кратность неисправностей в цифровой системе теоретически создает условия для логического сложения других ячеек столбца в целях формирования результата в виде множества дефектов, формирующих заданный вектор экспериментальной проверки, из которых необходимо вычесть проверяемые на тесте неисправности, которые не влияют на формирование некорректных реакций по выходам.

Интерес представляет поиск кратных дефектов на основе мультипроцессора Хассе [4,5], который ориентирован на решение задачи покрытия путем полного перебора событий, обеспечивающих точное покрытие вектора экспериментальной проверки столбцами таблицы неисправностей: $F(T) = (\bigvee_i F_i) \oplus A = 0$.

Здесь решением является такое сочетание столбцов, участвующих в векторной операции логического сложения, которое в совокупности дает результат, равный вектору экспериментальной проверки. Поскольку операция времязатратная, то для нее следует использовать мультипроцессор Хассе, ориентированный на взятие булеана в почти параллельном режиме.

Подводя итог, следует представить модель процесса диагностирования цифровых устройств, которая содержит функциональные преобразователи, связанные с выполнением следующих шагов (рис. 2):

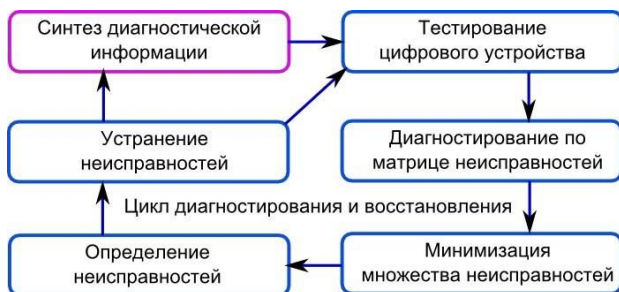


Рис. 2. Цикл диагностирования и ремонта логических блоков

1. Препроцессирование сводится к генерированию исходной диагностической информации в виде теста диагностирования, таблицы неисправностей и матрицы достижимостей цифровой системы.

2. Тестирование реального устройства на основе использования промышленного симулятора в целях сравнения фактических реакций устройства с эталонными значениями по наблюдаемым линиям-ассерциям, что дает возможность сформировать матрицу выходных реакций или вектор экспериментальной проверки в двоичном алфавите.

3. Вычисление матрицы активности графовой структуры на каждом входном тестовом наборе, равной по размерности таблице неисправностей, с помощью матрицы экспериментальной проверки и матрицы достижимостей, что дает возможность существенно сократить область подозреваемых дефектов.

4. Модификация содержимого таблицы неисправностей путем ее маскирования матрицей активности графовой структуры, в целях определения только тех неисправностей, которые действительно формируют матрицу экспериментальной проверки в процессе диагностирования.

5. Выполнение процедуры логического умножения над строками таблицы неисправностей для получения вектора подозреваемых дефектов.

6. Восстановление работоспособности цифрового устройства путем переадресации неисправных логических компонентов на их аналоги из ремонтного запаса и повторение процесса тестового диагностирования.

Таким образом, новизна предложенного метода диагностирования дефектов заключается в использовании для получения диагноза единственной параллельной операции логического умножения, что в сочетании со структурным маскированием неисправностей дает преимущества перед аналогами в части увеличения быстродействия и повышения глубины диагностирования.

3. Кубитное моделирование цифровых систем

Рассматриваются структуры данных, эффективные с точки зрения программной или аппаратной реализации исправного интерпретативного моделирования дискретных систем, описанных в форме кубитных векторов состояний выходов примитивов. Для описания цифровой схемы, представленной на рис. 3, традиционно используется структура взаимосвязанных элементов кубические покрытия (таблицы истинности) логических элементов.

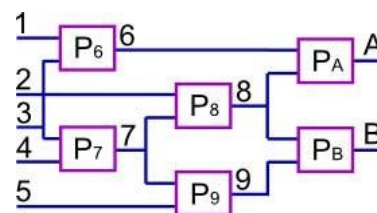


Рис. 3. Фрагмент цифровой схемы

Цель предлагаемого метода кубитного моделирования – заменить таблицы истинности компонентов циф-

рового устройства векторами состояний выходов. Пусть функциональный примитив с номером P_6 имеет следующую таблицу истинности:

$$P_6 = \begin{array}{|c|c|c|} \hline X_1 & X_2 & Y \\ \hline 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ \hline \end{array}.$$

Данное покрытие логического элемента можно трансформировать путем унитарного кодирования входных векторов на основе использования двухтактного алфавита [4-7]. Символы и их коды, предназначенные для описания автоматных переменных, представляют собой булеан на универсуме из четырех примитивов, что соответствует формату вектора, содержащего два кубита:

$$\begin{aligned} V^*(Y) = \{ & Q=(1000), E=(0100), H=(0010), J=(0001), \\ O = \{ & Q,H \}=(1010), I = \{ E,J \}=(0101), A = \{ Q,E \}=(1100), \\ B = \{ & H,J \}=(0011), S = \{ Q,J \}=(1001), P = \{ E,H \}=(0110), \\ C = \{ & E,H,J \}=(1110), F = \{ Q,H,J \}=(1011), \\ L = \{ & Q,E,J \}=(1101), V = \{ Q,E,H \}=(1110), \\ Y = \{ & Q,E,H,J \}=(1111), U=(0000) \}. \end{aligned}$$

С помощью двухтактного алфавита любое покрытие функционального одновыходового примитива всегда можно представить двумя кубами или даже одним, учитывая, что кубы взаимно инверсны:

$$P_6 = \begin{array}{|c|c|} \hline 00 & 1 \\ 01 & 1 \\ 10 & 1 \\ 11 & 0 \\ \hline \end{array} = \begin{array}{|c|c|} \hline Q & 1 \\ E & 1 \\ H & 1 \\ J & 0 \\ \hline \end{array} = \begin{array}{|c|c|} \hline V & 1 \\ J & 0 \\ \hline \end{array} = \begin{array}{|c|c|} \hline 1110 & 1 \\ 0001 & 0 \\ \hline \end{array} \rightarrow \boxed{1110}$$

Два куба показывают не только все решения, но и инверсию сигналов на выходе, что интересно с позиции активизации всех логических путей в схемной структуре при синтезе тестов. Например, для изменения состояния выхода необходимо создать на входах пару следующих друг за другом условий, где в первом такте должны быть первые три вектора (адреса), а во втором – четвертый вектор, формируемый двумя входными переменными.

Для моделирования исправного поведения достаточно иметь только один куб (нулевой или единичный), поскольку второй всегда является дополнением к первому. Следовательно, ориентируясь, например, на единичный куб, формирующий на выходе 1, можно убрать бит состояния выхода примитива, что уменьшит размерность куба или модели примитива до количества адресуемых состояний элемента, где адрес есть вектор, составленный из двоичных значений входных переменных, по которому определяется состояние выхода примитива.

Кубитное покрытие, или Q-покрытие есть векторная интерпретативная форма задания функциональности, где значение координаты определяет состояние выхода функции, соответствующее двоичному входному слову, формирующему адрес ячейки. Q-покрытие одновыходового примитива всегда представлено двумя

взаимноинверсными кубами (векторами), размерность которых равна степени двойки от числа входных переменных, где единичное значение координаты определяет участие адреса рассматриваемого бита в формировании соответствующего (0,1) состояния выхода примитива. Кубитные модели примитивов требуют создания новой теории моделирования, прямой и обратной импликации, синтеза тестов, моделирования неисправностей, поиска дефектов. Здесь и далее представлены основные процедуры исправного моделирования на основе манипулирования адресами, неявно представленными в координатах кубов Q-покрытия.

Модель для анализа цифровой системы на основе использования кубитных структур данных может быть описана четырьмя компонентами:

$$\begin{aligned} F = \langle & L, M, X, Q \rangle, \\ L = & (L_1, L_2, \dots, L_j, \dots, L_n); \\ M = & (M_1, M_2, \dots, M_j, \dots, M_n); \\ X = & (X_{n_x+1}, X_{n_x+2}, \dots, X_{n_x+i}, \dots, X_n); \\ Q = & (Q_{n_x+1}, Q_{n_x+2}, \dots, Q_{n_x+i}, \dots, Q_n). \end{aligned}$$

Здесь представлены соответственно: L – вектор идентификаторов эквипотенциальных линий схемы цифровой системы, который ввиду своей тривиальности может быть исключен из модели, но при этом необходимо иметь число входных переменных устройства и общее количество линий; M – вектор моделирования состояний всех линий схемы; X – упорядоченная совокупность векторов входных переменных каждого примитива схемы, привязанных к номерам выходов, Q – совокупность Q-покрытий примитивов, строго привязанных к номерам выходов и входным переменным примитивов; n – число линий в схеме, n_x – количество входных переменных.

В качестве примера кубитного задания модели цифрового устройства $F = \langle L, M, X, Q \rangle$, ниже приведен вариант структурной таблицы описания схемы для анализа исправного поведения:

L	1	2	3	4	5	6	7	8	9	A	B
M	1	1	1	1	0	1	0	1	1	1	0
X	13	34	27	75	68	89
Q	1	0	1	1	1	1
	1	1	0	0	0	0
	1	1	0	0	1	1
	0	1	0	1	0	1

Метод кубитного моделирования исправного поведения сводится к определению значения выхода элемента по адресу, формируемому конкатенацией двоичных состояний входных переменных каждого примитива цифровой системы

$$M(Y_i) = Q_i[M(X_{i1} * X_{i2} * \dots * X_{ij} * \dots * X_{ik_i})].$$

Здесь k_i – число входных линий в примитиве с номером i . Поскольку номера невходных линий век-

тора. Однозначно идентифицируют по выходам обрабатываемые примитивы, то формула моделирования может быть приведена к циклоопределению состояний всех невходных переменных:

$$M_i = Q_i[M(X_{i1} * X_{i2} \dots * X_{ij} \dots * X_{ik_i})] = Q_i[M(A_i)], \quad i = n_x + 1, n$$

Здесь процесс моделирования связан с конкатенированным формированием адреса бита в кубите функциональности, который определяет состояние примитива или невходной линии цифровой структуры, начиная с номера $i = n_x + 1$. Если переменные создают не двоичный адрес, то в данном случае существует возможность формирования состояния выхода логического элемента в троичном алфавите символом X. Состояния выходов формируются идеально примитивной процедурой обработки кубита примитива

$$M_i = Q_i[M(X_i)]$$

на основе простых итераций или итераций Зейделя [6,8]. Во втором случае необходима препроцессорная процедура ранжирования линий и примитивов схемы, которая позволяет существенно уменьшить количество проходов по элементам схемы для достижения сходимости, когда фиксируется равенство состояний всех линий в двух соседних итерациях. Кроме того, ранжирование примитивов по уровням формирования выходов дает возможность существенно повысить быстродействие моделирования за счет параллельной обработки функциональных элементов одного уровня. Например, для схемы, представленной на рис. 6, одновременно можно обрабатывать элементы с номерами 6,7, затем – 8,9 и далее – А,В. В первом случае, когда используются простые итерации, ранжирования не требуется, но платой за простоту алгоритма моделирования является существенно большее число итеративных проходов по примитивам схемы для достижения упомянутого критерия сходимости. Вычислительная сложность предложенного Q-метода моделирования на основе кубитных функциональностей определяется процедурами формирования адреса – входного вектора, содержащего k_i переменных, для каждого i -го примитива $[(r+w) \times k_i]$, считыванием бита из квант-вектора по конкатенированному адресу и записью $(r+w)$ данного бита в вектор моделирования:

$$\eta = \sum_{i=n_x+1}^n \{[(r+w) \times k_i] + (r+w)\} = \sum_{i=n_x+1}^n [(r+w) \times (k_i + 1)] = (r+w) \times \sum_{i=n_x+1}^n (k_i + 1).$$

Время моделирования одного тест-вектора Q-методом, при условии, что цифровая схема, составленная из 900 четырехходовых примитивов, имеет параметры: $r = w = 5ns$, $k_i = 4$, $n_x = 100$, $n = 1000$, равно 45 микросекунд:

$$\eta = (r+w) \times \sum_{i=n_x+1}^n (k_i + 1) = (5+5) \times 900 \times (4+1) = 10 \times 900 \times 5 = 45000ns = 45\mu s.$$

Это означает, что быстродействие интерпретативного Q-метода моделирования дает возможность для данной схемы обработать за одну секунду 22 222 входных наборов. При этом цифровое устройство имеет существенное преимущество – сервисную функцию online восстановления работоспособности в случае отказа примитива путем его переадресации на запасной элемент.

Для синтеза квазиоптимальных структур данных комбинационного устройства необходимо использовать следующие правила:

- 1) Ранжированная схема цифрового устройства по структурной глубине для моделирования по способу Зейделя должна иметь по возможности однотипные примитивы в каждом уровне (слое) срабатывания.
- 2) В каждом уровне желательно иметь одинаковое число примитивов. Поэтому синтез цифрового устройства следует ориентировать на создание прямоугольной (матричной) структуры однотипных логических элементов.
- 3) Реализация комбинационных примитивов предполагает использование адресуемых элементов памяти, имеющих место в программируемых логических устройствах (FPGA, CPLD), широко используемых для создания прототипов.
- 4) Формирование для каждого уровня комбинационного устройства ремонтных примитивов для восстановления работоспособности в режиме online из расчета – один запасной элемент на каждый тип компонента, используемый в уровне.
- 5) Стоимость аппаратных затрат для реализации комбинационного устройства, ориентированного на высокое быстродействие, должна определяться суммой всех примитивов, привязанных к уровням комбинационного устройства, дополненной линейкой запасных элементов по одному для каждого слоя (при условии существования в каждом слое одинаковых

$$\text{примитивов): } Q = \sum_{i=1, n}^{j=\Gamma, m} P_{ij} + n.$$

- 6) Реализация комбинационного устройства, ориентированного на минимизацию аппаратных затрат, определяется суммой всех типов примитивов, инвариантных к уровням комбинационного устройства, дополненной линейкой запасных элементов по одному

$$\text{для каждого типа: } Q = \sum_{i=1}^m P_i + m.$$

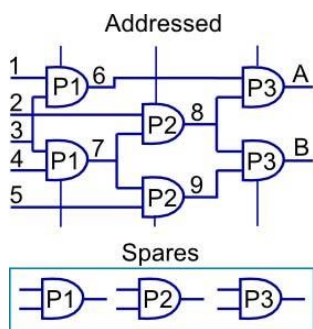
- 7) Обработка матрицы комбинационных элементов с помощью процессорной линейки примитивов, число которых равно мощности максимального уровня или слоя в прямоугольной структуре, что обеспечивает условия для параллельной обработки всех примитивов в каждом уровне элементов в целях повышения быстродействия комбинационного прототипа, реализуемого в PLD.

Таким образом, новизна предложенного Q-метода интерпретативного исправного моделирования цифровых схем заключается в существенном повышении быстродействия и уменьшении объемов структур данных за счет замены таблиц истинности на Q-покрытия, что практически делает разработку конкурентоспособной с технологиями компилятивного моделирования.

4. Восстановление работоспособности комбинационных устройств

Немногочисленные работы, посвященные восстановлению работоспособности логических схем [9-11], описывают две идеи. Первая заключается в реконфигурации структуры логических элементов в режиме offline, которая обеспечивает возможность замены каждого неисправных примитивов. Вторая создает условия для замены неисправных элементов путем использования запасных логических компонентов и мультиплексоров для переадресации отказавших примитивов.

Структуры кубитных данных модифицируются в сторону дополнения строкой типов примитивов $F = \langle L, M, X, P, Q \rangle$, $P = (P_1, P_2, \dots, P_1, \dots, P_m)$, задействованных при синтезе цифровой системы, если необходимо в процессе функционирования выполнять ремонт или восстановление работоспособности за счет введения запасных примитивов, которые, так же как и основные, реализуются на основе элементов памяти. На рис. 4 изображен пример схемной структуры из адресуемых и трех запасных элементов.



L	1	2	3	4	5	6	7	8	9	A	B
M	1	1	1	1	1	0	0	1	1	1	0
X	13	34	27	75	68	89
P	1	2	3	4	5	6
Q	1	1	0	0	1	1	1	0	1	.	.
	1	1	1	1	0	0	1	1	0	.	.
	1	1	1	1	1	1	1	1	1	.	.
	0	0	0	0	0	0	0	0	0	.	.

Рис. 4. Пример схемной структуры из адресуемых и запасных элементов

Таблица (см. рис. 4) оперирует номерами структурных примитивов, что дает возможность заменить любой отказавший элемент исправным из ремонтного запаса путем изменения адресного номера в строке

примитивов P. Ремонтные элементы в данной таблице начинаются со столбца номер 7.

В следующей таблице представлены строка типов логических элементов, а также адреса типов этих примитивов, отмеченные номерами:

L	1	2	3	4	5	6	7	8	9	A	B
M	1	1	1	1	1	0	0	1	1	1	0
X	13	34	27	75	68	89
P	1	1	2	2	3	3
Q	1	0	1	1	0	1
	1	1	0	1	1	0
	1	1	1	1	1	1
	0	0	0	0	0	0

Эта структура данных ориентирована на программную реализацию моделирования, а ремонтные примитивы начинаются с номера 4. Если существует возможность перепрограммирования логики в элементе памяти с одинаковым числом входных переменных, то данную процедуру следует выполнять после фиксации неисправного элемента, когда становится известно – какой элемент в структуре и какой тип примитива отказал. Процедура восстановления работоспособности ориентирована на PLD-реализацию цифровых систем. Если кубитные модели схем не имеют запасных примитивов, то соответствующий формат таблиц будет иметь следующий вид:

L	1	2	3	4	5	6	7	8	9	A	B
M	1	1	1	1	1	0	0	1	1	1	0
X	13	34	27	75	68	89
P	1	1	2	2	3	3
Q	1	0	1
	1	1	0
	1	1	1
	0	0	0

Таким образом, квантовые структуры данных ориентированы на компактность описания функционалов цифрового изделия кубитными векторами, повышение быстродействия процедур моделирования за счет адресации состояний выходов примитивов, а также на восстановление работоспособности отдельных логических элементов, благодаря их реализации в элементах памяти PLD или в форме программных модулей. Очень важно, что в последнем случае не нужно хранить ремонтные примитивы, поскольку предложенные здесь интерпретативные структуры табличных данных изначально ориентированы на технологические удобства устранения дефектов в процессе функционирования прототипа цифрового изделия.

Обработка схемы в кристалле сводится к определению адреса, составленного двоичными битами вектора моделирования, по которому находится значение логической функции. Каждый примитив имеет цикл обработки, содержащий три процедуры:

- 1) Адресное считывание номеров входных переменных из соответствующего столбца матрицы X для формирования адреса состояния входной переменной вектора моделирования: $A = X_{ij}$, $i = \overline{1, n}$; $j = \overline{1, s_p - 1}$.

2) Формирование адреса (двоичного кода) для вычисления логической функции путем конкатенации соответствующих состояний входных переменных в векторе моделирования $A = M(X_{ij}) * M(X_{ir})$.

3) Запись результата выполнения логической функции как состояния выхода в соответствующий разряд вектора моделирования

$$M(X_{is_p}) = P[M(X_{ij}) * M(X_{ir})].$$

Процесс обработки всех примитивов схемы в данном случае является строго последовательным, что представляет собой существенное замедление процедуры формирования состояний выходных переменных. Однако уменьшение быстродействия можно считать платой за сервис встроенной и автономной восстановления работоспособности цифровой структуры, который является одним из этапов функционирования инфраструктуры обслуживания SoC, представленной на рис. 5.

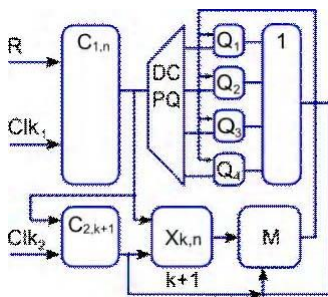


Рис. 5. Операционная структура комбинационной схемы

Комбинационная схема становится операционным устройством, где присутствуют операционный и управляющий автоматы. Заменяемыми компонентами в операционном автомате являются типы примитивов – функциональные элементы или структурные примитивы.

Операционное устройство для реализации элементно-адресуемых комбинационных схем содержит: счетчик обработки текущего примитива C_1 ; память для хранения типов примитивов, соответствующих структурным элементам P ; счетчик считывания номеров входных и выходных переменных текущего примитива C_2 ; дешифратор типов примитивов DC ; память для хранения вектора моделирования M ; матричная память для хранения номеров входов-выходов структурных примитивов X ; линейка памяти, реализующих функциональные примитивы $P(Q)$; регистр формирования входного адресного слова для обрабатываемого примитива RG ; логический элемент Or для коммутации результатов обработки функциональных примитивов.

Граф-схема алгоритма управления процессом моделирования структуры комбинационной схемы представлена на рис. 6 и содержит следующие шаги:

1. Инициализация (формирование) всех компонентов (номера и типы элементов, линии связей для входов и выходов логических элементов) схемной структуры:

$$P = (P_1, P_2, \dots, P_1, \dots, P_n); Q = (Q_1, Q_2, \dots, Q_j, \dots, Q_m);$$

$$X = [X_{pq}]; p = \overline{1, n}; q = \overline{1, s_p}.$$

2. Инициализация параметра обрабатываемого примитива и номера входного набора $i=0, t=0$ для его моделирования в двоичном алфавите $M_T = \{0, 1\}$.

3. Инкрементирование индекса примитива, номера теста и инициализация входного тестового (рабочего) набора: $i = i + 1, t = t + 1, M(X) = T_t(X), |T_t(X)| = \eta$.

4. Конкатенация (#) разрядов слова для формирования входного воздействия $\#_{j=1}^k M(X_{ij})$ логического элемента P_i (типа Q_i) и выполнение процедуры определения состояния его выхода с последующей записью в соответствующую координату вектора моделирования: $M(X_{k+1})$:

$$M(X_{k+1}) = \{P_i, Q_i\} \#_{j=1}^k M(X_{ij}).$$

5. Повторение пунктов 3 и 4 в целях получения состояний выходов всех логических элементов до выполнения условия: $i = n$.

6. Повторение пунктов 2–4 в целях моделирования всех входных тестовых (рабочих) наборов, до выполнения равенства: $t = \eta$, где η – длина теста.

7. Окончание процесса моделирования цифрового устройства.

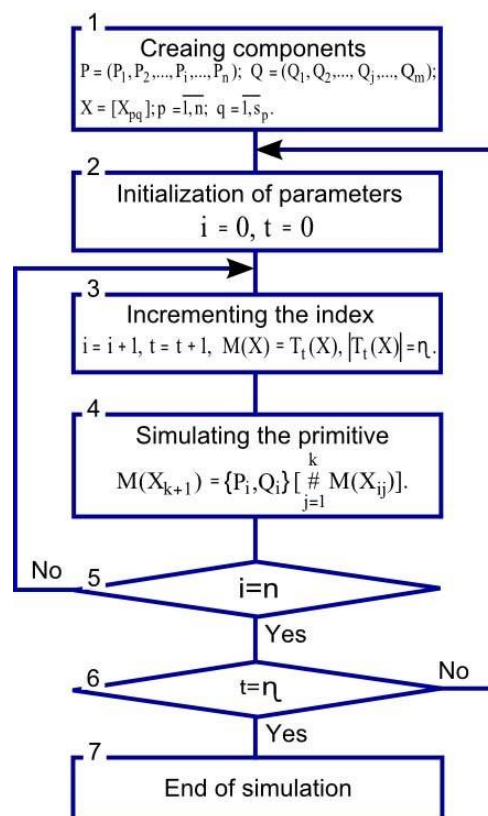


Рис. 6. Граф-схема алгоритма управления процессом моделирования

Таким образом, научная новизна предложенной модели цифровой системы заключается во введении в структуру устройства избыточных ремонтных компонентов и управляющего автомата, ориентированного на последовательную обработку комбинационных примитивов, что дает возможность осуществлять процедуру переадресации примитивов в случае отказа одного из них. Нетрудно создать аналогичные автоматы для параллельной обработки слоев из примитивов ранжированной схемы, что максимально приблизит быстродействие устройства к его реализации в кристаллах PLD.

5. Заключение

Научная новизна и практическая значимость данной публикации, основанная на применении кубитных структур, ориентированных на параллельное вычисление теоретико-множественных сути данных, формируется в следующие пункты:

1. Усовершенствован метод диагностирования дефектов цифровых систем за счет использования единственной параллельной операции логического умножения, что в сочетании со структурным маскированием неисправностей дает преимущества перед аналогами в части компактности представления данных, увеличения быстродействия и повышения глубины диагностирования.
2. Предложен новый Q-метод интерпретативного исправного моделирования цифровых схем, который характеризуется использованием компактных Q-покрытий вместо таблиц истинности, что дает возможность существенно повысить быстродействие анализа за счет адресного формирования выходов функциональных примитивов и уменьшить объемы структур данных, что практически делает метод конкурентоспособным с технологиями компилятивного моделирования.
3. Усовершенствована модель цифровой системы путем дополнения в структуру устройства избыточных ремонтных компонентов и управляющего автомата, ориентированного на последовательную обработку комбинационных примитивов, что дает возможность осуществлять процедуру переадресации отказавших примитивов в режиме штатного функционирования.
4. Показаны примеры использования кубитных структур данных и квантовых преобразований для моделирования цифровых схем и решения задач диагностирования путем использования векторных параллельных логических операций и ремонтных исправных модулей на основе адресуемых логических примитивов.
5. Основная инновационная идея квантовых или кубитных вычислений заключается в переходе от вычислительных процедур над байт-операндом, определяющим в дискретном пространстве одно решение (точку), к квантовым параллельным процессам над кубит-операндом, одновременно формирующим булеан решений.

Литература: 1. *Michael A. Nielsen & Isaac L. Chuang*. Quantum Computation and Quantum Information. Cambridge University Press. 2010. 676р. 2. *Проектирование и тестирование цифровых систем на кристаллах*/В.И.Хаханов, Е.И. Литвинова, О.А. Гузь. Харьков: ХНУРЭ. 2009. 484 с. 3. *Hahanov V., Wajeb Gharibi, Litvinova E., Chumachenko S.* Information analysis infrastructure for diagnosis // Information an int. interdisciplinary journal. Japan. 2011. Vol.14. № 7. P. 2419-2433. 4. *Хаханов В.И., Мурад Али А., Литвинова Е.И., Гузь О.А., Хаханова И.В.* Квантовые модели вычислительных процессов // Радиоэлектроника и информатика. 2011. № 3. С.35-40. 5. *Бондаренко М.Ф., Хаханов В.И., Литвинова Е.И.* Структура логического ассоциативного мультипроцессора. Автоматика и телемеханика. 2012. №10. С. 71-92. 6. *Хаханов В.И.* Техническая диагностика цифровых и микропроцессорных структур. К.: ИСИО. 1995. 242 с. 7. *Vladimir Hahanov, Alexander Barkalov and Marian Adamsky.* Infrastructure intellectual property for SoC simulation and diagnosis service. Springer, 2011. P. 289-330. 8. *Горбатов В.А.* Основы дискретной математики. М.: Высш. шк. 1986. 311 с. 9. *Хаханов В.И., Литвинова Е.И., Хаханова И.В., Murad Ali Abbas.* Инфраструктура встроенного восстановления логических PLD-схем // Радиоэлектроника и информатика. 2012. №2. С. 54-57. 10. *Hahanov V., Litvinova E., Gharibi W., Murad Ali Abbas.* Qubit models for SoC Synthesis Parallel and cloud computing. USA. 2012. Vol.1. Iss 1. P. 16-20. 11. *Hahanov V.I., Litvinova E.I., Chumachenko S.V., Baghdadi Ammar Awni Abbas, Eshetie Abebech, Mandefro.* Qubit Model for solving the coverage problem // Proc. of IEEE East-West Design and Test Symposium. IEEE. USA. Kharkov. 14-17 September 2012. P.142 - 144. 12. *Чжен Г., Мэннинг Е., Мему Г.* Диагностика отказов цифровых вычислительных систем. М.: Мир, 1972. 230 с. 13. *Koal T., Scheit D., Vierhaus H.T.* A comprehensive scheme for logic self repair // Conf. Proc. on Signal Processing Algorithms, Architectures, Arrangements, and Applications. 2009. P. 13 – 18.

Поступила в редколлегию 25.08.2013

Рецензент: д-р техн. наук, проф. Кривуля Г.Ф.

Хаханов Владимир Иванович, декан факультета КИУ, д-р техн. наук, профессор кафедры АПВТ ХНУРЭ. Научные интересы: проектирование и тестирование вычислительных систем, сетей и программных продуктов. Увлечения: баскетбол, футбол, теннис, горные лыжи. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. +3805770-21-326, E-mail: hahanov@kture.kharkov.ua.

Baghdadi Ammar Awni Abbas, аспирант кафедры АПВТ ХНУРЭ. Научные интересы: проектирование и тестирование вычислительных систем. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. +380 57 70-21-326, E-mail: hahanov@kture.kharkov.ua.

Чумаченко Светлана Викторовна, д-р техн. наук, профессор, заведующая кафедрой АПВТ ХНУРЭ. Научные интересы: дискретная математика, моделирование вычислительных систем. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. +38057 70-21-326, E-mail: ri@kture.kharkov.ua.

Шкиль Александр Сергеевич, канд. техн. наук, доцент, доцент кафедры АПВТ ХНУРЭ. Научные интересы: проектирование и тестирование вычислительных систем. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. +380 57 70-21-326, E-mail: shkil@kture.kharkov.ua.

Меликян Вазген, д-р техн. наук, профессор Государственного инженерного университета Армении. Научные интересы: design&test. Адрес: Армения, Ереван, ул. Теряна, 105, тел: (374 10) 524 629.

АННОТАЦИИ

УДК 681.326:519.713

КУБИТНЫЕ ТЕХНОЛОГИИ АНАЛИЗА И ДИАГНОСТИРОВАНИЯ ЦИФРОВЫХ УСТРОЙСТВ

*ХАХАНОВ В.И., BAGHDADI AMMAR AWNI ABBAS, ЧУМАЧЕНКО С.В., ШКИЛЬ А.С.,
МЕЛИКЯН ВАЗГЕН*

Предлагаются теория и примеры реализации кубитных моделей, методов и алгоритмов для повышения быстродействия существующих программных и аппаратных средств анализа цифровых вычислительных устройств за счет увеличения размерности структур данных и памяти для одновременного хранения обрабатываемых состояний. Описываются результаты исследований, касающихся моделей и методов диагностирования цифровых систем, моделирования исправного поведения, восстановления работоспособности отказавших примитивов.

ABSTRACTS

UDC 681.326:519.713

Qubit technologies for analysis and diagnosis of computer systems / V. I. Hahanov, Baghdadi Ammar Awni Abbas, S.V. Chumachenko, A.S. Shkil // Radioelektronika i informatika. 2013. № 3. P.36-45.

A theory and examples of quantum models, methods and algorithms to improve the performance of existing software and hardware for analyzing and synthesizing digital computing devices by increasing the dimension of the data structures and memory для одновременного хранения обрабатываемых состояний are proposed. The investigation results of design theory for computer products, as well as models and methods of quantum diagnosis of digital systems, qubit modeling fault-free behavior, analysis of digital devices with repair of failed entities are presented.

Fig. 6. Ref.: 13 items.

РЕФЕРАТИ

УДК 681.326:519.713

Кубітні технології аналізу і діагностування цифрових пристроїв / В.І. Хаханов, Baghdadi Ammar Awni Abbas , С.В. Чумаченко, А.С. Шкіль // *Радіоелектроніка і інформатика*. 2013. № 3 . С.36-45.

Запропоновано теорію і приклади реалізації кубітних моделей, методів і алгоритмів для підвищення швидкодії існуючих програмних і апаратних засобів аналізу і синтезу цифрових обчислювальних пристроїв за рахунок збільшення розмірності структур даних і пам'яті для одночасного зберігання оброблюваних станів. Представлені результати досліджень, що стосуються теорії проектування комп'ютерних виробів, а також моделей і методів кубітного діагностування цифрових систем, кубітного моделювання справної поведінки, аналізу дискретних пристроїв з відновленням працездатності примітивів, що відмовили.

Лл. 6 . Бібліогр. : 13 назв.