

УДК 681.326:519.713

**В.И. Хаханов**, д-р техн. наук,  
**И.В. Емельянов**, аспирант, **М.М. Любарский**, аспирант,  
**С.В. Чумаченко**, д-р техн. наук, **Е.И. Литвинова**, д-р техн. наук  
Харьковский национальный университет радиоэлектроники  
(Украина, 61166, Харьков, пр-т Ленина, 14,  
тел. (057) 7021326, hahanov@icloud.com)

### **Квантовый метод синтеза тестов на основе кубитных структур данных**

Предложено одно из возможных решений проблемы создания и апробирования на классических компьютерах теории и методов квантовых вычислений на памяти (без использования логики) для последующего применения во всех сферах человеческой деятельности. Сформулированы инженерно-ориентированные определения видов компьютинга, включая квантовый, использующий понятия суперпозиции и перепутывания, а также компьютинг на памяти без использования логики. Показана необходимость совместного и параллельного решения проблемы создания рыночно доступного квантового компьютера и разработки кванто-ориентированных приложений и облачных сервисов. Приведены примеры квантового проектирования и тестирования фрагментов цифровых схем. Предложен метод синтеза и минимизации тестов для функциональностей с использованием матрицы кубитных производных и секвенсор для определения квазиоптимального покрытия.

*К л ю ч е в ы е с л о в а: синтез тестов, кубитное покрытие, компьютинг на памяти, цифровая схема, булева кубитная производная, моделирование неисправностей.*

Запропоновано одне з можливих вирішень проблеми створення та апробування на класичних комп'ютерах теорії і методів квантових обчислень на пам'яті (без використання логіки) для подальшого застосування в усіх сферах людської діяльності. Сформульовано інженерноорієнтовані визначення видів комп'ютингу, у тому числі квантового, який використовує поняття суперпозиції і змішування, а також комп'ютингу в пам'яті. Обумовлено необхідність спільного і паралельного вирішення проблеми створення ринково доступного квантового комп'ютера і розробки кванто-орієнтованих додатків і хмарних сервісів. Наведено приклади квантового проектування і тестування фрагментів цифрових схем. Запропоновано метод синтезу і мінімізації тестів для функціональностей з використанням матриці кубітних похідних і секвенсор для віднаходження квазіоптимального покриття.

*К л ю ч о в і с л о в а: синтез тестів, кубітне покриття, комп'ютинг на пам'яті, цифрова схема, булева кубітна похідна, моделювання несправностей.*

Существует следующее определение квантового компьютера: это — вычислительное устройство, в котором используются явления квантовой

© В.И. Хаханов, И.В. Емельянов, М.М. Любарский, С.В. Чумаченко, Е.И. Литвинова, 2018

суперпозиции и квантовой запутанности для передачи и обработки данных [1, 2]. В целом определение правильное, но в нем не должно быть передачи данных, поскольку она не является функцией двух указанных свойств. В [1, 2] отсутствуют четкие пояснения словосочетаний, используемых в определении. Относительно преемственности квантового и классического компьютеринга рыночная ситуация подобна революционной. Первый еще не может занять достойное место в развивающейся нише аддитивных технологий, а второй скоро не сможет соответствовать закону Мура, имеющему технологический предел 2,5 нм разрешения линий на силиконовом кристалле [3].

Предлагаются определения отношений между квантовым и классическим компьютерингом, формирующие сходства и различия по метрике структур данных и вычислительных процессов.

*Квантовый компьютер* — вычислительное устройство, в котором используется структура элементарных частиц (электронов, атомов) для выполнения квантовых транзакций.

*Квантовый компьютеринг* — вычислительный процесс выполнения квантовых транзакций на структурах элементарных частиц (электронов), реализующих программные инструкции на основе операций суперпозиции и запутанности, которые формируют квантовый логический базис для обработки данных.

*Квантовая транзакция* — процесс считывания-записи данных на структуре элементарных частиц (электронов) при выходе и приеме кванта электромагнитного излучения.

*Квантовая суперпозиция* — объединение в пространстве более одного дискретного состояния. Одно из основных правил дискретной математики гласит, что любой объект, компонент или система может находиться одновременно в нескольких дискретных состояниях. Это может быть описано алгеброй (алфавитом) Кантора, которая использует в пределе  $n$  примитивных символов для одновременного представления  $n$  состояний объекта. Более того, диаграмма Хассе, построенная с помощью любого алфавита Кантора, есть не что иное как параллельный процессор или секвенсор для одновременного получения множества решений, например, при рассмотрении задачи покрытия.

*Квантовая запутанность* — дополнение к состоянию элементарной частицы, входящей в универсум. Естественно, что дополнение к любому подмножеству примитивов универсума создает взаимозависимость двух его частей, каждая из которых обладает полной информацией о своем дополнении.

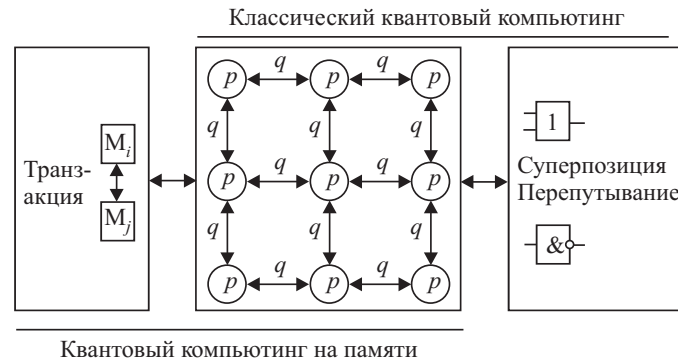


Рис. 1. Классический и квантовый компьютеринг на памяти

О появлении квантовых компьютеров и возрастании популярности вычислений в памяти свидетельствуют следующие факты.

1. На рынке появилось несколько квантовых компьютеров (IBM, D-wave, Google, NASA) на основе суперпозиции-запутанности (or-not) элементарных частиц, что соответствует классической теории проектирования компьютеров (рис. 1).

2. Возможно, не следует создавать или искать в природе структуры логического базиса. Они становятся избыточными с позиции квантового компьютеринга на памяти (см. рис. 1). Проще найти структуры памяти в любой форме существования материи для реализации квантовых транзакций считывания-записи. Однако это уже будет разрушением классической теории проектирования компьютеров.

3. Memory-driven компьютеринг становится доминирующим на рынке обработки больших данных. Программируемая архитектура «FPGA Family Xilinx® UltraScale™» принадлежит классу СБИС и обеспечивает производительность реализованных смарт-функциональностей на уровне сотен гигабит в секунду. Компания Hewlett Packard Enterprise (HPE) создала прототип суперкомпьютера «The Machine» — компьютеринг архитектуру без использования логики на основе фотонно-оптических соединений, которая на четыре порядка (в 8000 раз) превосходит по производительности существующие компьютеры [4]. Для вычислений используется 8 терабайт памяти, созданной на мемристорах, что в десятки раз превосходит мощности современных серверов. Компания полагает с помощью «The Machine» решить проблемы «больших данных»: транспортные системы, умные города, распознавание голоса и видео, безопасность, здоровье и др. Идея проста и технологична — перенести все вычисления в практи-

чески безразмерную память, убрать из архитектуры компьютера логический процессор, решив при этом проблему «бутылочного горлышка» — шинного взаимодействия памяти и логики. При этом производительность обработки больших данных при решении упомянутых рыночных проблем в компьютеринге на памяти возрастает на один-два порядка.

Согласно принципу Гейзенберга [2], частица (электрон) может находиться в двух точках пространства одновременно, как и два электрона могут одновременно находиться в одной точке пространства. Это означает, что два дискретных состояния (электрона) могут накладываться друг на друга в одной точке пространства, образуя их суперпозицию. Математически естественно и тривиально, что разделение частицы, составляющей универсум состояний, на две взаимодополняющих субчастицы в пространстве обеспечивает знание состояния каждой из них, образуя их перепутывание.

Квантовое пересечение — определение в пространстве общей части для более одного дискретного состояния. Данной операции соответствует скалярное произведение кет-векторов в гильбертовом пространстве, которое равно нулю, если они ортогональны. Данная операция не фигурирует в определениях квантового компьютеринга, возможно, по причине ее избыточности. Тем не менее, квантовые операции (суперпозиции, перепутывания и пересечения), являясь избыточными с позиции компьютеринга на памяти, повышают эффективность квантовых вычислений [1, 2, 5—7].

Следовательно, для создания классического квантового компьютера необходимо построить устойчивую во времени структуру (носитель) элементарных частиц, на которой будут определены операции сигнатуры: суперпозиция (or), перепутывание (not), пересечение (and). Транзакции на памяти, реализующие все логические функции, разрушают один из основных принципов теории вычислительных процессов, а именно нельзя создать универсальный вычислитель без элемента инверсии.

Предлагается соединение трех тенденций в области современного компьютеринга: квантовый компьютеринг, компьютеринг на памяти и облачные структуры для повышения производительности решения задач, связанных с проектированием, тестированием и верификацией цифровых систем на кристаллах. Глобальная цель — создание нового квантового компьютера, использующего парадигму <Память — Адрес — Транзакция> без свойств суперпозиции и перепутывания [8—10], что дает возможность существенного расширения условий для более технологичной и простой реализации вычислителя в практически любой форме существования материи (твердой, жидкой, газообразной, плазменной).

Мотивирующим аргументом для создания квантового компьютеринга на памяти является наличие на рынке электроники следующих очевидных проблем:

1. Уже существует квантовый компьютер, но отсутствует компьютинг, предполагающий создание новых квантовых технологий программирования моделей, методов и алгоритмов, ориентированных на параллельное решение традиционных и новых задач.

2. Существуют эффективные аппаратные решения, например от компании Apple, но пока отсутствуют такого же качества программные приложения и облачные сервисы.

3. Существуют на рынке микроэлектроники тенденции ухода от структуры <АЛУ (арифметико-логическое устройство) — шины данных — память> в сторону реализации регулярных вычислительных архитектур типа компьютинг на памяти. Есть желание передовых технологических компаний уйти от арифметики и логики процессоров. Но имеется сильное лобби в среде ведущих электронных фирм планеты, не заинтересованное в закрытии данного направления.

4. Существуют специалисты и ученые, не до конца понимающие, как можно синтезировать арифметические и логические устройства только на элементах памяти.

5. Доставлять инструкции по обработке больших данных в память намного эффективней, чем перекачивать информацию из памяти в АЛУ и обратно по бутылочному горлышку шины данных, что существенно снижает производительность вычислительного устройства.

В настоящее время представляется актуальным решение проблемы создания и апробирования на классических компьютерах высокопроизводительной теории и методов квантового компьютинга для их последующего применения во всех сферах человеческой деятельности. Иначе ученым необходимо совместно и параллельно решать проблемы создания рыночно доступного квантового компьютера и разработки кванто-ориентированных теорий и приложений (облачных сервисов).

Определения типов компьютеров и компьютинга в зависимости от области применения и базиса исполнения архитектуры:

Компьютер (классический) — радиоэлектронная цифровая система, объединяющая процессор, память и периферию для программного управления выполнением арифметико-логических инструкций.

Компьютер (новый) — киберфизическая система интеллектуального управления архитектурой и высокопроизводительным исполнением программных инструкций.

Компьютинг (системный) — интеллектуальное актюаторное управление киберфизической архитектурой на основе сенсорного мониторинга данных для высокопроизводительного исполнения программных инструкций.

Компьютинг (физический) — процесс выполнения адресных транзакций в структуре памяти.

Компьютинг (кибер) — процесс выполнения адресных транзакций в структуре данных.

Компьютинг (космологический) — процесс выполнения квантовых транзакций в пространственно-временной структуре материально-энергетической сущности для выполнения программы гармонического развития Вселенной.

**Проектирование сумматора и синтез теста без использования логики.** Покажем технологичность и эффективность использования квантовых кубитных структур данных для синтеза и минимизации тестов на основе кубитных булевых производных.

Задачи:

1. Синтез кубитных структур данных для memory-driven описания логических схем на примере одноразрядного сумматора.

2. Синтез тестов на основе использования кубитных производных входных переменных логической схемы.

3. Минимизация полученных тестов посредством решения задачи покрытия при использовании полученной матрицы кубитных производных.

Одноразрядный сумматор представляет собой примитивный логический блок для получения суммы и переноса при сложении двоичных чисел. Таблицы истинности, кубитные покрытия и логические функции двух отдельных модулей сумматора, формирующих сумму и перенос, имеют следующий вид:

<table style="border-collapse: collapse; width: 100%;"> <tr><th><i>a</i></th><th><i>b</i></th><th><i>c</i></th><th><i>S</i></th><th><i>P</i></th></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> </table>	<i>a</i>	<i>b</i>	<i>c</i>	<i>S</i>	<i>P</i>	0	0	0	0	0	0	0	1	1	0	0	1	0	1	0	0	1	1	0	1	1	0	0	1	0	1	0	1	0	1	1	1	0	0	1	1	1	1	1	1	→	<table style="border-collapse: collapse; width: 100%;"> <tr><th><i>S</i></th><th><i>P</i></th></tr> <tr><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td></tr> </table>	<i>S</i>	<i>P</i>	0	0	1	0	1	0	0	1	1	0	0	1	0	1	1	1	→	$S = \bar{a}\bar{b}c \oplus \bar{a}b\bar{c} \oplus a\bar{b}\bar{c} \oplus abc,$ $P = \bar{a}bc \oplus a\bar{b}c \oplus ab\bar{c} \oplus abc.$
<i>a</i>	<i>b</i>	<i>c</i>	<i>S</i>	<i>P</i>																																																															
0	0	0	0	0																																																															
0	0	1	1	0																																																															
0	1	0	1	0																																																															
0	1	1	0	1																																																															
1	0	0	1	0																																																															
1	0	1	0	1																																																															
1	1	0	0	1																																																															
1	1	1	1	1																																																															
<i>S</i>	<i>P</i>																																																																		
0	0																																																																		
1	0																																																																		
1	0																																																																		
0	1																																																																		
1	0																																																																		
0	1																																																																		
0	1																																																																		
1	1																																																																		

Структуры данных (вектор моделирования  $M$ , списки входных переменных  $X$  и кубитные векторы-покрытия  $Q$ ) для анализа сумматора имеют вид, представленный на рис. 2. Все данные помещаются в структуры памяти, которые с использованием адресуемости компонентов и ячеек могут быть модифицированы в режиме онлайн, а также отремонтированы

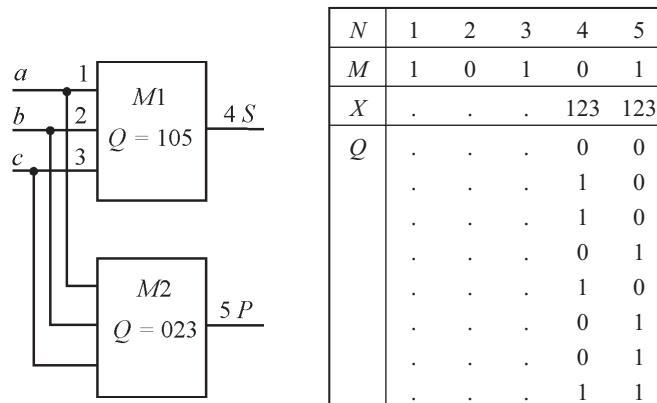


Рис. 2. Структуры данных для реализации сумматора на памяти

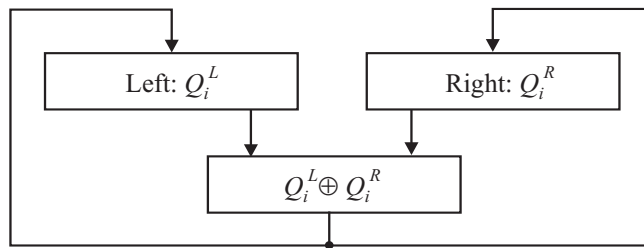


Рис. 3. Схема получения кубитной булевой производной [11]

в случае возникновения отказов. Таким образом, использование адресуемых элементов памяти при реализации цифровых схем дает возможность избавиться от логических элементов, а также от сложных проблем тестирования, верификации и ремонта цифровых систем.

В основе синтеза и анализа тестов для цифровых функциональностей лежит взятие булевой производной по кубитному вектору, которое сводится к одновременному выполнению всего лишь двух регистровых операций (хот — write), схематически представленных на рис. 3 [11]. По сути, производная есть условие для активизации заданной входной переменной, представляемое в виде функции, где фигурируют все переменные за исключением заданной, по которой берется производная:

$$\frac{df(x_1, x_2, \dots, x_i, \dots, x_n)}{dx_i} =$$

$$= f(x_1, x_2, \dots, x_i = 0, \dots, x_n) \oplus f(x_1, x_2, \dots, x_i = 1, \dots, x_n).$$

Суть кубитной производной заключается в приведении структуры взаимодействия входных переменных к формату вектора кубитного покрытия функциональности. Более того, кубитная производная создает определенную симметрию скрытого или неявного хог-взаимодействия входных переменных посредством выполнения операции взятия производной на двух частях  $Q$ -покрытия:

$$Q'(X) = \{Q_i^L, Q_i^R\} = Q_i^L \oplus Q_i^R, \quad i = \overline{1, 2^{k-1}}, \quad k = \overline{1, n}. \quad (1)$$

Таким образом,  $n$  кубитных производных создают условия активизации всех  $n$  переменных, которые записываются уже в форме матрицы или совокупности соответствующих векторов. Иными словами, кубитная производная с помощью единичных значений своих координат представляет собой скрытую форму теста для проверки всех логических путей от рассматриваемой переменной до выхода функциональности. С учетом изложенного синтеза теста, в данном случае для сумматора, можно выполнить только на основе использования кубитного вектора. Формула (1) взятия булевых производных по всем переменным функциональности [1] дает возможность получить тест посредством регистрового объединения полученных кубитных производных:

$$T(Q') = \bigoplus_{i=1}^n Q'(X_i). \quad (2)$$

Используя формулу (2), можно достаточно просто синтезировать тест в формате кубитного вектора, который в рассматриваемом случае равен восьми двоичным наборам:

$Q'(X) \setminus Q$	0	1	1	0	1	0	0	1
$Q'(X_1)$	1	1	1	1	1	1	1	1
$Q'(X_2)$	1	1	1	1	1	1	1	1
$Q'(X_3)$	1	1	1	1	1	1	1	1
$T(Q') = \bigoplus_{i=1}^n Q'(X_i)$	1	1	1	1	1	1	1	1

Единичные координаты полученной матрицы кубитных производных означают, что любой входной набор по всем координатам активизирует (проверяет) одиночные константные неисправности, инверсные по отношению к исправному состоянию переменных. Данное свойство сумматора является позитивным для диагностирования, но его можно рассматривать как негативное по метрике синтеза минимального теста. Минимизация теста может быть реализована на основе моделирования неисправностей, что дает возможность построить таблицу покрытия неисправностей и выбрать минимальное число тест-векторов, покрывающих все одиночные де-



фекты. Для этого необходимо построить дедуктивные формулы анализа одиночных константных неисправностей [9].

Другой подход основан на анализе матрицы векторов булевых кубитных производных, для которой необходимо найти минимальное число единиц (столбцов), покрывающих единичными значениями все строки или производные входных переменных. Для последней матрицы такие 1-столбцы, составляющие минимальный тест, которые активизируют все одиночные константные дефекты на входных и выходных переменных, представлены двумя решениями:

$Q'(X)\backslash Q$	0 1 1 0 1 0 0 1	$Q'(X)\backslash Q$	0 1 1 0 1 0 0 1
$Q'(X_1)$	1 . . . . . 1	$Q'(X_1)$	. . . 1 1 . . .
$Q'(X_2)$	1 . . . . . 1	$Q'(X_2)$	. . . 1 1 . . .
$Q'(X_3)$	1 . . . . . 1	$Q'(X_3)$	. . . 1 1 . . .
$T(Q) = \bigoplus_{i=1}^n Q'(X_i)$	1 0 0 0 0 0 0 1	$T(Q) = \bigoplus_{i=1}^n Q'(X_i)$	0 0 0 1 1 0 0 0

Формально входные наборы {000-00, 111-11} или {011-01, 100-10} проверяют все одиночные константные неисправности для функциональности суммы  $S$ . Если аналогично построить тест для кубитного покрытия функции переноса  $P$ , то окажется, что пара входных наборов {000-00, 111-11} не является проверяющим тестом для второй функциональности  $S$  одноразрядного сумматора:

$Q'(X)\backslash Q$	0 0 0 1 0 1 1 1
$Q'(X_1)$	0 0 1 1 1 1 0 0
$Q'(X_2)$	0 1 0 1 1 0 1 0
$Q'(X_3)$	0 1 1 0 0 1 1 0
$T(Q) = \bigoplus_{i=1}^n Q'(X_i)$	0 1 1 1 1 1 1 0

Это связано с тем, что не каждое изменение одной входной координаты (0-1, 1-0) на тестовых наборах 000-00 и 111-11 приведет к изменению состояния выхода  $P$ .

В общем случае метод синтеза минимального теста по матрице кубитных производных для любого «черного ящика» функциональности связан с решением задачи покрытия, а именно:

разбиение множества столбцов  $Q$ -матрицы производных на нулевые и единичные по соответствующим значениям координат исходного  $Q$ -покрытия функциональности;

определение минимального числа 0-столбцов, покрывающих все строки единичными значениями своих координат;

определение минимального числа 1-столбцов, покрывающих все строки единичными значениями своих координат;

объединение полученных минимальных покрытий для подмножества 0- и 1-столбцов позволяет получить минимальный тест проверки одиночных константных неисправностей для любой функциональности черного ящика.

Выполнение описанного метода синтеза минимального теста для функциональностей черного ящика сводится к выполнению процедур, представленных в следующей формуле:

$$T = \min [T(Q^0) \oplus T(Q^1)] = \min \left[ \left( \bigoplus_{\substack{i=\overline{1,n} \\ \forall i(Q_i=0)}} Q_{ij} \right) \oplus \left( \bigoplus_{\substack{i=\overline{1,n} \\ \forall i(Q_i=1)}} Q_{ij} \right) \right]. \quad (3)$$

Использование формулы (3) при синтезе тестов для функциональности переноса  $P$  одноразрядного сумматора на основе полученной матрицы кубитных производных дает следующий результат:

$Q'(X) \setminus Q$	0	0	0	1	0	1	1	1
$Q'(X_1)$	.	.	1	1	1	1	.	.
$Q'(X_2)$	.	.	0	1	1	0	.	.
$Q'(X_3)$	.	.	1	0	0	1	.	.
$\min [T(Q^0) \oplus T(Q^1)]$	0	0	1	1	1	1	0	0

При совместной минимизации теста для двух функциональных элементов одноразрядного сумматора формируется оптимальный результат, имеющий четыре тестовые последовательности:

$Q'(S) \setminus Q(S)$	0	1	1	0	1	0	0	1
$Q'(X_1)$	.	.	.	1	1	.	.	.
$Q'(X_2)$	.	.	.	1	1	.	.	.
$Q'(X_3)$	.	.	.	1	1	.	.	.
$T(S)$	0	0	0	1	1	0	0	0

 $\oplus$ 

$Q'(P) \setminus Q(P)$	0	0	0	1	0	1	1	1
$Q'(X_1)$	.	.	1	1	1	1	.	.
$Q'(X_2)$	.	.	0	1	1	0	.	.
$Q'(X_3)$	.	.	1	0	0	1	.	.
$T(P)$	0	0	1	1	1	1	0	0

 $=$ 

$$= T(S) \oplus T(P) = (00111100).$$

Таким образом, метод синтеза минимального теста на основе взятия кубитных производных отличается от известных тем, что здесь не рассматриваются входные и выходные переменные функциональности, а все

операции используют только кубитные векторы и их производные. Вычислительная сложность взятия кубитных производных имеет линейную зависимость от числа переменных. Однако процедура минимизации теста, приведенная к задаче покрытия, имеет степенную зависимость от числа переменных, которая может быть существенно уменьшена посредством разбиения матрицы кубитных производных и параллельного нахождения квазиоптимальных покрытий для фрагментов матрицы [3].

**Метод получения квазиоптимального теста или покрытия** является составной частью синтеза минимального теста цифровых схем. Он основан на использовании регистровых параллельных операций над аппаратно-ориентированными структурами данных, которые представляют собой матрицу кубитных производных для функциональности черного ящика. Архитектура секвенсора для реализации метода представлена на рис. 4. Аналитическая модель и вычислительные процедуры для минимизации проверяющего теста на основе использования матрицы кубитных покрытий имеют следующий вид:

$$A = \langle Q, Q', Q^0, Q^1, T, q^0, q^1, h^0, h^1, p \rangle,$$

$$1) Q = (Q_1, Q_2, \dots, Q_i, \dots, Q_{2^n});$$

$$2) Q' = [Q'_{ij}], i = \overline{1, 2^n}, j = \overline{1, n};$$

$$3) Q^0 = (Q_1^0, Q_2^0, \dots, Q_j^0, \dots, Q_n^0);$$

$$4) Q^1 = (Q_1^1, Q_2^1, \dots, Q_j^1, \dots, Q_n^1);$$

$$5) T = (T_1, T_2, \dots, T_i, \dots, T_{2^n});$$

$$6) q^0 = (Q_1^0 \oplus Q_2^0 \oplus \dots \oplus Q_j^0 \oplus \dots \oplus Q_n^0);$$

$$7) q^1 = (Q_1^1 \oplus Q_2^1 \oplus \dots \oplus Q_j^1 \oplus \dots \oplus Q_n^1);$$

$$8) h^0 = 1 \leftrightarrow q^0 = 1;$$

$$9) h^1 = 1 \leftrightarrow q^1 = 1;$$

$$10) p = \bigoplus_{j=1}^n [(Q_j^{0(1)} \oplus Q'_{ji}) \oplus Q'_{ji}].$$

Здесь 1 — кубитное покрытие функциональности черного ящика; 2 — матрица кубитных производных по всем переменным; 3 и 4 — буферный накопительный регистр для индикации процесса получения квазиоптимального покрытия по нулевым и единичным координатам  $Q$ -вектора; 5 — кубитный тест-вектор, где единичными координатами отмечены тестовые

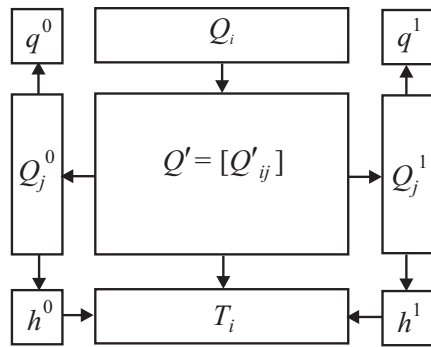


Рис. 4. Архитектура секвенсора для определения квазиоптимального теста

наборы, которые необходимо подавать на тестируемое устройство; 6 и 7 — индикаторы достижения квазиоптимального покрытия по нулевым и единичным координатам  $Q$ -вектора; 8 и 9 — переключатели окончания анализа столбцов матрицы производных по нулевыми и единичным координатам  $Q$ -вектора; 10 — указатель приращения степени покрываемости строк матрицы кубитных производных для рассматриваемого столбца. Если указатель равен нулю (приращения нет), то столбец из нулевого или единичного подмножества матрицы не включается в тест:

$$p = \bigoplus_{i=1}^n [(Q^{0(1)} \oplus Q'_i) \oplus Q'_i].$$

Структурная схема алгоритма синтеза квазиоптимального теста на основе разбиения матрицы производных имеет две симметричные ветви, каждая из которых ориентирована на анализ нулевого и единичного подмножества столбцов (рис. 5). Основная идея получения квазиоптимального теста заключается в определении минимального числа столбцов в нулевом и единичном подмножествах матрицы кубитных производных, которые своими единичными координатами покроют все строки или переменные рассматриваемой функциональности. При этом, если очередной столбец не добавляет проверяющих свойств к ранее включенным в тест векторам, то он исключается из списка.

Рассмотрим использование описанных процедур метода синтеза квазиминимального теста для логической схемы Шнейдера, представленной на рис. 8 в работе [11]. Кубитное покрытие цифровой схемы представлено вектором (100000000000001), по которому были получены четыре кубитных производных. Существенно, что для каждой переменной кубитная производная составлена парами единичных координат в векторе. В совокупности каждая пара при выполнении ог-операции дает функциональный терм из трех входных переменных, где отсутствует четвертая, по которой берется производная. Например, производная  $Q'(X_1) = 11$  на адресных координатах входных переменных 0000 и 0001 означает условия активизации 000 для первой переменной. Таким образом, кубитная производная, как

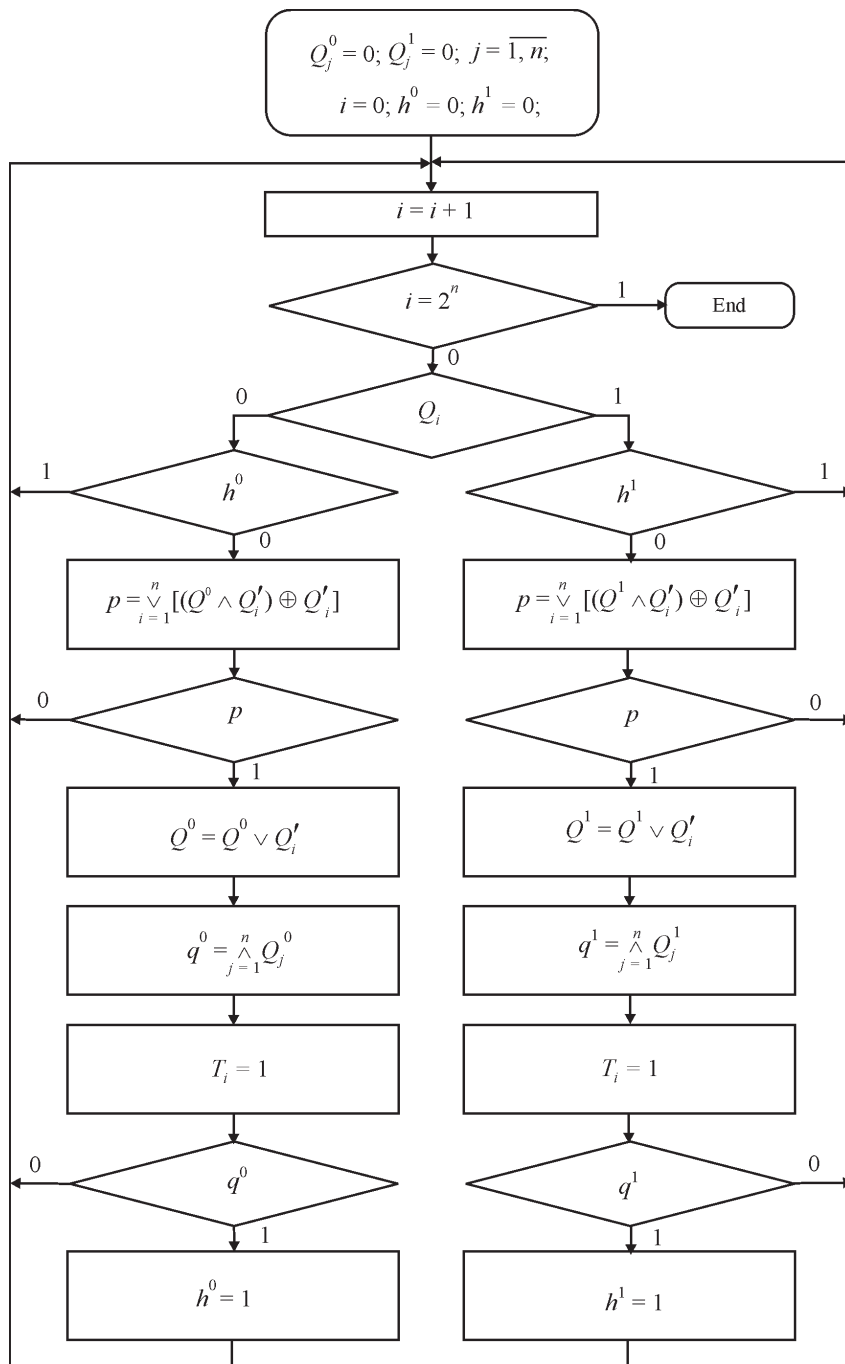


Рис. 5. Алгоритм поиска квазиоптимального теста

пара входных наборов, представляет собой тест для проверки одиночных константных неисправностей рассматриваемой входной переменной. Естественно, что таких пар для каждой входной линии может быть несколько. В этом случае их использование связано с проверкой внутренних переменных функциональности.

В табл. 1 приведены кубитное покрытие, булевы производные и состояния входных переменных, соответствующие значениям координат кубитного покрытия (для визуального восприятия общей картины данных точками отмечены нулевые состояния координат таблицы истинности и матрицы производных). В результате выполнения ог-операции над векторами кубитных производных получен тест, содержащий 10 входных наборов  $T = (1110100110010111)$ . Он является тестом диагностирования (избыточным проверяющим) для одиночных константных неисправностей цифро-

Таблица 1

$X_1$	.	.	.	.	.	.	.	.	1	1	1	1	1	1	1	1
$X_2$	.	.	.	.	1	1	1	1	.	.	.	.	1	1	1	1
$X_3$	.	.	1	1	.	.	1	1	.	.	1	1	.	.	1	1
$X_4$	.	1	.	1	.	1	.	1	.	1	.	1	.	1	.	1
$Q$	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	1
$Q'(X_1)$	1	.	.	.	.	.	.	1	1	.	.	.	.	.	.	1
$Q'(X_2)$	1	.	.	.	1	.	.	.	.	.	.	1	.	.	.	1
$Q'(X_3)$	1	.	1	.	.	.	.	.	.	.	.	.	.	1	.	1
$Q'(X_4)$	1	1	.	.	.	.	.	.	.	.	.	.	.	.	1	1
Test =	1	1	1	.	1	.	.	1	1	.	.	1	.	1	1	1
$T_{min} =$	1	1	.	.	.	.	.	1	1	.	.	.	.	.	1	1

Таблица 2

TEST	1	2	3	4	5	6	7	8	9	10	11	12	FD	FC
0 0 0 0 1 1 1 0 0 0 0 0 1	1	1	1	1	0	0	0	1	1	1	1	0	50	50
0 0 0 1 1 1 0 0 0 0 0 1 0	.	.	.	0	.	.	1	.	.	.	0	1	16	66
0 1 1 1 0 0 0 0 1 0 0 0 0	.	.	.	.	.	1	.	.	0	.	.	1	12	75
1 0 0 0 0 1 1 1 0 0 0 0 0	0	.	.	.	1	.	.	0	.	.	.	1	16	87
1 1 1 0 0 0 0 0 0 0 1 0 0	.	.	.	.	.	1	.	.	.	0	.	1	12	91
1 1 1 1 0 0 0 0 0 0 0 0 1	0	0	0	0	.	.	.	1	1	1	1	0	37	100
$\oplus =$	X	X	X	X	X	X	X	X	X	X	X	X		

вой логической схемы. Процедура минимизации теста, с учетом структуры схемы, путем нахождения квазимиимального покрытия всех входных переменных нулевыми и единичными подмножествами кубитных производных выдает результат из шести тестовых наборов, приведенный в табл. 2, где точками отмечены координаты с непроверяемыми неисправностями.

Таким образом, минимальный тест для логической схемы Шнейдера содержит всего шесть входных последовательностей, которые проверяют 100 процентов одиночных константных неисправностей для входных, внутренних и выходных линий устройства.

Результаты проведенных экспериментов минимизации тестов над 16 фрагментами цифровых устройств (4—10 входных переменных) следующие:

в 25 % случаев получены оптимальные тесты;

в 70 % случаев тесты отличались от оптимальных не более, чем на 25 %;

в 5 % случаев тесты были близки к максимальному числу входных последовательностей;

вычислительная сложность предложенного кубитного метода синтеза минимального теста для логической функциональности от  $n$  переменных определяется оценкой, формирующей временные затраты на взятие кубитных производных и минимизацию теста:  $E = 2n + 2n^2 = 2n(n + 1)$ .

Таким образом, метод синтеза тестов на основе использования кубитных производных позволяет сгенерировать входные наборы, проверяющие все одиночные константные неисправности входных и внутренних линий. Однако для синтеза минимального теста необходимо использовать структуру цифрового устройства.

## Выводы

Предложенный метод, алгоритм и структура секвенсора для синтеза и минимизации тестов функциональностей черного ящика основана на использовании матрицы кубитных производных для нахождения квазиоптимального покрытия.

Обусловлена необходимость совместного и параллельного решения проблемы создания рыночно доступного квантового компьютера и разработки кванто-ориентированных приложений и облачных сервисов.

Проведенные эксперименты на фрагментах цифровых схем показали практическую значимость и технологичность предложенной архитектуры и метода синтеза квазимиимального теста для логических функциональностей черного ящика.

Дальнейшие исследования будут направлены на создание семейства интеллектуальных алгоритмов квантового синтеза тестов, моделирования и диагностирования неисправностей с помощью аппарата кубитных производных.

#### СПИСОК ЛИТЕРАТУРЫ

1. *Almudever C.G. et al.* The engineering challenges in quantum computing. Design, Automation. Test in Europe Conference & Exhibition (DATE), 2017. Lausanne, 2017, p. 836—845.
2. *Nielsen M.A., Chuang I.L.* Quantum Computation and Quantum Information. Cambridge University Press, 2010.
3. *Williams R.S.* What's Next? [The end of Moore's law] // Computing in Science & Engineering, 2017, Vol. 19, No. 2, p. 7—13.
4. [Электронный ресурс]. — Режим доступа: <https://www.labs.hp.com/next-next/mdc>.
5. *Singh J., Singh M.* Evolution in Quantum Computing. Int. Conf. System Modeling and Advancement in Research Trends (SMART). Moradabad, 2016, p. 267—270.
6. *Shaikh T.A., Ali R.* Quantum Computing in Big Data Analytics: A Survey. IEEE Int. Conf. on Computer and Information Technology (CIT). Nadi, 2016, p. 112—115.
7. *Vandersypen L., Van Leeuwenhoek A.* 1.4 Quantum computing — the next challenge in circuit and system design. IEEE International Solid-State Circuits Conference (ISSCC). San Francisco, CA, 2017, p. 24—29.
8. *Hahanov V. et al.* Qubit test synthesis of the functionality, 14th International Conf. The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM). Lviv, 2017, p. 251—255.
9. *Hahanov I., Chumachenko S., Iemelianov I. et al.* Deductive qubit fault simulation, 14th Int. Conf. The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM). Lviv, 2017, p. 256—259.
10. *Хаханов В.И., Тимер Бани Амер, Чумаченко С.В., Литвинова Е.И.* Кубитные технологии для анализа и диагностирования цифровых устройств, Электрон. моделирование, 2015, **37**, № 3, с. 17—40.
11. *Хаханов В.И., Емельянов И.В., Любарский М.М. и др.* Кубитный метод дедуктивного анализа неисправностей для логических схем // Электрон. моделирование, 2017, **39**, № 6, с. 59—91.

Поступила 14.07.17

#### REFERENCES

1. Almudever, C.G. et al. (2017), The engineering challenges in quantum computing, *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, Lausanne, pp. 836-845.
2. Nielsen, M.A. and Chuang, I.L. (2010), *Quantum computation and quantum information*, Cambridge University Press, Cambridge, UK.
3. Williams, R.S. (2017), What's Next? [The end of Moore's law], *Computing in Science & Engineering*, Vol. 19, no. 2, pp. 7-13.
4. Available at: <https://www.labs.hp.com/next-next/mdc>.
5. Singh, J. and Singh, M. (2016), Evolution in quantum computing, *Int. Conference on System Modeling & Advancement in Research Trends (SMART)*, Moradabad, 2016, pp. 267-270.
6. Shaikh, T.A. and Ali, R. (2016), Quantum computing in big data analytics: A Survey, *IEEE Int. Conference on Computer and Information Technology (CIT)*, Nadi, 2016, pp. 112-115.



7. Vandersypen, L. and van Leeuwenhoek, A. (2017), 1.4 Quantum computing - the next challenge in circuit and system design, *IEEE International Solid-State Circuits Conference (ISSCC)*, San Francisco, CA, 2017, pp. 24-29.
8. Hahanov, V. et al. (2017), Qubit test synthesis of the functionality, *The 14th International Conference The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM)*, Lviv, 2017, pp. 251-255.
9. Hahanov, I., Chumachenko, S., Iemelianov, I., Hahanov, V., Larchenko, L. and Daniyil, T. Deductive qubit fault simulation, *The 14th Int. Conference The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM)*, Lviv, 2017, pp. 256-259.
10. Hahanov, V.I., Tamer Bani Amer, Chumachenko, S.V. and Litvinova, E.I. (2015), "Qubit technologies for analysis and diagnostics of digital devices", *Elektronnoe modelirovanie*, Vol. 37, no. 3, pp. 17-40.
11. Hahanov, V.I., Iemelianov, I.V., Liubarskyi, M.M. et al. (2017), "Qubit method of deductive analysis of failures for logical schemes", *Elektronnoe modelirovanie*, Vol. 39, no. 6, pp. 59-91.

Received 14.07.17

*V.I. Hahanov, I.V. Iemelianov,  
M.M. Liubarskyi, S.V. Chumachenko, E.I. Litvinova*

#### QUANTUM MEMORY-DRIVEN METHOD FOR TEST SYNTHESIS BASED ON QUBIT DATA STRUCTURES

One of the possible solutions to the problem of creating and testing the theory and methods of quantum memory-driven computing on the classical computers for their subsequent application in all fields of human activity is proposed. Engineering-focused definitions of computing types, including quantum ones, are used, including the notions of superposition and entanglement, and also memory-driven computing. The necessity of joint and parallel solution of the problem of creation of a market-accessible quantum computer and development of quantum-focused applications and cloud services is explained. Examples of quantum memory-driven design and test of digital circuit fragments are presented. A method for synthesizing and minimizing tests for black-box functionality is proposed, using a matrix of qubit derivatives and a sequencer for defining a quasi-optimum coverage.

*Keywords: test synthesis, qubit coverage, memory-driven computing, digital circuit, Boolean qubit derivative, fault simulation.*

*ХАХАНОВ Владимир Иванович, д-р техн. наук, профессор, гл. науч. сотр., профессор кафедры автоматизации проектирования вычислительной техники Харьковского национального университета радиоэлектроники. Область научных исследований — компьютерная инженерия, киберфизические системы и облачный компьютеринг.*

*ЕМЕЛЬЯНОВ Игорь Валерьевич, аспирант кафедры автоматизации проектирования вычислительной техники Харьковского национального университета радиоэлектроники. Область научных исследований — компьютерные системы и сервис-компьютеринг.*

*ЛЮБАРСКИЙ Михаил Михайлович, аспирант кафедры автоматизации проектирования вычислительной техники Харьковского национального университета радиоэлектроники. Область научных исследований — компьютерные системы и сервис-компьютинг.*

*ЧУМАЧЕНКО Светлана Викторовна, д-р техн. наук, профессор, зав. кафедрой автоматизации проектирования вычислительной техники Харьковского национального университета радиоэлектроники. Область научных исследований — математическое моделирование вычислительных процессов.*

*ЛИТВИНОВА Евгения Ивановна, д-р техн. наук, профессор, профессор кафедры автоматизации проектирования вычислительной техники Харьковского национального университета радиоэлектроники. Область научных исследований — проектирование и тестирование цифровых систем и сетей на кристаллах.*