

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ

ХАРЬКОВСКИЙ НАЦИОНАЛЬНЫЙ
УНИВЕРСИТЕТ РАДИОЭЛЕКТРОНИКИ

ISSN0135-1710

АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ УПРАВЛЕНИЯ И ПРИБОРЫ АВТОМАТИКИ

**Всеукраинский межведомственный
научно-технический сборник**

Основан в 1965 г.

Выпуск 164

Харьков
2013

В сборнике представлены результаты исследований, касающихся компьютерной инженерии, управления, технической диагностики, автоматизации проектирования, оптимизированного использования компьютерных сетей и создания интеллектуальных экспертных систем. Предложены новые подходы, алгоритмы и их программная реализация в области автоматического управления сложными системами, оригинальные информационные технологии в науке, образовании, медицине.

Для преподавателей университетов, научных работников, специалистов, аспирантов.

У збірнику наведено результати досліджень, що стосуються комп'ютерної інженерії, управління, технічної діагностики, автоматизації проектування, оптимізованого використання комп'ютерних мереж і створення інтелектуальних експертних систем. Запропоновано нові підходи, алгоритми та їх програмна реалізація в області автоматичного управління складними системами, оригінальні інформаційні технології в науці, освіті, медицині.

Для викладачів університетів, науковців, фахівців, аспірантів.

Редакционная коллегия:

В.В. Семенец, д-р техн. наук, проф. (гл. ред.); М.Ф. Бондаренко, д-р техн. наук, проф.; И.Д. Горбенко, д-р техн. наук, проф.; Е.П. Пуятин, д-р техн. наук, проф.; В.П. Тарасенко, д-р техн. наук, проф.; Г.И. Загарий, д-р техн. наук, проф.; Г.Ф. Кривуля, д-р техн. наук, проф.; Чумаченко С.В., д-р техн. наук, проф.; В.А. Филатов, д-р техн. наук, проф.; Е.В. Бодянский, д-р техн. наук, проф.; Э.Г. Петров, д-р техн. наук, проф.; В.Ф. Шостак, д-р техн. наук, проф.; В.М. Левыкин, д-р техн. наук, проф.; Е.И. Литвинова, д-р техн. наук, проф.; В.И. Хаханов, д-р техн. наук, проф. (отв. ред.).

Свидетельство о государственной регистрации
печатного средства массовой информации

КВ № 12073-944ПР от 07.12.2006 г.

Адрес редакционной коллегии: Украина, 61166, Харьков, просп. Ленина, 14, Харьковский национальный университет радиоэлектроники, комн. 321, тел. 70-21-326

© Харківський національний університет
радіоелектроніки, 2013

СОДЕРЖАНИЕ

ХАХАНОВ В.И., BAGHDADI AMMAR AWNI ABBAS, ЛИТВИНОВА Е.И., ХАХАНОВА И.В., ВРУБЛЕВСКИЙ Н.Н. КУБИТНЫЕ СТРУКТУРЫ ДАННЫХ ВЫЧИСЛИТЕЛЬНЫХ УСТРОЙСТВ.....	4
ГУБНИЦКАЯ Ю.С. МОДЕЛИ ЗАДАЧИ РАЗМЕЩЕНИЯ СТАТЕЙ И ОЦЕНИВАНИЯ КОМПОНОВОК ДЛЯ МЕТОДА ФОРМИРОВАНИЯ ПОЛОС ИЗДАНИЯ НА ДОПЕЧАТНОЙ СТАДИИ	20
ОКСАНИЧ А.П., КОГДАСЬ М.Г. АВТОМАТИЗИРОВАННЫЙ КОМПЛЕКС «МИКРОН-1» ДЛЯ ИЗМЕРЕНИЯ СЛОЖНЫХ ФОРМ ДЕФОРМАЦИЙ ПЛАСТИН АРСЕНИДА ГАЛЛИЯ.....	25
БАРАННИК В.В., ТУРЕНКО С.В. КОМБИНАТОРНАЯ МОДЕЛЬ ВЕКТОРА ДВУХКОМПОНЕНТНЫХ КОРТЕЖЕЙ ДЛЯ ОЦЕНКИ ИНФОРМАТИВНОСТИ УСЕЧЕННОЙ ЛИНЕАРИЗИРОВАННОЙ ТРАНСФОРМАНТЫ	33
РЯБУХА Ю.Н. ТРЕХМЕРНОЕ РАВНОМЕРНОЕ КОДИРОВАНИЕ ПЕРЕМЕННОЙ ДЛИНОЙ КОДА	38
КРИВОДУБСКИЙ О.А., ТЕРЕЩУК И.В. МНОЖЕСТВЕННЫЕ ПРЕДСТАВЛЕНИЯ И АЛГЕБРА ЛОГИЧЕСКИХ СВЯЗЕЙ ПРЕДМЕТНОЙ ОБЛАСТИ СППР БЮДЖЕТИРОВАНИЯ	42
РОГОЗА В.С., СЕРГЕЕВ-ГОРЧИНСКИЙ А.А. ВЫБОР МЕРЫ ПОДОБИЯ ЦИФРОВЫХ СИГНАЛОВ ДЛЯ СТОХАСТИЧЕСКОГО РАСЧЕТА ОПТИМАЛЬНЫХ ПАРАМЕТРОВ ЦИФРОВОГО ФИЛЬТРА	50
СТАСЮК В.В. ОСНОВИ СТВОРЕННЯ ЗАСОБІВ ВИМІРЮВАННЯ РІВНЯ МАГНІТНОГО ПОЛЯ ЗЕМЛІ. ІСНУЮЧІ РІШЕННЯ	58
РЯБЦЕВ В.Г., УТКИНА Т.Ю. СРЕДСТВА СЖАТИЯ ДИАГНОСТИЧЕСКОЙ ИНФОРМАЦИИ ПРИ САМОТЕСТИРОВАНИИ И РЕМОНТЕ МИКРОСХЕМ ПАМЯТИ.....	63
ЛИТВИН В.В. МЕТОД ВИДОБУВАННЯ ЗНАНЬ З ПРИРОДОМОВНИХ ТЕКСТІВ ДЛЯ АВТОМАТИЗОВАНОЇ РОЗБУДОВИ ОНТОЛОГІЙ.....	67
ЧАЛЫЙ С.Ф., БОГАТОВ Е.О. МОДЕЛЬ МНОГОУРОВНЕВОГО ПРЕДСТАВЛЕНИЯ БИЗНЕС-ПРОЦЕССОВ С НАБОРАМИ СПЕЦИАЛИЗИРОВАННЫХ ПРАВИЛ	72
ЧАЛЫЙ С.Ф., БУЦУКИНА И.Б. ПРЕДСТАВЛЕНИЕ ЗНАНИЙ В ФОРМЕ ПРОЦЕССОВ С ОГРАНИЧЕНИЯМИ В ЗАДАЧАХ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ПРОЦЕССОВ.....	76
ЧАЛЫЙ С.Ф., ШАБАНОВА-КУШНАРЕНКО Л.В. РЕЛЯЦИОННАЯ МЕТАСЕТЬ ВЫДЕЛЕНИЯ НЕЯВНЫХ ЗНАНИЙ	80
БОНДАРЕНКО М.А. ОПТИМІЗАЦІЯ АНІМАЦІЙНИХ ЗОБРАЖЕНЬ ПРИ РОБОТІ З WEB- ГРАФІКОЮ	84
ШТУКИН М.В. ТЕХНОЛОГИЯ ВЫБОРА РАЦИОНАЛЬНЫХ РЕШЕНИЙ ПРИ ЛИКВИДАЦИИ ПОСЛЕДСТВИЙ ЧРЕЗВЫЧАЙНЫХ СИТУАЦИЙ	89
РЕФЕРАТИ	96
ПРАВИЛА ОФОРМЛЕНИЯ СТАТЕЙ ДЛЯ АВТОРОВ НАУЧНО-ТЕХНИЧЕСКОГО СБОРНИКА.....	102

КУБИТНЫЕ СТРУКТУРЫ ДАННЫХ ВЫЧИСЛИТЕЛЬНЫХ УСТРОЙСТВ

Предлагаются кубитные модели и методы повышения быстродействия программных и аппаратных средств анализа цифровых устройств за счет увеличения размерности структур данных и памяти. Вводятся основные понятия, термины и определения, необходимые для имплементации квантовых вычислений в практику анализа виртуальных компьютеров. Описываются результаты исследований, касающихся проектирования и моделирования компьютерных систем в киберпространстве на основе использования двухкомпонентного автомата <память, транзакции>.

1. Введение

Рыночная привлекательность эмуляции квантовых методов вычислений при создании виртуальных (облачных) компьютеров (ВК) в киберпространстве основывается на использовании кубитных моделей данных, ориентированных на параллельное решение задач дискретной оптимизации за счет существенного повышения затрат памяти. Здесь не рассматриваются физические основы квантовой механики, касающиеся недетерминированного взаимодействия атомных частиц [1-4], но используется понятие кубита как двоичного или многозначного вектора для совместного и одновременного задания булеана состояний дискретной области киберпространства на основе линейной суперпозиции унитарных кодов, ориентированных на параллельное исполнение методов анализа и синтеза компонентов киберпространства.

Мотивация нового подхода для проектирования ВК связана с появлением облачных сервисов, которые представляют собой специализированные и рассредоточенные в пространстве виртуальные компьютерные системы, инвариантные по отношению к реализации в аппаратуре или в программном продукте. Иначе, сегодня уже не всегда интересно программисту как и куда имплементируются коды программных приложений, равно как и нет необходимости знать теорию проектирования цифровых автоматов на уровне вентилей, регистровых передач и использовать специфические компоненты вычислительной техники (триггеры, регистры, счетчики, мультиплексоры). Любой компонент функциональности покрывается векторной формой таблицы истинности, реализуемой с помощью памяти. Логические функции в традиционном исполнении здесь исключены из рассмотрения. От этого частично страдает быстродействие, но учитывая, что 94 процента SoC-кристалла сегодня есть память, то остальные 6 процентов также можно реализовать на памяти, что не будет критичным для большинства программных и аппаратных приложений. Поэтому для программирования эффективных виртуальных компьютеров практически полезной будет теория, основанная на двух, более высокого уровня абстракции, компонентах: память и транзакция.

Особенность организации данных в классическом компьютере заключается в том, что каждый бит, байт или другой компонент имеет свой адрес. Поэтому существует проблема в эффективной обработке ассоциации (конечного алфавита символов) равных по значимости элементов, которые не имеют порядка по определению, например, множество всех подмножеств. Решением может быть процессор, где элементарной ячейкой служит образ или шаблон универсума из n унитарно кодированных примитивов, которые используют

суперпозицию для формирования булеана $Q = 2^n$ всех возможных состояний такой ячейки в виде множества всех подмножеств [5,6]. Прослеживается определенная аналогия по структурам данных с квантовым компьютером, где понятию кубита в классическом компьютере можно поставить во взаимно-однозначное соответствие четыре состояния: $\{10, 01, 11, 00\}$ [7] векторного задания булеана примитивов Кантора $A^k = \{0, 1, X, \emptyset\}$, где X –

описывает «перепутанное», одновременно суперпозиционирующее состояние двух уровней сигналов 0 и 1. Большинство работ, в частности [1-4], рассматривает возможность эмуляции классических вычислительных процессов на квантовых компьютерах, но учитывая «реверсность» или обратимость упомянутого соответствия, далее предлагается обратное преобразование – эмуляция некоторых преимуществ квантовых вычислений на классических процессорах.

На рынке электронных технологий существует конкуренция между базами имплементации идеи [8]: 1) Гибкая (мягкая) реализация проекта связана с синтезом интерпретативной модели программной реализации устройства или в аппаратном исполнении программируемых логических устройств на основе FPGA, CPLD. Здесь преимущества заключаются в технологичности модификации проекта, недостатки – в невысоком быстродействии функционирования цифровой системы. 2) Жесткая реализация имеет ориентацию на использование компилятивных моделей при разработке программных приложений или на имплементацию проекта в кристаллах VLSI. Преимущества и недостатки жесткой реализации инверсны по отношению к мягкому исполнению проектов: высокое быстродействие и невозможность модификации. С учетом четырех изложенных базовых вариантов для реализации идеи ниже предлагаются квантовые структуры данных, ориентированные на повышение быстродействия гибких моделей программного или аппаратного исполнения проекта.

2. Квантовые структуры описания цифровых систем

n -Кубит есть векторная форма 2^n унитарного кодирования универсума из n примитивов для задания булеана состояний 2^2 с помощью 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) [1,8]. 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 анализа дискретных систем. Далее кубит отождествляется с n -кубитом или вектором, если это не мешает пониманию излагаемого материала. Поскольку квантовые вычисления связаны с анализом кубитных структур данных, то далее частично эксплуатируется определение «квантовый» для идентификации технологий, использующих два свойства квантовой механики: параллелизм обработки и суперпозицию состояний. Синонимом кубита при задании логической функциональности является Q-покрытие (Q-вектор) [7], как унифицированная векторная форма суперпозиционного задания выходных состояний, соответствующих унитарным кодам адресов входных переменных любой логической функции.

Кубит в цифровой системе выступает в качестве формы задания структурного примитива, инвариантной к технологиям реализации функциональности (hardware, software). Более того, «квантовый» синтез цифровых систем на основе кубитных структур уже не привязан жестко к теореме Поста, определяющей условия существования функционально-полного базиса. На предлагаемом уровне абстракции n -кубит дает исчерпывающие и более широкие возможности для векторного задания любой функции из множества $\beta(f) = 2^n$. Формат структурного кубитного компонента цифровой схемы $Q^* = (X, Q, Y)$ включает интерфейс (входные и выходную переменные), а также кубит-вектор Q , задающий функциональность $Y = Q(X)$, размерность которого определяется степенной функцией от числа входных линий $k = 2^n$.

Практически ориентированная новизна кубитного моделирования заключается в замене таблиц истинности компонентов цифрового устройства векторами состояний выходов. Достаточно просто можно продемонстрировать такие преобразования применительно к логическому элементу. Пусть функциональный примитив имеет следующее двоичное покрытие:

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

которое можно трансформировать путем унитарного кодирования входных векторов на основе использования двухтактного алфавита [8,9]. Изначально он был предназначен для компактного описания всех возможных переходов автоматных переменных, что иллюстрируется на рис. 1 соответствующим графом и интерпретацией символов.

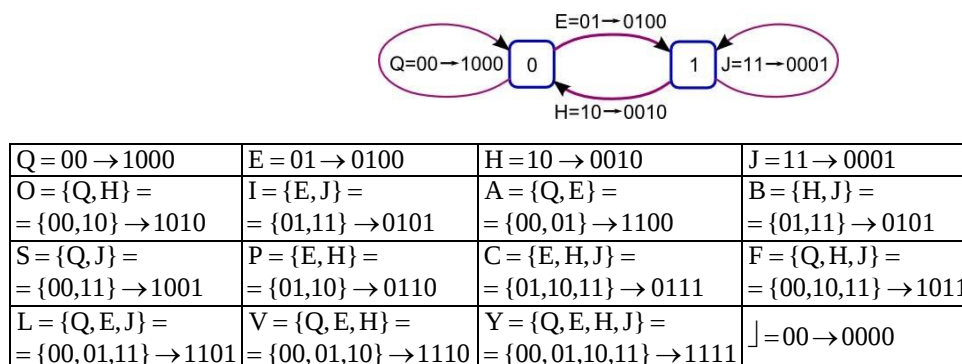


Рис. 1. Двухтактный алфавит автоматных переменных

Здесь представлены: символы, их двоичные и унитарные коды (например, Q = 00 – 1000), предназначенные для описания двух соседних состояний автоматных переменных. Структурно алфавит представляет собой булеан (множество всех подмножеств) состояний на универсуме из четырех примитивов $Y = \{Q, E, H, J\}$. Унитарный код соответствует формату вектора, содержащего два кубита, с помощью которых формируются 16 символов двухтактного алфавита. Используя последний, любое покрытие функционального двухвходового логического примитива можно представить двумя кубитами или даже одним, учитывая, что они взаимно инверсны:

$$P = \begin{bmatrix} 00 & 1 \\ 01 & 1 \\ 10 & 1 \\ 11 & 0 \end{bmatrix} = \begin{bmatrix} Q & 1 \\ E & 1 \\ H & 1 \\ J & 0 \end{bmatrix} = \begin{bmatrix} V & 1 \\ J & 0 \end{bmatrix} = \begin{bmatrix} 1110 & 1 \\ 0001 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1 & 1 & 0 \end{bmatrix}$$

Здесь сначала кодируются все пары символами двухтактного алфавита, затем выполняется объединение первых трех кубов по правилу оператора сограющей [5]: векторы отличающиеся в одной координате, минимизируются в один. Далее полученное покрытие из двух кубов кодируется соответствующими данным символам кубитными векторами. 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. Ввиду тривиальности нет смысла показывать, что по аналогии любую таблицу истинности можно привести к кубитной функциональности в форме вектора выходных состояний логического элемента, имеющего n входов.

Процедура моделирования на Q -векторе функциональности сводится к записи в выходную переменную Y состояния бита, адрес которого сформирован на основе конкатенации значений входных переменных: $Y = Q(X) = Q(X_1 * X_2 \dots * X_j \dots * X_k)$. Для моделирования цифровых систем, где компонентами выступают взаимосвязанные на основе M -вектора эквипотенциальных линий Q -примитивы, процедура обработки последних определяется выражением: $M(Y) = Q[M(X)] = Q[M(X_1 * X_2 \dots * X_j \dots * X_k)]$. С учетом сквозной нумерации Q -примитивов универсальная процедура моделирования текущего i -элемента будет иметь формат: $M(Y) = Q[M(X)] = Q[M(X_1 * X_2 \dots * X_j \dots * X_k)]$. В данном случае существенно упрощается алгоритм анализа цифровой системы и в 2^n раз повышается быстродействие интерпретативного моделирования за счет увеличения объема памяти для описания функциональности схемной структуры.

Синтез Q-покрытия цифровой системы сводится к выполнению операции суперпозиции над Q-векторами функциональностей, входящих в нее. Например, для трех примитивов (элементы and, and-not, and-not), составляющих схему, операция суперпозиции формирует Q-вектор всей функциональности, где его размерность будет больше, чем сумма Q-покрытий исходных примитивов:

$$\begin{array}{|c|c|c|c|c|c|} \hline a & 0 & 0 & 0 & 1 & c \\ \hline b & & & & & \\ \hline d & 1 & 1 & 1 & 0 & f \\ \hline e & & & & & \\ \hline \end{array} \begin{array}{|c|c|c|c|c|} \hline c & 1 & 1 & 1 & 0 & g \\ \hline f & & & & & \\ \hline \end{array} = \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|} \hline a & & & & & & & & & & & & & & & \\ \hline b & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & g \\ \hline d & & & & & & & & & & & & & & & & \\ \hline e & & & & & & & & & & & & & & & & \\ \hline \end{array}.$$

Но при этом процедура моделирования Q-вектора структуры будет иметь более высокое быстродействие, поскольку она представлена только одним обращением к Q-покрытию для выемки содержимого из ячейки вместо трех, когда система представлена тремя примитивами.

Трехэлементная схема, представленная выше Q-векторами, может быть задана схемотехнически (удобно для человека), где вместо векторов будут фигурировать соответствующие десятичные номера:

$$\begin{array}{|c|c|c|} \hline a & 1 & c \\ \hline b & & \\ \hline d & 14 & f \\ \hline e & & \\ \hline \end{array} \begin{array}{|c|c|c|} \hline c & 14 & g \\ \hline f & & \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline a & & \\ \hline b & 34679 & g \\ \hline d & & \\ \hline e & & \\ \hline \end{array}.$$

При обработке такой формы функциональных покрытий необходимо «развернуть» десятичный код в двоичный вектор и вычислить адрес ячейки, содержимое которой будет определять состояние выходной переменной, в данном случае g. Естественно, что десятичный код существует на бумаге, а в компьютере это представление – всегда двоичный вектор. На самом деле «мягкая» схемотехника идентификации (нумерации) межсоединений имеет будущее, поскольку не связана с соединительными проводами, которые заменяются адресами или номерами линий, создающими структуру цифрового изделия, которая обладает гибкостью замены примитивов в случае обнаружения ошибок проектирования или дефектов.

Кубитное представление функциональных элементов дает также возможность ввести новые схемотехнические обозначения, связанные с десятичным номером Q-вектора, задающего функциональность. Если система логических элементов имеет $n=2$ входа, то число всех возможных функций равно $k = 2^{2^n}$, где типы или номера функционалов представлены в нижней строке следующей таблицы:

00	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
01	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
10	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
11	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
f =	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Более того, на основе множества кубитов первого уровня, задающих функции от двух переменных, можно ввести кубит второго уровня, унитарно кодирующий двухвходовые функции, что дает возможность создавать структуру одновременного задания и анализа всех неупорядоченных состояний дискретной системы, где входными переменными выступают уже функционалы первого уровня:

00	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
01	0	0	0	0	1	1	1	0	0	0	0	1	1	1	1
10	0	0	1	1	0	0	1	1	0	1	0	1	0	0	1
11	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
Q =	0	1	1	1	1	0	0	0	0	0	0	1	1	0	0

В данной таблице представлены четыре вектора-примитива входных переменных (00,01,10,11), образующие $k = 2^{2^2} = 2^4$ полное множество всех возможных функций, кото-

рые рассматриваются в качестве примитивов второго уровня. Затем – векторы-примитивы выходных переменных (16 столбцов от 0000 до 1111) формируют уже $k = 2^2 = 2^{16}$ функциональных примитивов, входящих в состав более сложной дискретной системы, которые можно анализировать параллельно! Далее можно экстраполировать создание более сложной системы кубитов, где вектор $Q = 0111110000001100$, представленный нижней строкой, будет рассматриваться как один из $k = 2^2$ примитивов третьего уровня иерархии. В каждом уровне иерархии кубитов количество или булеан состояний экспоненциально зависит от числа примитивов-векторов $k = 2^2$. Если вектор Q имеет все единичные значения $Q = 1111111111111111$, то он одновременно определяет пространство, содержащее 16 символов двухтактного алфавита, которые соответствуют булеану на универсуме из четырех примитивов [9].

Основная инновационная идея квантовых вычислений по сравнению с машиной фон Неймана 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. В этом тезисе сформулировано будущее всех высокопроизводительных компьютеров для параллельного нецифрового анализа структур и сервисов дискретного киберпространства. Иначе, вычислительная сложность выполнения процедуры обработки множества из n элементов в «квантовом» процессоре и одного в машине фон Неймана равны между собой из-за соответствующего n -кратного повышения аппаратной сложности «квантовой» структуры.

3. Графовые структуры описания цифровых схем

Несколько отличная модельная схемотехника, не привязанная непосредственно к транзисторам, может быть представлена графовыми структурами, где каждая вершина (дуга) отождествляется с функциональным преобразованием, которое задается Q -вектором. Тогда дуга (вершина) определяет взаимосвязи между функциональными Q -покрытиями, а также входные и выходные переменные. Вопрос реализации таких структур изначально привязан к ячейкам памяти (LUT FPGA), которые способны хранить информацию в виде Q -вектора, где каждый бит или разряд имеет свой адрес, отождествляемый со входным словом. Тем не менее, программная реализация таких структур становится конкурентоспособной по быстродействию на рынке промышленных систем проектирования цифровых систем на кристаллах за счет адресной реализации процессов моделирования функциональных примитивов. Кроме того, аппаратная поддержка систем проектирования в виде Hardware Embedded Simulator (HES, Aldec) [10] приобретает новую мотивацию на системном уровне проектирования цифровых изделий, когда программные и аппаратные решения имеют один и тот же кубитный формат. Далее для рассмотрения предлагается комбинационная схема (рис. 2), содержащая 6 примитивов и три различных логических элемента.

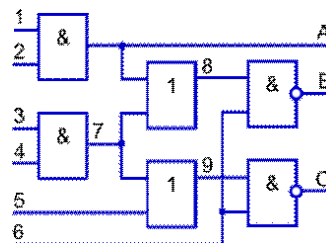
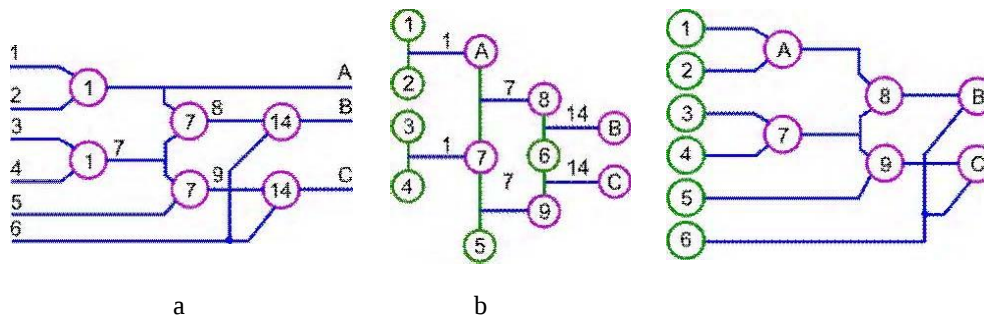


Рис. 2. Комбинационная структура логических примитивов

Данной схеме соответствуют три универсальные графовые формы цифровой функциональности (рис. 3), использующие Q -векторы для задания поведения логических примитивов.



с Рис. 3. Графовые формы квантовых функциональностей

Структура, представленная на рис. 3,а, содержит 12 линий (дуг), нагруженных на квантовые функциональности (1 = 0001, 7=0111, 14=1110). Она подобна традиционной структурно-функциональной модели комбинационной схемы. Граф на рис. 3,б напоминает модель регистровых передач С.Г. Шаршунова [11], который является обратным по отношению к первой структуре. Здесь горизонтальные синие дуги отождествляются с функциональностями, а вершины – с группами входных для функциональностей линий, объединенных в регистровые переменные посредством зеленых вертикальных дуг, состояния которых образуют двоичный вектор, используемый в качестве адреса для вычисления состояния логического элемента или более сложного функционала. Переменные, участвующие в формировании адреса для Q-вектора функциональности, можно объединить в одну вершину с указанием всех идентификаторов линий, создающих вектор-адрес. Регистровый граф комбинационной схемы является ранжированным по уровням формирования входных сигналов, что обеспечивает условия параллелизма обработки элементов одного уровня и выполнимости итераций Зейделя [5,9], повышающих быстродействие алгоритмов исправного моделирования цифровых систем. Структура на рис. 3,б интересна своей регистровой реализацией, что можно использовать для формализации как программных, так и аппаратных моделей вентильного, регистрового и системного уровней. Такое представление трудно воспринимается человеком, но оно технологично и легко понимается компьютером для автоматического создания программных систем анализа вычислительных структур и сервисов киберпространства. Таким образом, регистровый граф цифровой схемы представляет собой разрывную по гальваническим связям, а потому гибкую систему взаимосвязанных адресных примитивов для формирования функциональной структуры любой сложности, прежде всего в масштабах PLD, где все комбинационные примитивы представлены постоянными запоминающими устройствами (LUT), что обеспечивает высокое быстродействие функционирования и ремонт логических модулей в режиме online.

Одномерный Q-вектор описания функциональности можно привязать к выходной (внутренней) линии устройства, состояние которой формируется в процессе моделирования рассматриваемого Q-покрытия. Тогда регистровая реализация комбинационного устройства может быть представлена вектором моделирования M, на невходные линии которого нагружена функциональность с дугами, идущими от входных переменных, значения которых задают адрес бита Q-вектора, формирующего состояние рассматриваемой невходной линии (рис. 3,с). Иначе, если функциональности описываются одновыходовыми примитивами, то каждый из них можно отождествить или идентифицировать номером или координатой невходной линии, на которую нагружен данный элемент. Если функциональность многovýchодовая, то Q-покрытие уже представляется матрицей с количеством строк, равным числу выходов. Эффект от такого примитива заключается в параллелизме одновременного вычисления состояний нескольких выходов за одно обращение к матрице по текущему адресу! Данное обстоятельство является существенным аргументом в пользу синтеза обобщенных кубитов для фрагментов цифрового устройства или всей схемы в целях их параллельной обработки в одном временном такте. Близкой к идеальной по компактности и времени обработки структурой данных, где Q-векторы функциональностей и номера входных переменных привязаны к невходным линиям устройства, является следующая таблица:

L	1	2	3	4	5	6	7	8	9	A	B	C
M	1	1	1	1	1	0	0	1	1	1	0	1
X	.	.	.	.	.	.	34	A7	75	12	86	96
Q	.	.	.	.	.	.	0	0	0	0	1	1
	.	.	.	.	.	.	0	0	1	1	1	1
	.	.	.	.	.	.	0	0	1	1	1	1
	.	.	.	.	.	.	1	1	1	1	0	0

Она дает представление о том, какие переменные цифровой схемы являются внешними, сколько функциональных примитивов имеется в структуре, а также какие входы нагружены на каждый Q-вектор. Достоинством таблицы является отсутствие вектора номеров выходов для каждого примитива, но при этом сохраняется необходимость иметь номера входных переменных для формирования адресов, манипулирование которыми есть достаточно времязатратный процесс. Модель анализа схемной структуры упрощается до вычисления двух адресов (!) при формировании вектора моделирования: $M_i BQ_i[M(X_i)]$ путем исключения сложного адреса выхода примитива в процессе записи состояний выходов в координаты M-вектора.

Кубитно-регистровый граф из рис. 3,с может быть представлен в виде матрицы $\Delta \Delta_{ij}$, $i \in \Delta 1, p$; $j \in \Delta 1, q$ параллельно-последовательной обработки логических примитивов:

μ_{ij}	1	2	3
1	$\begin{array}{ c } \hline 1 \\ \hline 2 \end{array} \begin{array}{ c } \hline 1 \\ \hline A \end{array}$	$\begin{array}{ c } \hline A \\ \hline 7 \end{array} \begin{array}{ c } \hline 7 \\ \hline 8 \end{array}$	$\begin{array}{ c } \hline 8 \\ \hline 6 \end{array} \begin{array}{ c } \hline 14 \\ \hline B \end{array}$
2	$\begin{array}{ c } \hline 3 \\ \hline 4 \end{array} \begin{array}{ c } \hline 1 \\ \hline 7 \end{array}$	$\begin{array}{ c } \hline 7 \\ \hline 5 \end{array} \begin{array}{ c } \hline 7 \\ \hline 9 \end{array}$	$\begin{array}{ c } \hline 6 \\ \hline 9 \end{array} \begin{array}{ c } \hline 14 \\ \hline C \end{array}$
3	$\begin{array}{ c } \hline X \\ \hline X \end{array} \begin{array}{ c } \hline 1 \\ \hline X \end{array}$	$\begin{array}{ c } \hline X \\ \hline X \end{array} \begin{array}{ c } \hline 7 \\ \hline X \end{array}$	$\begin{array}{ c } \hline X \\ \hline X \end{array} \begin{array}{ c } \hline 14 \\ \hline X \end{array}$

который отображает взаимодействие Q-покрытий по трем уровням срабатывания относительно формата (X-Q-Y) входы- Q-вектор -выход каждого примитива: [(1,2-1-A), (3,4-1-7)], [(A,7-7-8), (7,5-7-9)], [(8,6-14-B), (6,9-14-C)]. Чтобы корректно отработала функциональность примитива, необходимо к рассматриваемому моменту сформировать все его входные переменные. Поэтому кубитно-регистровый граф разделен на уровни срабатывания, где все примитивы внутри одного уровня могут обрабатываться параллельно, а сами уровни – последовательно друг за другом. Кубитная матрица своей регулярной структурой ориентирована на решение задач: 1) Ремонт логических примитивов в процессе функционирования за счет переадресации дефектных элементов на примитивы из запаса (строка 3) [12], подобно тому, как это делается в матричной памяти. 2) Индексная адресация каждого кванта матрицы $\mu_{ij} \in \mu$, $\mu_{ij} = (X_{ij}, Q_{ij}, Y_{ij})$ для оперативного ремонта отказавших примитивов (в примере можно заменить три дефектных примитива, по одному из каждого слоя). 3) Обеспечение высокого быстродействия прототипа комбинационного устройства, реализованного на основе кубитных примитивов, которые имплементируются на кристалле PLD в LUT-элементы [8], за счет параллельной обработки элементов одного слоя. 4) Создание матричного кубитного мультипроцессора, ориентированного на анализ аппаратных прототипов комбинационных устройств большой размерности, позволяющих существенно ускорить процессы тестирования и верификации цифровых систем на кристаллах, как это делает HES компании Aldec [10]. 5) Разработка методов анализа комбинационных схем, ориентированных на матричное исполнение кубитных структур логических элементов путем их имплементации в элементы памяти кристаллов PLD. 6) Создание генератора кода для масштабируемого синтеза квантовых матриц комбинационных схем на основе использования структур схемотехнических примитивов кристаллов PLD. 7) Проектирование управляющего автомата для функциональной обработки и сервисного обслуживания (восстановления работоспособности) кубитной матрицы комбинационного устройства, имплементированного в PLD структуру.

Модель управляющего автомата для симулирования кубитной структуры комбинационной схемы укладывается в три пункта:

1) Инициирование очередного входного воздействия для комбинационного устройства.
 2) Выбор очередного слоя (столбца матрицы) с номером i для параллельной обработки кубитных примитивов Q в целях формирования состояний выходов по адресу входного слова, представленного вектором $M(X_{ij})$, где X_{ij} – вектор номеров входных переменных для примитива Q_{ij} , M – вектор моделирования всех линий комбинационного устройства: $M(Y_{ij}) = Q_{ij}[M(X_{ij})]$, $j = 1, q$.

3) Приращение индекса столбца $i = i + 1$ и переход к пункту 2 обработки очередного слоя кубитных примитивов. По окончании анализа всех столбцов матрицы $i = V_r$ выполняется инкремент индекса очередного входного воздействия $t = t + 1$ с последующим переходом на пункт 1. При достижении конечного числа входных наборов $t = V_{n_{\max}}$ цикл обработки теста для кубитной матрицы заканчивается.

4. Автоматная структура MQT-процессора

Основой процесс-модели функционирования вычислителя (комбинационной схемы), представленного в форме кубитной матрицы, является операция транзакции (считывания-записи) двоичной информации на LUT-элементах памяти структуры PLD: $M(Y) = Q[M(X)]$, которая формирует сколь угодно сложные функциональности и сервисы. Иначе, все вычислительные процессы в компьютерных системах, сетях и в киберпространстве можно свести к одной операции транзакции на любой структуре, способной хранить информацию. Все технологические и схемотехнические узлы, на которых реализуются аппаратные и программные продукты, можно не принимать во внимание, чтобы синтезировать виртуальные информационные сервисы, где следует использовать только операцию запись-считывание, как базовую процедуру над Q -покрытиями, которые своей гибкой универсальностью поглощают все конструктивы вычислительной техники для синтеза и анализа объектов, процессов и явлений в киберпространстве. Появляется новая MQT-модель «мягкого» кубитного виртуального компьютера, основанная на транзакции (Transaction) между компонентами памяти Q (Qubit), объединенными в систему с помощью вектора моделирования M (Memory). Естественно, что любые логические функции реализуются путем считывания битов Q -вектора $Q = (q_1, q_2, \dots, q_i, \dots, q_n)$, $q_i \in \{0, 1\}$. Но при этом становится несколько избыточной теорема Поста, формирующая условия в виде математического функционального базиса логических функций, необходимо и достаточно полного для создания любой вычислительной системы, поскольку Q -вектор есть универсальная, гибкая и компактная форма описания как простой, так и сложной функциональности киберпространства. Транзакция может существовать только при наличии заданных отношений на компонентах памяти, число которых должно быть не менее одного. Цикличность транзакции

$M \xrightarrow{Y_i} Q_i \xleftarrow{X_i} M$ (рис. 4, a, b), представленной Read-Write операциями $M(Y_i) = Q_i[M(X_i)]$ в MQT-структуре, определяется использованием вектора M взаимных связей компонентов, который образует «мягкую», адресно ориентированную и гальванически разорванную автоматную модель, где главным компонентом вычислительной системы фигурирует кубит-память:

$$\begin{aligned} \spadesuit A &= \langle M, Q, f, g, X, Y \rangle, \\ \spadesuit M(t+1) &= f[X(t), Q(t), M(t)]; \\ \spadesuit Y(t) &= g[X(t), Q(t), M(t)]; \\ \spadesuit M(Y_i) &= Q_i[M(X_i)]. \end{aligned}$$

Данные выражения подходят под классическое определение автомата Мура, но здесь равенство $M(Y_i) = Q_i[M(X_i)]$ задает основную и единственную процедуру анализа адресуемых Q -компонентов цифровой системы для формирования координат вектора состояния M . Последний представляет собой адресно вычисляемые реакции Q -примитивов уст-

ройства на входные воздействия, полученные конкатенацией координат вектора М, адреса которых соответствуют номерам входных переменных рассматриваемого примитива. Автомат имеет входную регистровую переменную X, вектор выходных сигналов Y, который, являясь подмножеством переменных вектора М, может быть исключен из рассмотрения.

Адресные регистровые переменные Y_i, X_i каждого примитива дают возможность формировать состояние вектора М путем моделирования структуры квантовых примитивов Q.

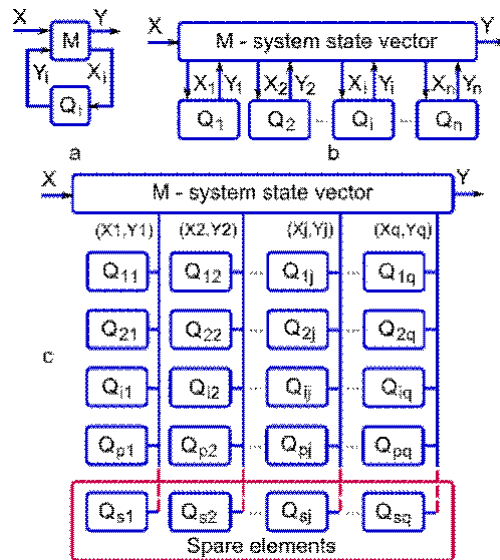


Рис. 4. Битовая, векторная и матричная структуры кубитных примитивов

Здесь (рис. 4) модель цифрового изделия в виде структурного MQT-автомата включает вектор моделирования или состояния системы, который объединяет и «мягко» структурирует все кубитные примитивы для формирования функциональности, заданной спецификацией. При этом цикл обработки каждого Q-элемента заключается в транзакционном взаимодействии между собой пары компонентов М–Q (см. рис. 4,а), которые в совокупности реализуют универсальную функциональность $M(Y_i) = Q_i[M(X_i)]$ с помощью двух транзакций $M \leftarrow Y_i$ и $Q_i \leftarrow X_i$.

Обработка всех кубитных элементов на входном воздействии X дает возможность сформировать вектор М состояния цифровой системы (System State Vector – SSV), который при необходимости имеет и выходные переменные Y (см. рис. 4,б) для управления другими компонентами вычислительной структуры. Быстродействие анализа (запись-считывание) векторной (линейной) структуры из $n = p \times q$ примитивов имеет оценку: $\gamma = (R + W) \times n$. Строго последовательный характер обработки упорядоченных по возрастанию номеров выходов квантовых примитивов можно усовершенствовать в сторону уменьшения времени анализа схемы. Для этого необходимо построить уже двумерную структуру – матрицу Q-элементов, ранжированных по уровням параллельной обработки групп примитивов, оформленных в столбцы (см. рис. 4,с). Быстродействие такой структуры, по сравнению с линейной повышается в q раз, что становится уже соизмеримым со временем обработки комбинационной схемы на основе жестко заданных гальванических

связей $\gamma = \frac{1}{q} (R + W) \times p$. Но главное преимущество матрицы «мягких» связей компонен-

тов – наличие нижней строки запасных примитивов для ремонта в режиме online, что делает любой проект, содержащий комбинационную логику, тестопригодным, благодаря оперативной переадресации отказавшего Q- примитива на элемент из ремонтного запаса. Для этого желательно при синтезе такой двумерной структуры располагать в одном столбце однотипные примитивы, что уменьшит издержки на количество запасных элементов.

Инновационная идея матрицы кубитных примитивов характеризуется реализацией комбинационных Q-покрытий на адресуемых координатах кубита и элементах памяти, мягко соединенных в цифровую схему с помощью вектора состояний линий, что дает возможность ремонтировать отказавшие логические примитивы в реальном времени с помощью их переадресации на запасные компоненты при достаточно высоком быстродействии функционирования вычислительного устройства. Платой за такие преимущества является существенная аппаратная избыточность по сравнению с жесткой комбинационной схемой, которая заключается в добавлении следующих компонентов (рис. 5): CU – устройство управления; Q-ADC – дешифратор адресов примитивов; M-SSV – вектор состояния цифровой системы; X-ADC – дешифратор адресов входов; X-M – память входов; Y-ADC – дешифратор адресов выходов; Y-M – память выходов.

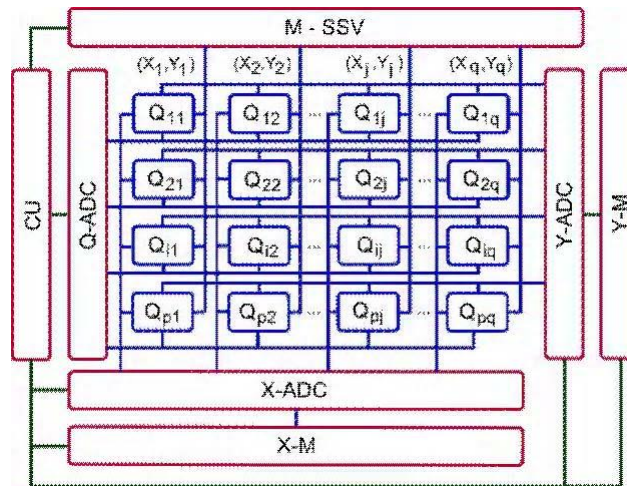


Рис. 5. Матричная структура операционного устройства

Упрощение автомата Мура до пары транзакционно взаимодействующих компонентов на основе M-вектора моделирования, адресно структурирующего элементы, привело к тривиальной модели MQT-автомата:

$$\begin{cases} A = \langle M, Q, X, Y \rangle, \\ M(Y_i) = Q_i[M(X_i)]. \end{cases}$$

Привязка идентификаторов функциональных Q-примитивов к номеру (адресу) выходной линии цифровой системы дает возможность упростить MQT-автомат управления еще на одну транзакционную операцию, связанную с формированием адреса состояния выхода примитива Y:

$$\begin{cases} A = \langle M, Q, X \rangle, \\ M_i = Q_i[M(X_i)]. \end{cases}$$

Здесь уже нет системы выходных переменных, которые заменяются координатами M-вектора моделирования линий цифровой системы. Представленный MQT-автомат имеет особенности: 1) Отсутствие жестких гальванических связей между элементами, формирующими комбинационную логику. 2) Наличие «мягких» или гибких связей Q-примитивов, формируемых с помощью M-вектора (моделирования) состояния цифровой системы (System State Vector – SSV). 3) Частичное уменьшение быстродействия вычислительной структуры за счет устранения гальванических связей. 4) Обеспечение новой возможности комбинационной структуры, которая заключается в восстановлении работоспособности системы в случае возникновения отказов в примитивных элементах. 5) Инвариантность реализации MQT-компьютеров по отношению к среде использования (виртуальное пространство, критические объекты) и субстанции имплементации (твердые кристаллы и материалы, жидкие, газообразные и плазменные формы существования материи). 6) Структура адресного взаимодействия Q-компонентов памяти между собой посредством M-вектора моделирования (сосредоточение входов и выходов цифровой системы) напоминает технологию

использования всех устройств компании Apple, когда входом и выходом любого гаджета является киберпространство (интернет или облака данной компании). Иначе, можно сделать вывод – как на макро-, так и на микроуровне среды применения автоматов уже нет необходимости использовать внешние входы и выходы цифровой системы, которая уже сегодня ориентирована на жесткое взаимодействие с киберпространством и не представляет особого интереса как автономное вычислительное изделие. Вместо входов и выходов существует единое адресное пространство, в данном случае вектор M , откуда вычислительная система берет входные данные, а после преобразований она записывает туда же результаты или выходные данные:

$$\begin{aligned} \uparrow A &= \langle M, Q \rangle, \\ \downarrow M_i &= Q_i(M_j). \end{aligned}$$

Поэтому MQT-модель системного описания структур данных подобна отношениям между вычислительными устройствами (Q-компоненты) и интернетом (M-память). По-видимому, это есть настоящее и будущее технологической культуры, когда компьютер (гаджет) связывается с киберпространством, которое является входом и выходом для любого вычислительного устройства. 7) Что касается обработки кубитных вычислительных MQT-структур более низкого уровня, то здесь работает простая и эффективная транзакционная процедура – кубит функциональности берет данные из адресного пространства M , а затем туда же записывает результат преобразования: $M_i = Q_i(M_j)$. 8) Иначе, предлагаемая MQT-модель вычислительной ячейки киберпространства включает структурный MQ-автомат адресной мягкой организации взаимодействия функциональных примитивов $MQ = \{M, Q, [(M \times Q) \rightarrow M] \times [M_x = Q(M_x)]\}$ посредством M-вектора моделирования и MT-автомат управления обработкой кубитных примитивов на основе использования единственной операции транзакции, регламентируемой характеристическим выражением $M_i = Q_i(M_j)$, определяющим взаимосвязи адресуемых компонентов и процедуру их транзакционной обработки. Таким образом, предложенный MQT-автомат, используя только элементы памяти и единственную операцию транзакции, дает возможность системно проектировать высоконадежные вычислительные и информационные сервисы как в реальном, так и в виртуальном мире.

В качестве примера далее предлагается анализ цифровой схемы на одном тестовом наборе, для которой вектор моделирования и кубитная структура представлены ниже:

$$M = \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|c|} \hline 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & A & B & C \\ \hline 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ \hline \end{array}$$

Q_{ij}	1					2					3				
1	1	0001 A				A	0111 8				8	1110 B			
	2					7					6				
2	3	0001 7				7	0111 9				6	1110 C			
	4					5					9				

Здесь сущность анализа цифровой структуры заключается в заполнении вектора моделирования по всем координатам: 1) Запись в вектор моделирования M по шестикординатам входных переменных двоичного набора 000111. 2) Обработка примитива с номером Q_{11} , который имеет входные переменные 1 и 2, а выход, обозначенный символом A. Для этого формируется адрес 00 путем конкатенации содержимого ячеек 1 и 2 вектора M . Применяя данный адрес к Q-вектору 0111, можно определить содержимое нулевой ячейки, равное нулю, которое записывается в вектор M по адресу A выходной переменной обрабатываемого кубита. 3) Повторение процедуры анализа, описанной в пункте 2, ко всем Q-примитивам дает возможность полностью определить двоичными сигналами координаты вектора моделирования: 000111001010.

Аппаратная имплементация приведенного и масштабируемого примера кубитной структуры цифрового устройства на основе использования элементов памяти представлена на рис. 6.

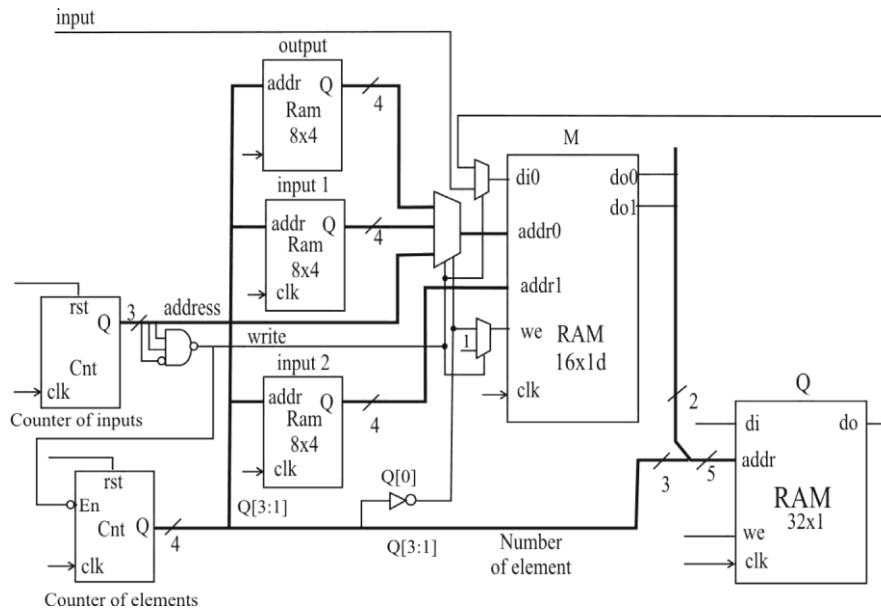


Рис. 6. Аппаратная реализация кубитной структуры комбинационной схемы

Структура схемной реализации содержит переменные и функциональные элементы, имеющие следующее назначение: input – вход для последовательного занесения входных значений вектора M; rst – общий сброс системы, в данном случае счетчиков; clk – вход синхронизации; counter of inputs – счетчик заполнения входных координат вектора M; counter of element – счетчик номера обрабатываемого примитива, который предоставляет два такта для считывания входного набора из двух координат вектора M; Q[3:1] – шина номера обрабатываемого примитива; Q[0] – переменная режима считывания входных значений из вектора M или записи результата в M. Блоки памяти: Ram 8x4 output – хранит номера выходных линий примитивов; Ram 8x4 input 1 и Ram 8x4 input 2 – сохраняют номера входных линий примитивов; Ram 16x1d – двухпортовая память хранения вектора моделирования M, где addr0 – адрес входа 1 при значении 00 на входах управления мультиплексора, адрес записи результата при значении 01 на входах управления мультиплексора, адрес для инициализации входных данных при значении 1X на входах управления мультиплексора; addr1 – адрес входа 2 обрабатываемого примитива; di0 – вход данных памяти при обработке примитива (MUX=1) или внешний вход input при инициализации входных данных (MUX=0); we – разрешение записи в вектор M; do0 – выход, соответствующий входу addr0; do1 – выход, соответствующий входу addr1. Компонент RAM 32x1 предназначен для хранения Q-векторов задания функциональностей комбинационной схемы: di – вход данных, может быть использован для инициализации (записи) структуры кубитов; addr – [4:0]: addr[4:2] – номер элемента, addr[1:0] – входной набор для примитива. Сложность аппаратной реализации примера комбинационной схемы составляет 150 вентилей, включающих 20 LUT системы элементов компании Xilinx Spartan 3E. Быстродействие функционирования или формирования вектора моделирования равно 180 ns.

5. MQT-структуры для будущих исследований

Введение MQT-автомата для синтеза и анализа вычислительных сервисов позволяет сделать и более общие выводы. Мир имеет дуализм материально-информационной целостности, а значит детерминированной воспроизводимости объектов, процессов и явлений. Любой материальный объект способен хранить информацию, следовательно, его можно использовать для записи и считывания, а значит – для создания вычислительной системы, пусть даже в некоторой степени специфической. Человек есть биологический компьютер с заданной программой функционирования (жизнедеятельности) всех его органов во времени. Как любую программу, код (геном) функционирования человека можно считать-записать, а значит – можно и скорректировать. Любые объекты на планете имеют свои программы существования во времени – информационные геномы. Процессы и явления на

планете также имеют свои программы развития, но более высокого уровня иерархии в управлении. Вселенная также имеет собственный геном – информационную программу развития. Естественно, что любую программу можно считать-записать, а значит скорректировать – это оптимистическая нота информационной модели объектов, процессов и явлений материального мира! Декодировать программу, как и дизассемблировать двоичный код, сложно, но возможно. Для этого нужно определить место – носитель кода и способ шифрования каждого объекта, процесса или явления природы. Можно предположить, что геном биологического объекта имеет форму, подобную структуре многозначных кубит-векторов, когда каждый из них $Q = (q_1, q_2, \dots, q_i, \dots, q_n), q_i \in \{a_1, a_2, \dots, a_j, \dots, a_k\}$ может быть трансформирован в двоичную матрицу, где символ алфавита описания разрядов кванта может быть развернут в двоичный вектор или код-столбец. Многозначность функциональности достаточно распространена в природе, удобна для восприятия глазом человека, но для компьютера символы любого алфавита пока трансформируются в двоичные векторы или коды. Геном человека с позиции компьютерной инженерии есть структура данных, записанная в биопамять, компактная программа развития и существования во времени и пространстве белковых конструктивов, которая имеет аппарат управления-мониторинга (чтения-записи) без существования жестких гальванических связей. Такие принципы биоинженерии следует применять и к объектам киберпространства, чтобы обеспечить их надежность, ремонтпригодность, восстанавливаемость на микро- и макроуровнях.

Таким образом, кубитные структуры дают возможность сделать из комбинационной схемы простейший автомат (интегрирующая память, кванты функциональностей, операция транзакции) и перейти от программного моделирования цифровых систем к аппаратной эмуляции структур и процессов, составляющей уже пользовательские функции компьютера, инвариантные по отношению к технологиям имплементации. Показательным аналогом и прообразом служит аппаратный ускоритель процессов моделирования PRUS [7] доктора Stanley Hyduke, Aldec Inc., ориентированный на уменьшение времени проектирования и верификации цифровых систем на кристаллах. Но в сравнении с [7] здесь предлагается использовать по прямому функциональному назначению процессор «мягкого» адресно ориентированного аппаратного моделирования кубитных структур в качестве вычислительного изделия, доставляющего сервисы потребителю. При этом сохраняется высокое быстродействие функционирования устройства, дополненное существенной для критических систем возможностью восстановления работоспособности в режиме реального времени.

Технологическая сингулярность (Вернор Виндж) – взрыв в понимании законов вселенной на коротком промежутке времени и создание кибермозга человечества в целях повышения качества жизни очевидно произойдет благодаря развитию трех компонентов: биоинженерии, искусственного интеллекта киберпространства и нанотехнологий проектирования. Здесь имеется в виду: 1) встроенная и непосредственная интеграция человека с компьютером (киберпространством) путем устранения языковых барьеров и интерфейсов между ними; 2) создание искусственного интеллекта для самообучения, во-первых, и самосовершенствования, во-вторых, когда компьютеры и киберпространство будут способны к самовоспроизводству более совершенных моделей, структур и процессов; 3) «нано-выращивание» компьютера путем структурирования атомов. Одним из главных условий достижения данной точки развития технологической культуры является создание принципиально нового компьютера (вычислительного процесса), инвариантного по отношению к любой точке пространства или его субстанции, которую можно структурировать и использовать вместо кремния. Пространственный параллелизм существующих компьютеров и точечная одновременная многозначность – два пути в развитии интеллектуального киберпространства. Иначе, компьютер можно сделать на чем угодно, что способно сохранять во времени и/или в пространстве не менее двух управляемых и мониторируемых состояний. Регистр или память, как основа для хранения данных и выполнения вычислительных процедур, представляет собой несколько физических точек или триггеров, к которым можно применить параллельную операцию, разнесенную в пространстве. При этом повышение производительности цифрового устройства сегодня осуществляется только за счет расширения субстанции вычислительной структуры. Вопрос – можно ли объединить упомянутые точки

пространства в одну для выполнения параллельных операций над совокупностью данных, сосредоточенных в одной материальной точке? Возможно, так и ставился практическими учеными вопрос, который предсказал появление квантового компьютера. Несомненно, что будущее компьютерной индустрии должно быть связано с многовариантностью физической точки пространства как примитива структур данных, которая должна иметь возможность хранения булеана состояний в качестве операнда для реализации вычислительных процедур. Но здесь возникает только одно решение, заслуживающее внимания. Все, что физически является стабильным во времени, может иметь только одно состояние в каждый конкретный момент времени, включая цифровые дискретные автоматы, устройства и системы. Решение совершенно очевидно – дуализм частицы и волны на всем спектре электромагнитного излучения обладает искомыми свойствами пространственной точки: 1) Каждая точка пространства может иметь в конкретный момент времени множество волн или квантов из известных и неизвестных человечеству диапазонов. 2) Это означает возможность наличия в точке пространства множества частиц соответствующих диапазонов, которые формируют совокупность информационных состояний данной точки. 3) Неопределенность (многовариантность) во времени процессов и явлений есть ключ к созданию принципиально новых компьютеров, которые можно называть как квантовыми, так и электромагнитными, учитывая дуализм электродинамики в отношении носителя (частица или волна). 4) На каждой стадии развития технологической культуры необходимо выбирать подходящую точку пространства или субстанции (газообразная, жидкая, твердая, плазменная), которую можно использовать для мониторинга и управления ее многозначных состояний с помощью существующих технологий. Предсказания фантаста Станислава Лема и ученого В.И. Вернадского о возможности существования разума в жидкой (океан Солярис) и газообразной среде (ноосфера) сегодня уже близки во времени к созданию глобального искусственного разума киберпространства, который мы должны получить к 2050 году. 5) Дискретизация спектра частот в каждой точке пространства может служить основой для кодирования как примитивов (двоичных, многозначных), так и более сложных ассоциаций, например, булеана состояний. Иначе, 8-разрядный регистр, имеющий 256 состояний, можно представить одной точкой в пространстве для последующей обработки на основе мониторинга и управления соответствующими квантами или спектром частот. 6) Если предположить, что кванты существуют во всех диапазонах непрерывного и бесконечного спектра, то вопрос заключается в том, какой диапазон частот в точке пространства или субстанции будет освоен первым для создания новых как мощных, так и простых квантовых электродинамических вычислителей на основе экономически приемлемых технологий мониторинга и управления (атомными примитивами) частицами или волнами. 7) Рыночная привлекательность «зеленых» нанотехнологий («восходящее» проектирование) – построение или выращивание вычислителя (квантового компьютера) путем структурирования атомов – заключается в безотходности, микроминиатюрности, сверхнизком энергопотреблении, абсолютно минимальной затратности материалов, а в будущем и стоимости, сверхвысоком быстродействии и требуемой (по принципу разумной достаточности) масштабируемости, соизмеримой с классом доставляемых сервисов. Микротехнологии («нисходящее» проектирование), суть которых: взять заготовку и отсечь все лишнее, становятся все менее привлекательными для рынка ввиду несовершенства качества и сверхзатратности создаваемых изделий.

Ученые научились сегодня не только сканировать атомные структуры, но и последовательно строить или выращивать их. Однако на пути решения проблемы нанотехнологического направления разработки квантового компьютера на рынке имеется три привлекательных пункта: 1) Открытие технологий с высоким быстродействием выращивания требуемых гетерогенных атомных структур в соответствии с заданной программной спецификацией вычислителя. 2) Создание эффективного транзакционного механизма для реализации простого мониторинга и управления квантовыми состояниями выращенных атомных вычислительных структур с адресуемыми компонентами. 3) Обеспечение требуемой стабильности во времени состояний компонентов атомной структуры, реализующей память. Для решения проблемы необходимо объединить достижения и усилия ученых, работающих в областях: квантовой физики атомных взаимодействий, биоинженерии, микроэлектроники, электрохимии и компьютерной инженерии.

6. Заключение

1. Предложены кубитные модели описания цифровых систем и компонентов, которые характеризуются компактностью описания таблиц истинности в форме Q-покрытий благодаря унитарному кодированию входных состояний, что дает возможность повысить быстродействие программных и аппаратных средств интерпретативного моделирования вычислительных устройств за счет адресной реализации анализа логических примитивов.

2. Представлена матричная модель кубитных примитивов для реализации комбинационных схем, которая характеризуется адресным объединением Q-покрытий на элементах памяти, «мягко» соединенных в цифровую схему с помощью вектора состояний линий, что дает возможность ремонтировать отказавшие логические примитивы в реальном времени с помощью их переадресации на запасные компоненты при достаточно высоком быстродействии функционирования вычислительного устройства.

3. Введена гибкая автоматная MQT-модель компьютера, которая характеризуется использованием только адресуемых структур памяти и операции транзакции для программной и аппаратной реализации комбинационных и последовательностных функциональностей, что дает возможность создавать быстродействующие и надежные вычислители для создания сервисов киберпространства на основе параллельных логических операций и ремонта неисправных адресуемых функциональных примитивов.

4. Описана инновационная идея квантовых вычислений, которая характеризуется переходом от вычислительных процедур над байт-операндом, определяющим в дискретном пространстве одно решение (точку), к логическим регистровым параллельным процессам над кубит-операндом, одновременно формирующим булеан решений, что дает возможность определить новые перспективы на пути создания высокопроизводительных компьютеров параллельного анализа и синтеза структур и сервисов дискретного киберпространства.

5. Представлена концепция системного проектирования программных и аппаратных облачных сервисов киберпространства на основе MQT-автомата, которая характеризуется применением транзакций на кубитных адресно связанных компонентах вычислителя с помощью вектора состояний (моделирования), что дает возможность программировать логические функциональности в кубитных примитивах интерпретативной структуры цифрового устройства.

6. Рыночная привлекательность MQT-компьютера определяется: примитивизмом его реализации в программном и аппаратном исполнении; высоким уровнем использования памяти в структуре современных вычислителей, достигающим 94% от площади кристалла; и как следствие – уменьшением влияния комбинационной логики на быстродействие системы в целом; повышением надежности компьютеров за счет онлайн-ремонта адресуемых элементов памяти, включая логические примитивы. Существующее неприятие MQT-компьютера рынком связано с лоббированием вычислителей на основе жесткой комбинационной логики в VLSI-проектах со стороны компаний, разрабатывающих процессоры на основе технологий *reusable logic*.

7. Направления будущих исследований: Синтез кубитных моделей цифровых систем на основе использования Q-покрытий, а также разложение (анализ) функциональностей на кубитные компоненты. Моделирование неисправностей и синтез тестов на основе применения Q-покрытий компонентов цифровых систем. Разработка специализированных программных приложений для решения задач синтеза комбинационных устройств на основе использования матричной модели MQT-процессора, ориентированного на высокий параллелизм решения практических задач и восстановление работоспособности устройства в реальном времени.

Список литературы: 1. *Michael A. Nielsen & Isaac L. Chuang*. Quantum Computation and Quantum Information. Cambridge University Press. 2010. 676p. 2. *Stig Stenholm, Kalle-Antti Suominen*. Quantum approach to informatics. John Wiley & Sons, Inc., 2005. 249p. 3. *Mark G. Whitney*. Practical Fault Tolerance for Quantum Circuits. PhD dissertation. University of California, Berkeley. 2009. 229p. 4. *Mikio Nishihara*. Quantum Computing. An Overview. Higashi-Osaka. Kinki University, 2010. 53p. 5. *Горбатов В.А.* Основы дискретной математики. М.: Высш. шк. 1986. 311 с. 6. *Hahanov V.I., Litvinova E.I., Chumachenko S.V., Baghdadadi 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. **7.** *Хаханов В.И., Мурад Али Аббас, Литвинова Е.И., Гузь О.А., Хаханова И.В.* Квантовые модели вычислительных процессов // Радиоэлектроника и информатика. 2011. №3. С.35-40. **8.** Проектирование и тестирование цифровых систем на кристаллах / В.И. Хаханов, Е.И. Литвинова, О.А. Гузь. Харьков: ХНУРЭ. 2009. 484 с. **9.** *Хаханов В.И.* Техническая диагностика цифровых и микропроцессорных структур. К.: ИСИО. 1995. 242 с. **10.** *Бондаренко М.Ф., Хаханов В.И., Литвинова Е.И.* Структура логического ассоциативного мультипроцессора // Автоматика и телемеханика. 2012. № 10. С. 71-92. **11.** *Борисовец Б. Э., Шаршунов С. Г.* Общая модель и синтез тестов для механизмов управления межрегистровым обменом данными в микропроцессорах // Автоматика и телемеханика, 1992. № 8. С. 142–149. **12.** *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. 18.

Поступила в редколлегию 22.09.2013

Хаханов Владимир Иванович, декан факультета КИУ ХНУРЭ, д-р техн. наук, профессор кафедры АПВТ ХНУРЭ. IEEE Senior Member. IEEE Computer Society Golden Core Member. 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. +380 57 70-21-326. 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.

Литвинова Евгения Ивановна, IEEE Member, д-р техн. наук, профессор кафедры АПВТ ХНУРЭ. Научные интересы: проектирование и тестирование цифровых систем и сетей на кристаллах. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. +380 57 70-21-421. E-mail: kiu@kture.kharkov.ua.

Хаханова Ирина Витальевна, д-р техн. наук, профессор кафедры АПВТ ХНУРЭ. Научные интересы: проектирование и тестирование цифровых систем и сетей на кристаллах. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. +380 57 70-21-326. E-mail: hahanova@mail.ru.

Врублевский Николай Николаевич, студент факультета КИУ ХНУРЭ. Научные интересы: проектирование и тестирование цифровых систем и сетей на кристаллах. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. +380 57 70-21-326.

АННОТАЦИИ

УДК 681.326:519.713

В.И. ХАХАНОВ, BAGHDADI AMMAR Awni Abbas, Е.И. ЛИТВИНОВА, И.В. ХАХАНОВА, Н.Н. ВРУБЛЕВСКИЙ

КУБИТНЫЕ СТРУКТУРЫ ДАННЫХ ВЫЧИСЛИТЕЛЬНЫХ УСТРОЙСТВ

Предлагаются кубитные модели и методы повышения быстродействия программных и аппаратных средств анализа цифровых устройств за счет увеличения размерности структур данных и памяти. Вводятся основные понятия, термины и определения, необходимые для имплементации квантовых вычислений в практику анализа виртуальных компьютеров. Описываются результаты исследований, касающихся проектирования и моделирования компьютерных систем в киберпространстве на основе использования двухкомпонентного автомата <память, транзакции>.

ABSTRACTS

UDC 681.326:519.713

Qubit data structures of computer systems / V. I. Hahanov, Baghdadi Ammar Awni Abbas, E.I. Litvinova, I.V. Hahanova, M.M. Vrublevskiy // Management Information System and Devices. 2013. N 164. P. 4-19.

Qubit models and methods for improving the performance of software and hardware for analyzing digital devices through increasing the dimension of the data structures and memory are proposed. The basic concepts, terminology and definitions necessary for the implementation of quantum computing when analyzing virtual computers are introduced. The investigation results of computer system design and modeling in a cyberspace based on the use of two-component structure <memory - transactions> are presented.

Fig. 6. Ref.: 12 items.

РЕФЕРАТИ

УДК 681.326:519.713

Кубітні структури даних обчислювальних систем / В.І. Хаханов, Baghdadadi Ammar Awni Abbas, Є.І. Литвинова, І.В. Хаханова, М.М. Врублевський // АСУ та прилади автоматики. 2013. Вип. 164. С. 4-19.

Запропоновані кубітні моделі і методи підвищення швидкодії програмних і апаратних засобів аналізу цифрових пристроїв за рахунок збільшення розмірності структур даних і пам'яті. Введені основні поняття, терміни та визначення, необхідні для імплементції квантових обчислень в практику аналізу віртуальних комп'ютерів. Представлені результати досліджень, що стосуються проектування і моделювання комп'ютерних систем в кіберпросторі на основі використання двокомпонентного автомата < пам'ять, транзакції >.

Іл. 6. Бібліогр. : 12 назв.