КОМПЬЮТЕРНАЯ ИНЖЕНЕРИЯ И ТЕХНИЧЕСКАЯ ДИАГНОСТИКА

УДК004.45

# КУБИТНАЯ ФОРМА ОПИСАНИЯ ВЫЧИСЛИТЕЛЬНЫХ СТРУКТУР

*TAMER BANI AMER, ЧУМАЧЕНКО С.В., ЕМЕЛЬЯНОВ И.В.*

Разрабатываются кубитные модели описания структур и функциональностей для повышения быстродействия анализа цифровых устройств за счет увеличения размерности структур данных и памяти. Вводятся основные понятия, термины и определения, необходимые для реализации квантовых вычислений в практику моделирования компьютерных структур. Описываются примеры, подтверждающие эффективность использования кубитных структур данных для параллельного выполнения операций над данными.

**Ключевые с лова:** квантовые векторы, графы и функциональности, кубитные структуры, параллельные вычисления, моделирование цифровых систем.

## Введение

Рыночная привлекательность эмуляции квантовых методов вычислений при создании компьютинговых структур (КС) в киберпространстве основана на использовании кубитных моделей данных, ориентированных на параллельное решение задач проектирования, тестирования, дискретной оптимизации при существенном повышении затрат памяти. Не вдаваясь в детали физических основ квантовой механики, касающиеся недетерминированного взаимодействия атомных частиц [1,3], далее используется понятие кубита как двоичного или многозначного вектора для совместного и одновременного задания булеана состояний в дискретной области киберпространства на основе линейной суперпозиции унитарных кодов, ориентированных на параллельное использование методов анализа и синтеза компонентов киберпространства. В быстро развивающейся теории квантовых вычислений векторы состояний образуют квантовый регистр из n кубитов, формирующих унитарное или гильбертово [4, 5] пространство H, размерность которого имеет степенную зависимость от числа кубитов Dim H = 2n.

Мотивация нового подхода для проектирования КС обусловлена появлением облачных сервисов в рамках новой киберкультуры Internet of Things, которые представляют собой специализированные и рассредоточенные в пространстве виртуальные компьютерные системы [6-9], реализуемые в аппаратуре или в программном продукте. Любой компонент функциональности, равно как и структура системы, представляется векторной формой, упорядоченной по адресам, таблицей истинности, реализуемой с помощью памяти. Логические функции в традиционном исполнении не рассматриваются. От этого частично уменьшается быстродействие, но, учитывая, что 94% SoC-кристалла составляет память [8], оставшиеся 6% следует реализовывать на памяти, что не будет критичным для большинства облачных сервисов. Практически для создания эффективных компьютинговых структур следует использовать теорию, основанную на вычислительных компонентах высокого уровня абстракции: адресуемая память и транзакция.

Особенность организации данных в классическом компьютере состоит в адресации бита, байта или другого компонента. Адресуемость создает проблему обработки ассоциации неадресуемых элементов множества, которые не имеют порядка по определению. Решением может быть процессор, где образ универсума из n унитарно кодированных примитивов использует суперпозицию для формирования булеана $|B(A)|=2^n$ всех возможных состояний [8].

Существует аналогия векторного представления булеана с кубитом квантового компьютера. Квантовому кубиту хранения информации в квантовом компьютере [1], разрешающей суперпозицию $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, можно поставить во взаимно-однозначное соответствие булеан состояний, формирующий алфавит Кантора $A^k = \{0,1, X, \varnothing\}, X = \{0,1\}$. Здесь примитивы алфавита унитарно кодируются векторами: 0 = (10) и 1 = (01). Коды других символов являются производными по операции суперпозиции $(10) \vee (01) = (11)$ и пересечения $(10) \wedge (01) = (00)$, что формирует булеан: {(10),(01),(11),(00)} [6].

Корректность использования прилагательного «кубитная» для моделей цифровых устройств основана на сравнении линейной и булевой алгебры Кантора с алфавитом $A^k = \{0,1, X, \varnothing\}$. Здесь первые два символа – примитивы. Третий определяется суперпозицией: $X = 0 \cup 1$. В гильбертовом пространстве линейная алгебра оперирует примитивами $\alpha|0\rangle$ и $\beta|1\rangle$ кубита, третий символ также есть суперпозиция двух составляющих: $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. В линейной алгебре нет символа, соответствующего пустому множеству, он есть производная от функции, обратной по отношению к суперпозиции. В теории множеств такой операцией является пересечение, которое дает пустое множество на примитивах: $\varnothing = 0 \cap 1$. В гильбертовом пространстве такой операцией является скалярное произведение или функция Дирака $\langle a|b\rangle$ [4], которая имеет геометрическую интерпретацию:

$$< a \mid b > \approx \mid a \mid \times \mid b \mid \times \cos \angle(a, b).$$

Если проекции a и b вектора квантового состояния ортогональны, получается:

$$<\alpha|\beta>=|a|\times|b|\times\cos\angle(a,b)=|a|\times|b|\times\cos\angle 90^{\circ}=0.$$

Скалярное произведение ортогональных векторов равно нулю, что является аналогом символа пустого множества в алгебре Кантора. Таблица соответствия алгебры множеств и линейной алгебры, представленная ниже, подтверждает свойства изоморфизма между символами булеана и состояниями кубит-вектора:

| $\text{Boolean}\,A^k=$ | 0 | 1 | $X=0\cup 1$ | $\varnothing=0\cap 1$ |
|---|---|---|---|---|
| $\text{Qubit}\ \|\psi\rangle=$ | $\|0\rangle$ | $\|1\rangle$ | $\alpha\|0\rangle+\beta\|1\rangle$ | $\alpha\|0\rangle\|\beta\|1\rangle$ |

Следовательно, структуру данных «булеан» можно рассматривать как детерминированный образ квантового кубита в алгебре логики, элементы которой унитарно кодируются двоичными векторами и обладают свойствами суперпозиции, параллелизма и перепутывания. Это дает возможность использовать предлагаемые модели для повышения быстродействия анализа цифровых устройств на классических вычислителях, а также без модификации – в квантовых компьютерах, которые появятся через несколько лет на рынке электроники.

**Квантовое описание цифровых функциональных элементов**

Кубит (n-кубит) есть векторная форма унитарного кодирования универсума из n примитивов для задания булеана состояний $2^{2^n}$ с помощью $2^n$ двоичных переменных. Если n=2, то 2-кубит задает 16 состояний с помощью четырех переменных, при n=1, кубит задает четыре состояния на универсуме из двух примитивов (10) и (01) с помощью двух двоичных переменных (00,01,10,11) [7]. При этом допускается суперпозиция в векторе $2^n$ состояний, обозначенных примитивами. Кубит дает возможность использовать параллельные векторные логические операции вместо поэлементных теоретико-множественных для существенного ускорения процессов анализа дискретных систем. Далее кубит отождествляется с n-кубитом или двоичным вектором, если это не мешает пониманию излагаемого материала. Синонимом кубита при задании двоичного вектора логической функции является Q-покрытие (Q-вектор) [8] как унифицированная векторная форма суперпозиционного задания выходных состояний, соответствующих адресным кодам входных переменных функционального элемента. Формат структурного кубитного компонента цифровой схемы $Q^*=(X,Q,Y)$ включает интерфейс (входные и выходную переменные), а также кубит-вектор Q, задающий функцию $Y=Q(X)$, размерность которого определяется степенной функцией от числа входных линий $k=2^n$. Новизна кубитной формы заключается в замене неупорядоченных по строкам таблиц истинности функциональных элементов векторами упорядоченных состояний выходов. Если функциональный примитив имеет двоичное покрытие, то ему можно поставить в соответствие кубит или Q-покрытие: Q=(1110):

$$P=\begin{array}{|c|c|c|}\hline X_1 & X_2 & Y\\\hline 0&0&1\\0&1&1\\1&0&1\\1&1&0\\\hline\end{array}\rightarrow Q=(1110).$$

Преимущества n-кубита заключается в способности параллельно выполнять логические операции над векторным форматом теоретико-множественных данных. Например, хог-операция над A={a,b,c,d,e,f} и B={a,c,f,g,h,k} выполняется параллельно за один такт, если каждый элемент будет представлен унитарным кодом, а подмножества –векторами, которые являются кубит-операндами:

| 9 – qubit | a | b | c | d | e | f | g | h | k |
|---|---|---|---|---|---|---|---|---|---|
| A = | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| B = | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| A⊕B = | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |

**Квантовое описание графовых структур**

Квантовый Q-вектор или кубит, естественно, можно и следует использовать для представления графовых структур. Функция и структура – две взаимно обратимые формы описания некоторой сущности. Рассмотренный выше кубит-вектор или квант идеально укладывается в технологию программируемых логических устройств, которые используют адресуемые элементы памяти. Учитывая определенный изоморфизм функций и структур, несложно «запихнуть» в память второй компонент описания дискретной сущности. Для этого необходимо представить некоторый ориентированный граф в виде вершин V, кодированных двоичными векторами и дугами A, создающими направленные переходы между

$$G=(V,A),\ V=(V_1,V_2,..,V_i,..,V_n),\ A=(A_1,A_2,..,A_j,..,A_{n^2}).$$

В этом случае ориентированный граф будет представлен таблицей адресов (кодов) всех вершин истоков и стоков, представляющих декартово произведение. Каждой паре вершин $(V_i,V_j),\ i,j=\overline{1,n}$ ставится в соответствие 1, если существует дуга между ними $(V_i\rightarrow V_j)$, и 0, если такого соединения нет. Таким образом, квадратичная таблица или матрица смежностей для задания графа превращается в вектор, размерностью n2, который технологически тривиально размещается в адресуемом элементе памяти. Такой вектор описания структуры ничем не отличается от задания любой дискретной функциональности и к нему применимы все формальные методы анализа, моделирования, тестирования, верификации и синтеза

дискретных компонентов и систем. Следовательно, создается единый универсальный формат данных в виде кубита или квантового вектора для представления структуры вычислительного устройства и функциональных описаний всех его компонентов. Далее приводятся примеры кубитного задания тривиальных графовых структур, представленных на рис. 1.

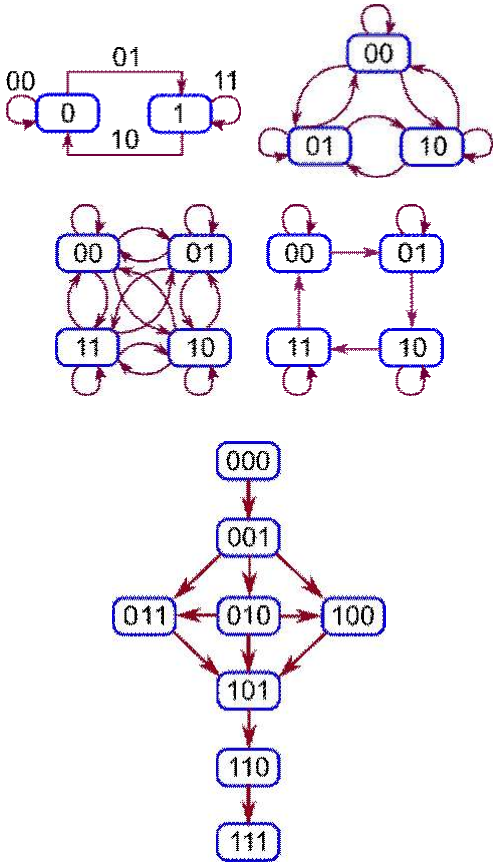



Рис. 1. Примеры примитивных графов для их кубитного описания (G1-G5)

Граф G1 представлен двумя вершинами, которые закодированы 0 и 1 соответственно. Между вершинами имеются четыре перехода, которые обозначены векторами: 00, 01, 10, 11. Данные пары двоичных символов составляют также адреса ячеек памяти, в которые записываются символы 1, поскольку каждый переход имеет место в графе:

$$\begin{array}{|c|c|}\hline 0&0\\\hline 0&1\\\hline 1&0\\\hline 1&0\\\hline\end{array}\rightarrow\begin{array}{|c|}\hline 1\\\hline 1\\\hline 1\\\hline 1\\\hline\end{array}\rightarrow\boxed{1\,1\,1\,1}=Q(G_1).$$

Отсутствие некоторого перехода в графе идентифицируется символом 0 в соответствующей ячейке памяти, адрес которой составлен кодами вершин (истока и стока). Граф G2 представлен тремя вершинами, что означает наличие двух двоичных разрядов для кодирования его вершин. Таблица истинности данного графа (повернута для горизонтального инкремента адресов в целях экономии места) также имеет нулевые координаты, поскольку все переходы, связанные с теоретически возможной вершиной 11, отсутствуют:

$$\begin{array}{|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|}\hline 0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1\\\hline 0&0&0&0&1&1&1&1&0&0&0&0&1&1&1&1\\\hline 0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1\\\hline 0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1\\\hline 1&1&1&0&1&1&1&0&1&1&1&0&0&0&0&0\\\hline\end{array}\rightarrow\boxed{1\,1\,1\,0\,1\,1\,1\,0\,1\,1\,1\,0\,0\,0\,0\,0}=Q(G_2).$$

Если использовать все четыре вершины и построить на них полный граф переходов (G3), то все ячейки кубита, содержащего 16 координат, будут равными 1:

$$\begin{array}{|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|}\hline 0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1\\\hline 0&0&0&0&1&1&1&1&0&0&0&0&1&1&1&1\\\hline 0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1\\\hline 0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1\\\hline 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\\hline\end{array}\rightarrow\boxed{1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1}=Q(G_3).$$

Граф G4 также представлен четырьмя вершинами, но он уже не имеет всех возможных переходов, что означает присутствие нулевых координат в кубитном векторе:

$$\begin{array}{|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|}\hline 0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1\\\hline 0&0&0&0&1&1&1&1&0&0&0&0&1&1&1&1\\\hline 0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1\\\hline 0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1\\\hline 1&1&0&0&0&1&1&0&0&0&1&1&1&0&0&1\\\hline\end{array}\rightarrow\boxed{1\,1\,0\,0\,0\,1\,1\,0\,0\,0\,1\,1\,1\,0\,0\,1}=Q(G_4).$$

Таким образом, кубитная форма описания графа будет более эффективной, если число вершин в структуре будет стремиться к степени двойки. Кубит следует рассматривать как компактную и неявную форму представления графа, которая однозначно и технологически просто может быть превращена в графический рисунок на основе анализа содержимого адресных ячеек памяти (двоичного вектора).

Вершины графа G5 кодируются уже тремя двоичными разрядами, поскольку число вершин n=8. Адрес каждой ячейки кубитного вектора содержит 6 разрядов – удвоенное число разрядов для кодирования вершины, а длина кубит-вектора представлена выражением $q = 2^{2\times\log_2 n}$. В данном случае число разрядов кубита равно q=64, номера координат, в которых находятся единицы: 1, 10, 11, 12, 19, 20, 21, 29, 36, 46, 55, а кубит-вектор имеет следующий вид:

$$Q(G_5)=(0100000001110000001110000000100000010000000001000000001000000000).$$

Таким образом, длина кубит-вектора зависит только от количества вершин в графе. Если число дуг мало по сравнению с полным графом, который содержит n2 направленных соединений, то эффективность E использования кубит вектора – отношение количества единичных координат кубита card(1) к его общей длине: $E = \mathrm{card}(1)\big/2^{2\times\log_2 n}$ будет невысокой. Например, для графа G5: $E = 11\big/2^{2\times\log_2 8} = 0{,}172$.

**Анализ кубитного описания графовых структур**

Физическая сущность кубита – память (двоичный вектор), в которой каждая адресуемая ячейка хранит значение нуля или единицы. Логическая его сущность – кортеж, состоящий из трех компонентов: (A,B,C),

где A – первая часть адреса ячейки памяти, представленная двоичным кодом вершины истока, B – вторая часть адреса, соответствующая коду вершины стока, C – идентификатор истинности C(AB)=1 или ложности C(AB)=0 конкатенируемого высказывания: «существует дуга из вершины A в вершину B». Являясь одномерной формой матрицы смежностей кубит-вектор, ориентирован на решение всех задач дискретного моделирования, синтеза, анализа и оптимизации. Но вопрос заключается в нахождении того класса задач, для которого использование кубитного описания и адресно-ориентированных процедур, исключающих перебор, дает технологические или экономические преимущества.

**Синтез (modeling) кубитной модели системы на основе кубитных описаний компонентов**

Анализ или моделирование (simulation) кубитной модели структуры на основе заданных входных условий: поиск всех путей от входной или выходной вершины.

Поиск максимального или минимального пути в графе на основе моделирования начальных условий.

Покрытие минимальным количеством путей всех вершин графа для тестирования и верификации цифровых систем.

Поиск дефектов в вычислительных структурах на основе обратного прослеживания некорректной выходной реакции.

Процесс моделирования графа по его кубиту сводится к выполнению единственной процедуры – извлечение состояния из ячейки кубит-вектора по адресу, полученному на основе конкатенации адреса, составленного из кодов вершин истока и стока: $M(A,B) = Q(A*B)$. Такая процедура отвечает на вопрос – существует ли в графе дуга (AB). Целью процедуры моделирования может быть: 1) нахождение максимального или минимального пути в графе; 2) определение всех путей от входной вершины; 3) вычисление всех вершин преемников для заданного выхода. Если некоторая система содержит совокупность взаимосвязанных графов, тогда процесс моделирования будет использовать следующее выражение: $M_i(A,B) = Q_i(A*B)$.

Интересен также квантовый формат представления графа для эффективного параллельного выполнения элементарных алгебраических операций. Речь идет об алгебро-логических опрациях над кубит-векторами одной размерности: конъюнкция (пересечение графов), дизъюнкция (сложение графов), исключающее или (сравнение графов), инверсия (дополнение к графу). Здесь фактически в векторной форме осуществляются манипуляции с дугами на кубит-векторах одинаковой длины. Они могут быть получены, если все графы привести к одному универсальному множеству вершин, которое формирует одинаковое число потенциально возможных дуг. Примеры выполнения алгебро-логических операций над графами, заданными кубит-векторами, представлены в следующей таблице:

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| G1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| G2 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| And(G1,G2) | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| Or(G1,G2) | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| Xor(G1, G2) | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| Not G1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |

Операция xor показывает расстояние между графами, выраженное в количестве единиц, соответствующих кодовому расстоянию по Хэммингу. В данном примере расстояние между двумя графами равно $d(G1,G2) = 8$. Если два или более графа имеют различное число вершин, то для выполнения алгебро-логических операций над ними необходимо коды вершин всех графов привести к одинаковой размерности. Такая процедура создает одинаковое адресное пространство для кодов (A,B) ориентированных дуг всех графов. При этом размерность всех кубит-векторов станет одинаковой, что является необходимым и достаточным условием для алгебро-логических манипуляций над ними. Задача приведения или нормализации всех графовых структур к одинаковой размерности фактически означает перекодировку уже существующих вершин графа в новой системе координат. Если решение задачи нормализации связано с последующим сравнением графов, в целях определения степени их сходства (различия), то необходимо выполнять синтез кодов дуг, приближающий все графы к структуре максимального из них. Данная задача (нормализации, распознавания, кластеризации, классификации графов) чрезвычайно важна для теории, представляет самостоятельный практический интерес и здесь не рассматривается. Тем не менее, ниже представлены четыре квантовых вектора, которые описывают первые четыре графа, приведенные к формату четырех вершин и двухбитовых векторов (пробелы в таблице соответствуют нулевым координатам):

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| G1 | 1 | | | | 1 | | | | | | | | | | 1 | |
| G2 | 1 | 1 | 1 | | | 1 | 1 | 1 | | | 1 | 1 | 1 | | | |
| G3 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| G4 | 1 | 1 | | | | | 1 | 1 | | | | | 1 | 1 | 1 | |
| And(G3,G1) = G1 | 1 | | | | 1 | | | | | | | | | | 1 | |
| And(G3,G2) = G2 | 1 | 1 | 1 | | | 1 | 1 | 1 | | | 1 | 1 | 1 | | | |
| And(G3,G4) = G4 | 1 | 1 | | | | | 1 | 1 | | | | | 1 | 1 | 1 | |
| Xor(G2,G4) = 7 | | | | 1 | | 1 | | | | | 1 | 1 | | | 1 | 1 |

Операция логического умножения показывает, что в приведенном формате, состоящем из четырех вершин, все три структуры (G1,G2,G4) являются подмножеством графа G3. Кодовое расстояние по Хэммингу между вторым и четвертым графами (последняя строчка таблицы) равно 7 из 16.

Таким образом, представленные результаты имеют определенную научную и практическую значимость: 1) Предложена новая кубитная форма описания графов, которая характеризуется компактностью описа-

ния всех ориентированных дуг, векторным представлением структуры межсоединений, что дает возможность существенно повысить быстродействие методов анализа за счет выполнения параллельных логических операций. 2) Кубитная форма пригодна также для описания логических функциональностей, что делает ее универсальной структурой для применения в проектировании, синтезе и анализе современных специализированных компьютинговых систем. 3) Высокая технологичность применения квантовых или кубитных форм в описании вычислительных устройств дает основания к ее эффективному использованию при решении широкого спектра задач дискретной оптимизации, синтеза и анализа, распознавания и диагностирования, моделирования и тестирования.

**Моделирование кубитного описания цифровых структур**

Процедура функционального моделирования на Q-векторе логической функциональности сводится к записи в выходную булеву переменную Y состояния бита Q-вектора, адрес которого сформирован на основе конкатенации значений входных переменных: $Y = Q(X) = Q(X_1 * X_2 ... * X_j ... * X_k)$. Для моделирования цифровых схем вводится M-вектор состояний линий, как аналог Q-вектора, задающий значения внутренних переменных системы, который связывает Q-векторы логических примитивов (Q-примитивов) в структуру с помощью нумерации входных и выходных переменных каждого функционального элемента [6]. Процедура обработки последнего определяется выражением:

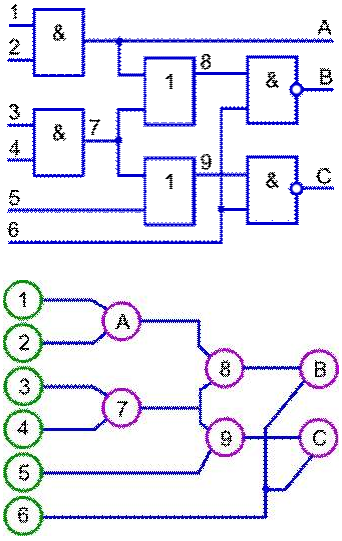
$$M(Y) = Q[M(X)] = Q[M(X_1 * X_2 ... * X_j ... * X_k)].$$

С учетом сквозной нумерации Q-примитивов универсальная процедура моделирования текущего i-элемента будет иметь формат

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

Существенно упрощается алгоритм моделирования цифровой системы, который сводится к процедуре формирования адреса, что дает возможность в $2^n$-1 раз повысить быстродействие интерпретативного моделирования за счет замены таблиц истинности примитивов на Q-векторы описания только выходных состояний. Кроме того, если логический элемент имеет n входов, то число строк таблицы истинности равно $2^n$, что означает – ее размерность равна $d = 2^n \times n$. Учитывая, что длина Q-вектора для любого логического примитива равна $2^n$, то выигрыш в объеме памяти для его хранения и обработки составляет: $r = \frac{2^n \times n}{2^n} = n$.

Комбинационная схема (рис. 2) содержит шесть примитивов и три различных логических элемента [6]. Данной схеме соответствует кубит графа и Q-векторы для задания логических примитивов.




$Q(G) = (.1.....11.....1.....1.....),$
$G = \{000-001,001-010,001-011,010-100,011-101\},$
$Q(A) = (0001), Q(7) = (0001), Q(8) = (0111),$
$Q(9) = (0111), Q(B) = (1110), Q(C) = (1110).$

Рис. 2. Граф цифрового устройства

Таким образом, структура графа Q(G1: A=000, 7-001, 8-010, 9-011, B-100, C-101), а также все функциональные элементы Q(A)-Q(8) представлены в форме кубитных покрытий или векторов, что существенно меньше по объему памяти, чем матрица смежностей и таблицы истинности логических элементов.

**Выводы**

1. Предложенные кубитные модели описания структуры цифровых систем и функций компонентов характеризуются компактностью описания матриц смежностей и таблиц истинности в форме Q-покрытий вследствие унитарного кодирования входных состояний, что позволяет повысить быстродействие программных и аппаратных средств для моделирования вычислительных устройств, благодаря адресной реализации анализа кубитных векторов.

2. Инновационная идея квантовых вычислений заключается в переходе от вычислительных процедур над байт-операндом, определяющим в дискретном пространстве одно решение, к логическим регистровым параллельным процессам над кубит-операндом, одновременно формирующим булеан решений. Это дает возможность определить новые пути создания высокопроизводительных компьютеров параллельного анализа и синтеза структур и компьютинговых сервисов.

3. Предложена новая кубитная форма описания графов, которая характеризуется компактностью описания всех ориентированных дуг, векторным представлением структуры межсоединений, что дает возмож-

ность существенно повысить быстродействие методов анализа и моделирования за счет выполнения параллельных логических операций.

4. Высокая технологичность применения квантовых или кубитных форм в описании вычислительных устройств дает основания к ее эффективному использованию при решении широкого спектра задач дискретной оптимизации, синтеза и анализа, распознавания и диагностирования, моделирования и тестирования.

5. Дальнейшие направления исследований связаны с разработкой быстродействующих методов и алгоритмов синтеза, анализа, визуализации, нормализации, распознавания, кластеризации, классификации графово-функциональных структур и систем на основе использования универсальной метрики (например, Манхэттэн) приведения и нормализации графов в единой системе координат.

**Литература: 1.** *Michael A. Nielsen & Isaac L.* Chuang. Quantum Computation and Quantum Information. Ï Cambridge University Press. 2010. 676 p. **2.** *Mark G. Whitney.* Practical Fault Tolerance for Quantum Circuits. PhD dissertation. University of California, Berkeley. 2009. 229 p. **3.** *Mikio Nfrfhara.* Quantum Computing. An Overview. Ï Higashi-Osaka: Kinki University, 2010. 53p. **4.** *Курош А.Г.* Курс высшей алгебры. М.: Наука. 1968. 426с. **5.** *Горбатов В.А.* Основы дискретной математики. М.: Высшая школа, 1986. 311 с. **6.** *Hahanov V.I., Litvinova E.I., Chumachenko S.V.* et al. Qubit Model for solving the coverage problem // Proc. of IEEE East-West Design and Test Symposium. Kharkov. 14-17 September, 2012. P.142-144. **7.** *Хаханов В.И., Ваджеб Гариби, Литвинова Е.И., Шкиль А.С.* Кубитные структуры данных вычислительных устройств // Электронное моделирование. 2015. № 1. С.76-99. **8.** *Хаханов В.И., Тамер Бани Амер, Чумаченко С.В., Литвинова Е.И.* Кубитные технологии анализа и диагностирования цифровых устройств //Электронное моделирование. 2015. Том 37. № 3 Р. 17-40. **9.** *Vladimir Hahanov, Wajeb Gharibi, Igor Iemelianov, Dmitry Shcherbin*. «Quantum» Processor for Digital Systems Analysis // Proceedings of IEEE East-West Design & Test Symposium (EWDTS-2015). 2015. Batumi, Georgia. P. 104-110.

Поступила в редколлегию 12.02.2016

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

**Tamer Bani Amer**, аспирант ХНУРЭ. Научные интересы: квантовые вычисления, тестирование и диагностика цифровых систем. Адрес: Україна, 61166, Харьков, пр. Науки, 14, тел. +3805770-21-326.

**Чумаченко Светлана Викторовна,** д-р техн. наук, профессор, зав. кафедрой АПВТ ХНУРЭ. Научные интересы: математическое моделирование, теория рядов, методы дискретной оптимизации. Адрес: Україна, 61166, Харьков, пр. Науки, 14, тел. + 3805770-21-326, e-mail: ri@kture.kharkov.ua.

**Емельянов Игорь Валерьевич**, научный сотрудник каф. АПВТ ХНУРЭ. Научные интересы: квантовые вычисления, тестирование и диагностика цифровых систем. Адрес: Україна, 61166, Харьков, пр. Науки, 14, тел. +3805770-21-326.