УДК 658.512.011:681.326:519.713

В.И. ХАХАНОВ[1], MURAD ALI ABBAS [1], Е.И. ЛИТВИНОВА[1], О.А. ГУЗЬ[2], И.В. ХАХАНОВА[1], BAGHDAD AMMAR AVNI ABBAS[1]

[1]Харьковский национальный университет радиоэлектроники, Украина
[2]Донецкая академия автомобильного транспорта, Украина

КВАНТОВЫЕ МОДЕЛИ ДАННЫХ И ВЫЧИСЛИТЕЛЬНЫХ ПРОЦЕССОВ

Предлагаются кубитные (квантовые) структуры данных и вычислительных процессов для существенного повышения быстродействия при решении задач поиска, распознавания, принятия решений, дискретной оптимизации и отказоустойчивого проектирования. Описаны аппаратно ориентированные модели параллельного (за один цикл) вычисления булеана (множества всех подмножеств) на универсуме из n примитивов для решения задач покрытия, минимизации булевых функций, сжатия данных, синтеза и анализа цифровых систем за счет реализации процессорной структуры в форме диаграмме Хассе. Представлен прототип квантового устройства, реализованного на основе программируемой логики для оптимального решения задачи покрытия при анализе киберпространства.

Ключевые слова: *кубит, квантовые вычисления, программируемая логика, киберпространство.*

Введение

Квантовые вычисления в последние годы становятся интересными для анализа кибернетического пространства, создания новых Интернет технологий, благодаря их некоторой альтернативности существующим моделям вычислительных процессов. Кроме того, рыночная привлекательность квантовых или кубитных моделей основывается на высоком параллелизме решения практически всех задач дискретной оптимизации, факторизации, минимизации булевых функций, эффективного сжатия, компактного представления и телепортации данных, отказоустойчивого проектирования [1 – 10] за счет существенного повышения аппаратных затрат. Но такая плата в настоящее время вполне допустима, поскольку существуют проблемы заполнения площадей силиконового кристалла, который содержит до 1 миллиарда вентилей при толщине пластины, равной 5 микрон. При этом современные технологии допускают создание пакета или «сэндвича», содержащего до 7 кристаллов, что соизмеримо с объемом нейронов головного мозга человека. Практически «беспроводное» соединение таких пластин основывается на технологической возможности сверления порядка 10 тысяч сквозных отверстий на 1 квадратном сантиметре. Наполнить полезной функциональностью такой объем допустимых на кристалле вентилей в настоящее время проблематично. Поэтому можно и нужно использовать «жадные» к аппаратуре модели и методы для создания быстродействующих средств параллельного решения практических задач. Имея ввиду дискретность и многозначность алфавитов описания информационных процессов, свойство параллелизма, заложенное в квантовых вычислениях, является особенно востребованным при создании эффективных и интеллектуальных «движков» для киберпространства или Интернета [11]; средств синтеза отказоустойчивых цифровых примитивов и систем [12]; тестирования и моделирования цифровых систем на кристаллах [13 – 15]; технологий защиты информации и компьютерных систем [5 – 7]. Здесь не рассматривается физическая основа квантовых вычислений, изначально заложенная в трудах ученых, ориентированных на возможность использования недетерминированных квантовых взаимодействий внутри атома [16, 17].

1. Кубитные, квантовые модели данных и вычислительных процессов

Квантовый компьютер предназначен для отказоустойчивого проектирования и решения оптимизационных задач, связанных с полным перебором на основе использования теории множеств. Особенность в том, что множество элементов в компьютере все равно упорядоченно, поскольку каждый байт имеет свой адрес. Поэтому теоретико-множественные операции сводятся к перебору всех адресов примитивных элементов. Адресный порядок структур данных хорош для задач, где компоненты моделей можно строго ранжировать, что дает возможность выполнять их анализ за один проход или одну итерацию. Там, где нет порядка в структуре, например, множество всех подмножеств, классическая модель памяти и вычислительных процессов наносит вред времени анализа ассоциации равных по рангу примитивов, или, в лучшем случае, обработка ассоциативных групп является неэффективной. Что можно предложить для неупорядоченных данных вместо строгого порядка? Процессор, где элементарной ячейкой служит образ или шаблон универсума из n примитивов, который генерирует $Q = 2^n$ всех возможных состояний такой ячейки в виде булеана или множества всех подмножеств. Прямое решение создания такой ячейки основано на унитарном позиционном кодировании состояний примитивов, которое с помощью суперпозиции последних образует множество всех подмножеств, формирующее в пределе универсум примитивов. Например, четыре примитива создают булеан, содержащий шестнадцать состояний (сочетаний), с помощью четырех двоичных разрядов:

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

Операции над символами теоретико-множественного алфавита сводятся к логическим командам and, or, not, xor, которые формируют функционально полный базис, согласно теореме Поста. Например, ниже представлены логические преобразования теоретико-множественных операций:

$$
\begin{aligned}
&Q \cup E = 1000 \vee 0100 = 1100 = A;\\
&S \cap V = 1001 \wedge 1110 = 1000 = Q;\\
&\tilde{B} = \overline{0011} = 1100 = A;\\
&F \Delta P = 1011 \oplus 0110 = 1101 = Y;\\
&H \Delta J = 0010 \oplus 0001 = 0011 = B;\\
&F \Delta Y = 1011 \oplus 1111 = 0100 = Y;\\
&F \Delta F = 1011 \oplus 1011 = 0000 = U(\varnothing);
\end{aligned}
$$

Другая интерпретация булеана из четырех примитивов (двоичные коды: 00, 01, 10, 11), представленная ниже:

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

создает 16 различных функций от двух переменных. В то же время последнюю таблицу можно представить как коды символов многозначного алфавита, которыми легко оперировать для решения задач синтеза и анализа булевых функций:

Q	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
E	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
H	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
J	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
	∅	J	H	B	E	I	P	C	Q	S	O	F	A	L	V	Y

Такую таблицу легко построить для любого числа примитивов (n=2, 3,4, 5, 6, 7, 8 …), где теоретико-множественные операции над символами сводятся к логическим операциям над векторами. Два наиболее традиционных примитива (0,1) создают известный алфавит Кантора:

0	0	0	1	1
1	0	1	0	1
	∅	1	0	X

Многозначность символов алфавита положительно влияет на минимизацию булевых функций. Например, компактное представление состояния входных переменных отдельных булевых функций от двух переменных при их кодировании символами 16-значного алфавита имеет не более двух кубов многозначного покрытия:

$$\begin{array}{|c|c|}\hline 00 & 0\\ 01 & 0\\ 10 & 0\\ 11 & 0\\ \hline\end{array} = \begin{array}{|c|c|}\hline Q & 0\\ E & 0\\ H & 0\\ J & 0\\ \hline\end{array} = \begin{array}{|c|c|}\hline Y & 0\\ \hline\end{array} = \begin{array}{|c|c|}\hline 1111 & 0\\ \hline\end{array};$$

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

$$\begin{array}{|c|c|}\hline 00 & 0\\ 01 & 0\\ 10 & 1\\ 11 & 0\\ \hline\end{array} = \begin{array}{|c|c|}\hline Q & 0\\ E & 0\\ H & 1\\ J & 0\\ \hline\end{array} = \begin{array}{|c|c|}\hline F & 0\\ H & 1\\ \hline\end{array} = \begin{array}{|c|c|}\hline 1101 & 0\\ 0010 & 1\\ \hline\end{array};$$

$$\begin{array}{|c|c|}\hline 00 & 0\\ 01 & 0\\ 10 & 1\\ 11 & 1\\ \hline\end{array} = \begin{array}{|c|c|}\hline Q & 0\\ E & 0\\ H & 1\\ J & 1\\ \hline\end{array} = \begin{array}{|c|c|}\hline A & 0\\ B & 1\\ \hline\end{array} = \begin{array}{|c|c|}\hline 1100 & 0\\ 0011 & 1\\ \hline\end{array};$$

$$\begin{array}{|c|c|}\hline 00 & 1\\ 01 & 1\\ 10 & 1\\ 11 & 1\\ \hline\end{array} = \begin{array}{|c|c|}\hline Q & 1\\ E & 1\\ H & 1\\ J & 1\\ \hline\end{array} = \begin{array}{|c|c|}\hline Y & 1\\ \hline\end{array} = \begin{array}{|c|c|}\hline 1111 & 1\\ \hline\end{array};$$

$$\begin{array}{|c|c|}\hline 00 & 0\\ 01 & 0\\ 10 & 1\\ 11 & 0\\ \hline\end{array} = \begin{array}{|c|c|}\hline Q & 0\\ E & 0\\ H & 1\\ J & 0\\ \hline\end{array} = \begin{array}{|c|c|}\hline F & 0\\ H & 1\\ \hline\end{array} = \begin{array}{|c|c|}\hline 1101 & 0\\ 0010 & 1\\ \hline\end{array};$$

$$\begin{array}{|c|c|}\hline 00 & 0\\ 01 & 1\\ 10 & 1\\ 11 & 1\\ \hline\end{array} = \begin{array}{|c|c|}\hline Q & 0\\ E & 1\\ H & 1\\ J & 1\\ \hline\end{array} = \begin{array}{|c|c|}\hline Q & 0\\ C & 1\\ \hline\end{array} = \begin{array}{|c|c|}\hline 1000 & 0\\ 0111 & 1\\ \hline\end{array};$$

$$\begin{array}{|c|c|}\hline 00 & 0\\ 01 & 1\\ 10 & 1\\ 11 & 0\\ \hline\end{array} = \begin{array}{|c|c|}\hline Q & 0\\ E & 1\\ H & 1\\ J & 0\\ \hline\end{array} = \begin{array}{|c|c|}\hline S & 0\\ P & 1\\ \hline\end{array} = \begin{array}{|c|c|}\hline 1001 & 0\\ 0110 & 1\\ \hline\end{array}.$$

Таким образом, переход от двоичных векторов входных сигналов к символам замкнутого многозначного алфавита дает принципиально новую возможность минимизации кубических покрытий (таблиц истинности), которые всегда будут иметь не более двух кубов

$$f(X) = \{C_1, C_0\},\ C_1 \cup C_0 = U,\ C_1 = \overline{C}_0,$$

формирующих единичное и нулевое значение выхода функции двумя взаимно дополняющими символами, в совокупности формирующими символ-универсум. При этом мощность универсума примитивов $\text{card}U = n$ формирует общее число состояний $Q = 2^n$ или производных от них символов, которые кодируются 2^n двоичными разрядами. Последующее двоичное кодирование входного символа каждого из двух кубов дает возможность максимально приблизиться к реализации функционального примитива как элемента памяти программируемых логических устройств (PLD), где входное слово логического элемента есть адрес ячейки памяти (бита), в котором записано состояние выхода. Однако таблица истинности в форме памяти есть ДНФ, которая необратима для решения задачи обратной импликации. Здесь выходом может служить явное задание функциональности в форме кубического покрытия, а точнее двух кубов покрытия, задающих все возможные решения по входам. При этом все логические элементы становятся одновходовыми, где входом является регистровая переменная или n-разрядный вектор, формирующий адрес памяти, хранящей $Q = 2^n$ бит, как значений функции

$$Y = f(A) = f(x_1, x_2, ..., x_i, ..., x_n).$$

Кубит есть двоичный вектор, содержащий n битов, для задания булеана (множества всех подмножеств) состояний $Q = 2^n$ на основе использования n примитивных символов (элементов).

Кубит – совокупность равнозначных двоичных n битов, формирующих единичным значением n примитивов, для обозначения $Q = 2^n$ состояний, составляющих булеан – множество всех подмножеств от n примитивов.

Здесь нет чисел! Все биты кубита равны при создании примитивов. Любая теоретико-множественная операция выполняется за один такт, что невозможно при задании ассоциации примитивов на счетном (упорядоченном) пространстве памяти компьютера. Метрика (векторная и скалярная) анализа расстояний, предложенная в [11, 13], абсолютно пригодная для измерения взаимодействия многозначных (двоичных) кубитных объектов, процессов и явлений, путем использования xor-операции.

В идеале использование кубитной структуры дает возможность любую функциональность представить в виде двух кубов, привязанных к нулю и единице. Такие кубы формируют КНФ и ДНФ соответственно. Можно упрощать и далее путем исключения из рассмотрения нуля и единицы, неявно имея их в виду. При этом два куба, формирующие входные условия, будут всегда взаимно инверсными, поскольку они дополняют друг друга до универсума примитивов. Следовательно, необходимо оставлять лишь одну букву (символ), а значит один двоичный код, который есть таблица истинности (двухвходового) функционального примитива:

$$\begin{array}{|c|c|}\hline 00 & 0 \\ 01 & 1 \\ 10 & 1 \\ 11 & 0 \\ \hline\end{array} = \begin{array}{|c|c|}\hline Q & 0 \\ E & 1 \\ H & 1 \\ J & 0 \\ \hline\end{array} = \begin{array}{|c|c|}\hline S & 0 \\ P & 1 \\ \hline\end{array} = \begin{array}{|c|c|}\hline 1001 & 0 \\ 0110 & 1 \\ \hline\end{array} \to P = \boxed{0110}$$

$$Y = P = E \vee H = A_1 \vee A_2 = \bar{x}_1 x_2 \vee x_1 \bar{x}_2.$$

Полученный вектор интерпретируется здесь не только как совокупность адресуемых битов, но как куб, формирующий единичное значение выхода примитива, над которым можно выполнять параллельные векторные логические операции. Формально существует два пути синтеза функциональности цифровой схемы, имеющей n входов. Первый формирует вектор длиной n, путем заполнения нулевыми и единичными значениями функции на основе прямого моделирования всех $q = m \times 2^n$ входных воздействий на m примитивах. Второй – метод обратного прослеживания – также осуществляет доопределение всех координат вектора состояний выходов, но путем суперпозиции m кубических покрытий, входящих в состав схемной структуры. В этом случае вычислительная сложность получения функционального покрытия равна $q = 2 \times m$, при условии, что структурные компоненты схемы предварительно были ранжированы в соответствии с порядком распространения сигналов, а покрытие каждого примитива имеет 2 куба.

Полученная структура вектора выходных значений функционального примитива изоморфна ДНФ, которая оперирует единичными термами булевых переменных. Аналогично можно записать КНФ, используя нулевой куб покрытия функционального примитива.

Результат минимизации интересен и для сжатия информации путем получения минимальной ДНФ, с помощью которой всегда можно восстановить исходную таблицу истинности. Вычислительная сложность минимизации в кубитном исчислении на многозначном алфавите линейна относительно числа переменных. Возникает естественный вопрос. Зачем минимизировать функцию, если она компактно представляется состояниями выходов по всем адресам, составленным из кодов от любого числа переменных. Единственная практически ориентированная цель – выполнение процедур обратной импликации для реализации регулярных алгоритмов синтеза тестов, которые на порядок улучшат свое быстродействие их выполнения за счет минимального числа кубов в покрытиях функциональных элементов.

2. Кубит-процессор (квантовый процессор)

Предлагаются кубитные (квантовые) структуры данных и вычислительных процессов для существенного повышения быстродействия при решении задач дискретной оптимизации. Описаны аппаратно ориентированные модели параллельного (за один цикл) вычисления булеана (множества всех подмножеств) на универсуме из n примитивов для решения задач покрытия, минимизации булевых функций, сжатия данных, синтеза и анализа цифровых систем за счет реализации процессорной структуры в форме диаграмме Хассе.

Цель создания кубит-процесора – существенное уменьшение времени при решении задач оптимизации путем параллельного вычисления векторных логических операций над множеством всех подмножеств от примитивных компонентов за счет увеличения памяти для хранения промежуточных данных. Задачи:

1. Определение структур данных для взятия булеана при решении задачи покрытия столбцов матрицы $M = |M_{ij}|, i = \overline{1,m}; j = \overline{1,n}$ единичными значениями строк. В частности, при m = n = 8, необходимо выполнить параллельно логическую операцию над 256 вариантами всех возможных сочетаний векторов (строк матрицы), составляющих булеан.

2. Система команд процессора должна включать следующие операции (and, or, xor) над векторами (словами), размерности m.

3. Разработка архитектуры кубит-процессора для параллельного вычисления $2^n - 1$ вариантов сочетаний, направленных на оптимальное решение NP-полной задачи покрытия.

4. Реализация прототипа кубит-процессора на базе программируемой логики PLD и верификация (валидация) аппаратного решения на примерах минимизации булевых функций.

5. Приведение других практических задач дискретной оптимизации к форме задачи покрытия для последующего решения на кубит-процессоре.

В качестве примера предлагается решить задачи поиска оптимального единичного покрытия всех столбцов минимальным числом строк матрицы M, представленной ниже:

M	1	2	3	4	5	6	7	8
a	1	.	.	.	.	1	.	.
b	.	.	1	.	.	.	1	.
c	1	.	.	.	1	.	1	.
d	.	1	.	1	.	.	.	1
e	.	1	.	.	1	.	.	.
f	1	.	1	.	.	1	1	.
g	.	1	.	1	.	.	.	1
h	.	.	1	.	1	.	.	.

Для этого необходимо сделать перебор всех 255 сочетаний: из восьми по одной строке, по двум, трем, четырем, пяти, шести, семи и восьми. Минимальное количество примитивов (строк), формирующее покрытие, есть оптимальное решение. Таких решений может быть несколько. Диаграмма Хассе есть компромиссное предложение, относительно времени и памяти, или такая стратегия решения задачи покрытия, когда ранее полученный результат впоследствии используется для создания более сложной суперпозиции. Поэтому для каждой таблицы покрытия, содержащей n примитивов (строк), необходимо генерировать собственную мультипроцессорную структуру в форме диаграммы Хассе, которая далее должна быть использована для почти параллельного решения NP-полной задачи. Например, для четырех строк таблицы покрытия диаграмма Хассе – структура мультипроцессора – будет иметь следующий вид:

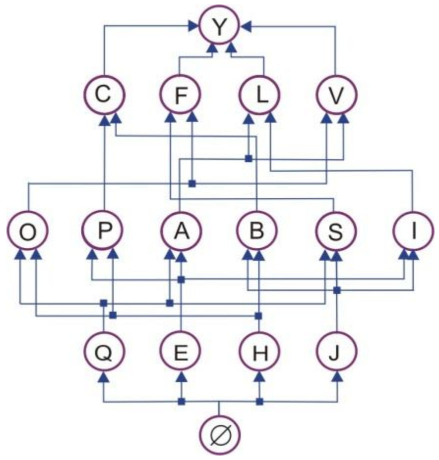

Рис. 1. Квантовая структура вычислительных процессов

Оптимальные решения задачи покрытия для матрицы M, которая генерирует 255 вариантов возможных сочетаний, представлены строками в форме ДНФ:

$$C = fgh \vee efg \vee cdf .$$

Автомат управления вычислительным процессом для квантовой структуры путем восходящего анализа вершин графа основан на последовательном выполнении описанных далее шагов. 1. Занесение информации о примитивах в регистры (матрицы) $L_i^1 = P_i$ первого уровня с последующим анализом качества покрытия каждым примитивом в двоичном формате (1 – есть покрытие, 0 – нет покрытия). Если один из примитивов организует покрытие $\mathop{\wedge}\limits_{j=1}^{m} L_{ij}^1 = 1$, процесс анализа структуры Хассе заканчивается. Иначе – выполнение перехода (r = r + 1) на следующий уровень графа:

$$L_i^1 = P_i \rightarrow \bigwedge_{j=1}^{m} L_{ij}^1 = \begin{cases} 0 \rightarrow n = n+1; \\ 1 \rightarrow \text{end}. \end{cases}$$

2. Инициирование команды обработки очередного (второго) уровня. Последовательное выполнение векторных (матричных) операций (or $L_i^r = L_{ij}^{r-1} \bigwedge_{j=1}^{m} L_{tj}^{r-1}$, and $\bigwedge_{j=1}^{m} L_{ij}^r = 1$) в целях анализа покрывающей способности сочетаний примитивных элементов r-уровня. Здесь $t = \overline{1,m}, i = \overline{1,m}, r = \overline{1,n}$, n – число ярусов или количество строк в таблице покрытия, m – число столбцов в ней. Если существует сочетание на рассматриваемом уровне, создающее полное покрытие, которое формирует оценку, равную 1, то обработка всех последующих уровней процессора не выполняется. В противном случае выполняется переход для анализа следующего уровня процессорной структуры:

$$L_i^r = L_{ij}^{r-1} \bigwedge_{j=1}^{m} L_{tj}^{r-1} \rightarrow \bigwedge_{j=1}^{m} L_{ij}^r = \begin{cases} 0 \rightarrow r = r+1; \\ 1 \rightarrow \text{end}. \end{cases}$$

Для поиска оптимального покрытия всегда достаточно двух элементов нижнего уровня, что означает – все операционные вершины имеют два регистровых (матричных входа), что существенно уменьшает стоимость аппаратурных затрат. Количество временных тактов для обработки структуры процессора в худшем случае равно n.

Выводы

Реализация квантового процессора на основе структуры Хассе позволила в n раз уменьшить аппаратурные затраты, по сравнению с число параллельной реализацией, что в свою очередь, уменьшило быстродействие процессора также в n раз. Вывод – необходимо искать новые структуры данных для снижения аппаратной стоимости квантовых вычислений, или более интеллектуальные алгоритмы решения задачи покрытия на диаграммах Хассе.

Научная новизна – впервые предложена модель данных и структура аппаратной реализации квантового компьютера, которая характеризуется использованием структуры Хассе, что дает возможность существенно (x100) повысить быстродействие решения практических задач дискретной оптимизации.

Практическая значимость – существенное повышение быстродействия при решении задач покрытия и других задач дискретной оптимизации за счет увеличения аппаратных затрат для параллельного выполнения векторных логических операций на Хассе-структуре квантового вычислительного устройства.

Литература

1. Beth, T. Quantum computing: an introduction [Text] / T. Beth // Proceedings of the IEEE International Symposium on Circuits and Systems, ISCAS. – 2000.– Geneva.– Vol. 1.– P. 735 – 736.

2. Jonker, P. On quantum and classical computing with arrays of superconducting persistent current qubits [Text] / P. Jonker, Jie Han // Proceedings of Fifth IEEE International Workshop on Computer Architectures for Machine Perception. – 2000.– P. 69 – 78.

3. Keyes, R.W. Challenges for quantum computing with solid-state devices [Text] / R.W. Keyes // Computer.– Jan. 2005.– Vol. 38, Issue 1.– P. 65 – 69.

4. Glassner, A. Quantum computing. 3. [Text] / A. Glassner // IEEE Computer Graphics and Applications.– Nov/Dec 2001.– Vol. 21, Issue 6.– P. 72 – 82.

5. Marinescu, D.C. The Promise of Quantum Computing and Quantum Information Theory – Quantum Parallelism [Text] / D.C. Marinescu // Proceedings of the 19th IEEE International Parallel and Distributed Proc/ Symposium (IPDPS'05).– 2005.– P. 112-114.

6. Oskin, M. A practical architecture for reliable quantum computers [Text] / M. Oskin, F.T. Chong, I.L. Chuang // Computer. – Jan. 2002. – Vol. 35, No. 1.- P. 79 – 87.

7. Glassner, A. Quantum Computing, Part 1 [Text] / A. Glassner // IEEE Computer Graphics and Applications.– July/August 2001.– P. 84 – 92.

8. Fault-tolerant computing with biased-noise superconducting qubits [Electronic resource] / P. Aliferis, F. Brito, D.P. DiVincenzo, J. Preskill, M. Steffen, B.M. Terhal // New Journal of Physics. – Jan. 30, 2009. – 19 p. – Access to: http://iopscience. iop.org/1367-2630/11/1/013061 /pdf/1367-2630_11_1_ 013061.pdf. . – 23.01.2012 г.

9. Huntting, B. Introduction to Quantum Computing [Electronic resource] / B. Huntting, D. Mertz. – Access to: http://www.ibm.com/developerworks/linux/ library/l-quant/index.html. – 23.01.2012 г.

10. DiVincenzo, D.P. The Physical Implementation of Quantum Computation [Electronic resource] / D.P. DiVincenzo // IBM T.J. Watson Research Center, Yorktown Heights, NY 10598 USA. – July 9, 2004. – Access to: http://arxiv.org/PS_cache/quant-ph/pdf/ 0002/ 0002077v3.pdf.

11. Инфраструктура мозгоподобных вычислительных процессов [Текст] / М.Ф. Бондаренко, О.А. Гузь, В.И. Хаханов, Ю.П. Шабанов-Кушнаренко. – Х.: Новое Слово. – 2010. – 160 с.

12. Whitney Mark Gregory. Practical Fault Tolerance for Quantum Circuits [Text] / Mark Gregory Whitney // PhD Dissertation in Computer Science. Berkeley: University of California, 2009. – 206 p.

13. Логический ассоциативный вычислитель [Текст] / В.И. Хаханов, Е.И. Литвинова, С.В. Чумаченко, О.А. Гузь // Электронное моделирование. – № 1. – 2011. – С. 73 – 90.

14. Information analysis infrastructure for diagnosis. [Text] / V. Hahanov, Wajeb Gharibi, E. Litvinova, S. Chumachenko // Information. An international interdisciplinary journal. – 2011. – Japan. – Vol. 14. – No 7. – P. 2419 - 2433.

15. Хаханов, В.И. Проектирование и тестирование цифровых систем на кристаллах [Текст] / В.И. Хаханов, Е.И. Литвинова, О.А. Гузь. – Х.: ХНУРЭ, 2009.– 484 с.

16. Stenholm, S. Quantum approach to informatics [Text] / S. Stenholm, K.-A. Suominen // Published by John Wiley & Sons, Inc. – Hoboken, New Jersey. – 2005. – 238 p.

17. Nielsen, M.A. Quantum Computation and Quantum Information [Text] / M.A. Nielsen, I.L. Chuang // Published in the United States of America by Cambridge University Press. – New York. – 2010. – 676 p.

Поступила в редакцию 12.02.2012

Рецензент: д-р техн. наук, проф. И.А. Фурман, Харьковский национальный технический университет сельского хозяйства им. Петра Василенко, Харьков, Украина.

КВАНТОВІ МОДЕЛІ ДАНИХ І ОБЧИСЛЮВАЛЬНИХ ПРОЦЕСІВ

В.І. Хаханов, Murad Ali Abbas, Є.І. Литвинова, О.А. Гузь,
І.В. Хаханова, Baghdad Ammar Avni Abbas

Запропоновано кубітні (квантові) структури даних і обчислювальних процесів для істотного підвищення швидкодії при вирішенні задач пошуку, розпізнавання, прийняття рішень, дискретної оптимізації та відмовостійкого проектування. Описано апаратно-орієнтовані моделі паралельного (за один цикл) обчислення булеана (множини всіх підмножин) на універсумі з n примітивів для вирішення задач покриття, мінімізації булевих функцій, стиснення даних, синтезу та аналізу цифрових систем за рахунок реалізації процесорної структури у формі діаграми Хассе. Представлено прототип квантового пристрою, реалізованого на основі програмованої логіки для оптимального вирішення задачі покриття при аналізі кіберпростору.

Ключові слова: квантові обчислення, кубіт, булеан, програмовна логіка.

QUANTUM DATA AND COMPUTING MODELS

V.I. Hahanov, Murad Ali Abbas, E.I. Litvinova, O.A. Guz,
I.V. Hahanova, Baghdad Ammar Avni Abbas

Qubit (quantum) structures of data and computational processes for significantly improving performance when solving problems of discrete optimization and fault-tolerant design are proposed. We describe a hardware-focused models for parallel (one cycle) calculating the power set (the set of all subsets) on the universe of n primitives for solving coverage problems, minimization of Boolean functions, data compression, analysis and synthesis of digital systems through the implementation of the processor structure in the form of the Hasse diagram. A prototype of quantum device, implemented by programmable logic, is described.

Key words: quantum computing , qubit, power set, programming logic.

Хаханов Владимир Иванович – д-р техн. наук, декан факультета компьютерной инженерии и управления, профессор кафедры автоматизации проектирования вычислительной техники Харьковского национального университета радиоэлектроники.

Abbas Murad Ali– аспирант кафедры автоматизации проектирования вычислительной техники Харьковского национального университета радиоэлектроники.

Литвинова Евгения Ивановна – д-р техн. наук, профессор кафедры автоматизации проектирования вычислительной техники Харьковского национального университета радиоэлектроники.

Гузь Олеся Алексеевна – канд. техн. наук, доцент, зав. кафедрой специализированных компьютерных систем Донецкой академии автомобильного транспорта.

Хаханова Ирина Витальевна – д-р техн. наук, профессор кафедры автоматизации проектирования вычислительной техники Харьковского национального университета радиоэлектроники.

Abbas Baghdad Ammar Avni– аспирант кафедры автоматизации проектирования вычислительной техники Харьковского национального университета радиоэлектроники.