УДК 658.512.011:681.326:519.713

В. И. Хаханов, Е. И. Литвинова, С. В. Чумаченко, доктора техн. наук
Харьковский национальный университет радиоэлектроники
(Украина, 61166, Харьков, пр. Ленина 14,
тел. (057) 70-21-421, (057) 70-21-326,
E-mail: hahanov@kture.kharkov.ua, kiu@kture.kharkov.ua),
О.А.Гузь, канд. техн. наук
Донецкая академия автомобильного транспорта
(Украина, 83086 Донецк, пр. Дзержинского, 7,
тел. (063) 470-52-09, E-mail: olesya_guz@ukr.net)

Логический ассоциативный вычислитель

Предложена архитектура быстродействующего вычислителя для параллельного анализа информации, представленной в виде аналитических, графовых и табличных форм ассоциативных отношений, при поиске, распознавании и принятии решений в *n*-мерном векторном дискретном пространстве. Рассмотрены векторно-логические модели (алгоритмы или вычислительные схемы) актуальных прикладных задач, качество решения которых оценивается введенной интегральной неарифметической метрикой взаимодействия булевых векторов.

Запропоновано архітектуру швидкодіючого обчислювача для паралельного аналізу інформації, представленої у вигляді аналітичних, графових і табличних структур асоціативних відношень, для пошуку, розпізнавання та прийняття рішень у *n*-вимірному векторному дискретному просторі. Розглянуто векторно-логічні моделі (алгоритми або обчислювальні схеми) актуальних прикладних задач, якість розв'язку яких оцінюється введеною інтегральною неарифметичною метрикою взаємодії булевих векторів.

К л ю ч е в ы е с л о в а: вычислитель, анализ информации, логическое ассоциативное отношение, процесс-модель.

Удаление из компьютера громоздких арифметических операций даст возможность направить освободившиеся ресурсы в ассоциативно-логическую архитектуру, использующую векторные операции and, or, not, xor, что существенно (×100) повысит быстродействие решения неарифметических задач. Исключение арифметических операций, использование параллелизма алгебры векторной логики, мультипроцессорность позволяет создать эффективную архитектуру, объединяющую математические и технологические подходы при решении прикладных задач, таких как поиск, распознавание и принятие решений. Рыночная привлекательность логического ассоциативного вычислителя определяется тысячами старых и новых ло-

гических по своей природе задач, в настоящее время неэффективно решаемых на избыточных универсальных компьютерах с мощным арифметическим процессором.

Цель исследования — существенное повышение быстродействия процедур поиска, распознавания и принятия решений посредством параллельной реализации ассоциативно-логических векторных операций для анализа графовых и табличных структур данных в дискретном булевом пространстве без использования арифметических операций. Для достижения поставленной цели необходимо разработать:

неарифметическую метрику оценивания ассоциативно-логических решений;

структуры данных и модели вычислительных процессов (МВП) для решения актуальных задач;

архитектуру логического ассоциативного вычислителя и направления его практического использования.

Кроме того, необходимо создать МВП экспертного обслуживания запросов в реальном масштабе времени, реализованную в виде функциональных блоков цифровой системы на кристалле на основе ассоциативно-логических структур данных для получения детерминированного решения, состоятельность которого оценивается неарифметическим интегральным критерием качества взаимодействия запроса с заданным дискретным пространством.

Объектом исследования являются МВП поиска, распознавания и принятия решений в дискретном булевом пространстве на основе использования алгебры векторной логики, вычислительной платформы анализа ассоциативно-логических структур данных и неарифметического интегрального критерия качества.

Ассоциативно-логические структуры данных и МВП поиска, распознавания и выбора решения на основе неарифметического интегрального критерия качества исследованы с помощью вычислительной системы на кристалле, оперирующей векторными логическими операциями.

В процессе исследований использованы такие источники научно-технической информации: аппаратная платформа ассоциативно-логического анализа информации [1—4]; ассоциативно-логические структуры данных для решения информационных задач [5—8]; модели и методы дискретного анализа и синтеза [9—12]; вычислительные средства решения информационно-логических задач [13—19]; ассоциативно-логические вычисления [20—25].

Интегральная метрика оценивания решения. Архитектура ассоциативно-логического вычислителя включает модели, методы и структу-

ры данных, ориентированные на аппаратную поддержку процессов поиска, распознавания и принятия решений [22—24] на основе векторных неарифметических операций. Оценка решения задачи определяется векторно-логическим критерием качества взаимодействия запроса (вектора m) с системой ассоциативных векторов (ассоциаторов), в результате которого генерируется конструктивный ответ в виде одного или нескольких ассоциаторов, а также численной характеристики степени принадлежности (функции качества) входного вектора m к найденному решению: $\mu(m \in A)$.

Входной вектор $m = (m_1, m_2, ..., m_i, ..., m_q)$, $m_i \in \{0, 1, x\}$ и матрица A_i ассоциаторов $A_{ijr} (\in A_{ij} \in A_i \in A) = \{0, 1, x\}$ имеют одинаковую размерность q. Далее степень принадлежности m-вектора вектору A будем обозначать $\mu(m \in A)$.

Существует пять типов теоретико-множественного (логического) Δ-взаимодействия двух векторов $m \cap A$. Они формируют все примитивные варианты реакции обобщенной системы поиска, распознавания и принятия решения на входной вектор-запрос. В технологической отрасли знаний — технической диагностике — указанная последовательность действий изоморфна маршруту: поиск дефектов, их распознавание, принятие решения на восстановление работоспособности. Эти три стадии технологического маршрута нуждаются в метрике оценивания решений для выбора оптимального варианта.

Понятия принадлежности и непринадлежности являются взаимодополняющими, но в данном случае технологичнее вычислять непринадлежность. Следовательно, необходимый критерий качества равен нулю, когда два вектора равны между собой. Оценка качества взаимодействия двух двоичных векторов убывает по мере возрастания критерия качества от нуля до единицы. Для того чтобы окончательно исключить арифметические операции при подсчете полученного векторного критерия качества, необходимо объединить три оценки в одну:

$$\begin{gathered} Q = d(m, A) \vee \mu(m \in A) \vee \mu(A \in m) = m \oplus A, \\ d(m, A) = m \oplus A, \\ \mu(m \in A) = A \wedge \overline{m \wedge A}; \\ \mu(A \in m) = m \wedge \overline{m \wedge A}. \end{gathered} \tag{1}$$

Здесь критерии представлены векторами, определяющими взаимодействие компонентов m, A. При этом увеличение числа нулей в трех векторах качества повышает критерий, а наличие единиц обусловливает ухудшение качества взаимодействия.

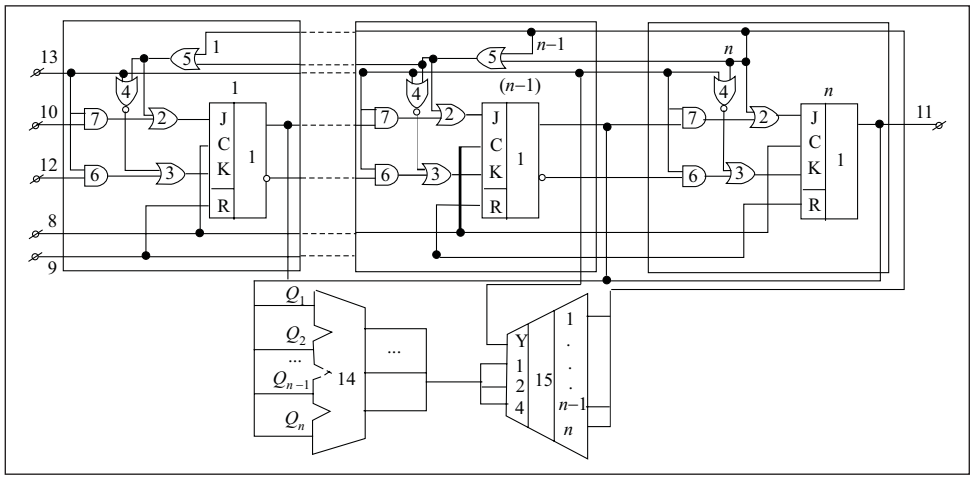


Рис. 1. Регистр сдвига и уплотнения единиц

Для сравнения оценок необходимо определять число единиц в каждом векторе без выполнения операций суммирования. Это можно сделать с помощью регистра сдвига [4] (рис. 1), который позволяет за один такт выполнить операцию slc (shift left bit crowding) — сдвиг влево с одновременным уплотнением всех единичных координат n-разрядного двоичного вектора. После процедуры сжатия номер правого единичного бита уплотненной серии единиц формирует значение критерия качества взаимодействия векторов. Для двоичных наборов $m=(110011001100)$, $A=(000011110101)$ определение качества их взаимодействия по формулам (1) представим в следующем виде (нулевые координаты отмечены точками):

m	1	1	.	.	1	1	.	.	1	1	.	.
A	.	.	.	.	1	1	1	1	.	1	.	1
$m \wedge A$	.	.	.	.	1	1	.	.	.	1	.	.
$\overline{m \wedge A}$	1	1	1	1	.	.	1	1	1	.	1	1
$d(m,A)=m \oplus A$	1	1	.	.	.	.	1	1	1	.	.	1.
$\mu(A \in m)=m \wedge \overline{m \wedge A}$	1	1	.	.	.	.	.	.	1	.	.	.
$\mu(m \in A)=A \wedge \overline{m \wedge A}$	.	.	.	.	.	.	1	1	.	.	.	1
$Q=d(m,A) \vee \mu(m \in A) \vee \mu(A \in m)$	1	1	.	.	.	.	1	1	1	.	.	1
$Q(m,A)=(6/12)$	1	1	1	1	1	1	.	.	.	.	.	.

Здесь сформирована оценка взаимодействия векторов $Q(m, A) = (6/12)$ и, что самое главное, единичные координаты строки $Q = d(m, A) \vee \mu(m \in A) \vee$

$\vee\,\mu\,(A \in m)$ идентифицируют все существенные переменные, по которым взаимодействие векторов не соответствует критерию качества.

Для сравнения двух решений, полученных в результате логического анализа, используем сжатые с помощью slc-операции векторы качества Q, над которыми выполняется процедура, включающая следующие векторные операции:

$$Q\,(m,A) = \begin{cases} Q_1(m,A) \leftarrow \text{or}\,[Q_1(m,A) \wedge Q_2(m,A) \oplus Q_1(m,A)] = 0; \\ Q_2(m,A) \leftarrow \text{or}\,[Q_1(m,A) \wedge Q_2(m,A) \oplus Q_1(m,A)] = 1. \end{cases} \qquad (2)$$

Вектор-бит — or-оператор редукции — формирует двоичное битовое решение на основе применения логической операции or к n разрядам вектора существенных переменных критерия качества. Схемотехническое решение процедуры выбора

$$Q = \begin{cases} Q_1 \leftarrow Y = 0, \\ Q_2 \leftarrow Y = 1 \end{cases}$$

и аналитическая МВП имеют три операции: $Y = \vee\,[(Q_1 \wedge Q_2) \oplus Q_1\,]$.

Для двоичных векторов, представляющих собой критерии качества, выполняем процедуру выбора лучшего из них на основании выражения (2):

$Q_1(m,A) = (6,12)$	1	1	1	1	1	1	.	.	.	.	.	.
$Q_2(m,A) = (8,12)$	1	1	1	1	1	1	1	1	.	.	.	.
$Q_1(m,A) \wedge Q_2(m,A)$	1	1	1	1	1	1	.	.	.	.	.	.
$Q_1(m,A) \oplus Q_1(m,A) \wedge Q_2(m,A)$	.	.	.	.	.	.	.	.	.	.	.	.
$Q(m,A) = Q_1(m,A)$	1	1	1	1	1	1	.	.	.	.	.	.

Векторные логические критерии качества взаимодействия ассоциативных наборов позволяют получать оценку поиска, распознавания и принятия решения с высоким быстродействием логических параллельных операций, что особенно существенно для критических систем реального времени.

Аналитическая форма записи обобщенной МВП для выбора лучшего взаимодействия входного запроса m с системой логических ассоциативных отношений имеет вид

$$P\,(m,A) = \min Q_i\left(m \overset{n}{\underset{i=1}{\Delta}} A_i \right) = \vee \left[\left(Q_i \overset{j \neq i}{\underset{j=1,n}{\wedge}} Q_j \right) \oplus Q_i \right] = 0\,;$$

$$Q\,(m,\,A) = (Q_1\,,\,Q_2\,,\,...,\,Q_i\,,...,Q_n)\,;$$

$$A = (A_1\,,\,A_2\,,...,\,A_i\,,...,\,A_n)\,;$$

$$\Delta = \{\text{and, or, xor, not, slc, nop}\};$$
$$A_i = (A_{i1}, A_{i2}, ..., A_{ij}, ..., A_{is});$$
$$A_{ij} = (A_{ij1}, A_{ij2}, ..., A_{ijr}, ..., A_{msq}); \quad (3)$$
$$m = (m_1, m_2, ..., m_r, ..., m_q);$$
$$Q_i = d(m, A_i) \vee \mu(m \in A_i) \vee \mu(A_i \in m),$$
$$d(m, A_i) = m \oplus A_i;$$
$$\mu(m \in A_i) = A_i \wedge \overline{m \wedge A_i};$$
$$\mu(A_i \in m) = m \wedge \overline{m \wedge A_i}.$$

Здесь выражение $P(m, A)$, определяющее функциональность, можно представить как аналитическую МВП в виде высказывания, минимизирующего интегральный критерий качества; структуры данных представлены в виде совокупности таблиц $A = (A_1, A_2, ..., A_i, ..., A_m)$, логически взаимодействующих между собой; каждая таблица задается упорядоченной совокупностью вектор-строк ассоциативной таблицы $A_i = (A_{i1}, A_{i2}, ...$ $..., A_{ij}, ..., A_{is})$ явных решений, а строка $A_{ij} = (A_{ij1}, A_{ij2}, ..., A_{ijr}, ..., A_{msq})$ представляет собой истинное высказывание.

Функционал, представленный в виде таблицы, не имеет постоянных во времени входных и выходных переменных. Равнозначность всех переменных в векторе $A_{ij} = (A_{ij1}, A_{ij2}, ..., A_{ijr}, ..., A_{msq})$ создает одинаковые условия их существования, что означает инвариантность решения задач прямой и обратной импликации в пространстве $A_i \in A$. Ассоциативный вектор A_{ij} определяет собой явное решение, где каждая переменная задается в конечном, многозначном и дискретном алфавите $A_{ijr} \in \{\alpha_1, \alpha_2, ...$ $..., \alpha_i, ..., \alpha_k\} = \beta$. Взаимодействие $P(m, A)$, входного вектора-запроса $m = (m_1, m_2, ..., m_r, ..., m_q)$ с множеством $A = (A_1, A_2, ..., A_i, ..., A_m)$, формирует решения с выбором лучшего из них по минимальному критерию качества:

$$P(m, A) = \min Q_i [m \wedge (A_1 \vee A_2 \vee ... \vee A_i \vee ... \vee A_m)].$$

Предложенная МВП анализа ассоциативных таблиц и введенные критерии качества получаемых решений являются основой для разработки специализированной мультипроцессорной архитектуры, ориентированной на параллельное выполнение векторных логических операций.

Архитектура логического ассоциативного вычислителя. Лучшее решение может быть получено в результате объединения достоинств центрального процессорного устройства, программируемой логической мат-

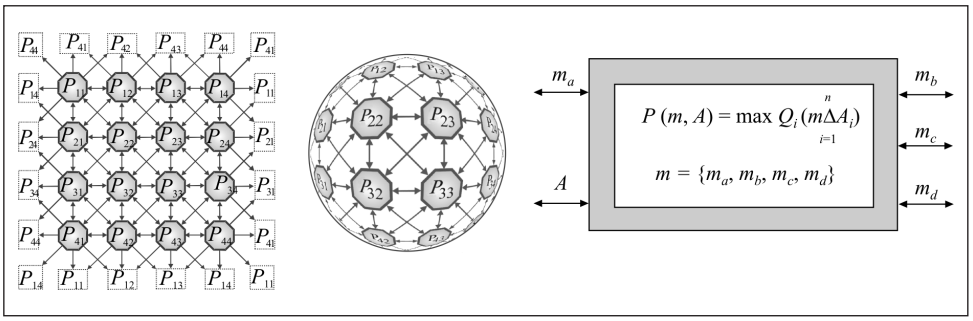


Рис. 2. Макроархитектура вычислителя и интерфейс

рицы и заказной СБИС [15, 16], таких как гибкость программирования, возможность корректирования исходных кодов; минимальное число команд и простые схемотехнические решения аппаратной реализации мультипроцессора; распараллеливание логических процедур на структуре однобитовых процессоров. Реализация вычислителя в кристалле заказной СБИС дает возможность получать максимальную тактовую частоту, минимальную стоимость микросхемы при больших объемах выпуска изделия, низкое энергопотребление. Объединение преимуществ перечисленных технологий определяет базовую конфигурацию вычислителя, имеющего сферическую структуру (рис. 2), состоящую из 16 векторных секвенсоров — устройств последовательного управления (УПУ), каждый из которых, включая граничные элементы, соединен с восемью соседними. Прототипом данного вычислителя является процессор PRUS, описанный в работе [17].

Занесение информации в процессор выполняется по классической схеме процесса проектирования, за исключением стадии размещения и трассировки, которая заменяется фазой распределения программ и данных между всеми логическими бит-процессорами, работающими параллельно. Компилятор обеспечивает распределение данных по процессорам, задает время формирования решения на выходе каждого из них, а также планирует передачу полученных результатов другому процессору.

Вычислитель — это эффективная сеть процессоров, которая обрабатывает данные и обеспечивает обмен информацией между компонентами сети в процессе их решения. Простая схемотехника каждого процессора позволяет эффективно обрабатывать сверхбольшие массивы, насчитывающие миллионы бит информации, затрачивая на это в сотни раз меньше времени по сравнению с универсальным процессором.

Базовая ячейка — векторный процессор для вычислителя может быть синтезирован на 200-х вентилях, что дает возможность сеть, содержащую

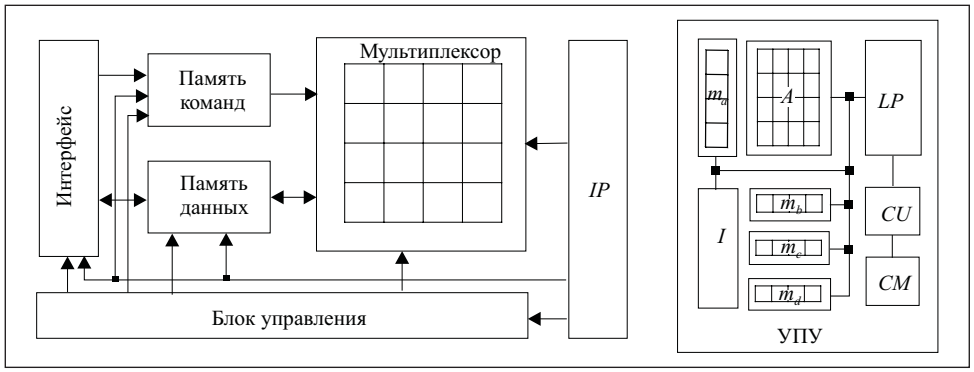


Рис. 3. Архитектура вычислителя и структура УПУ

4096 вычислителей, легко реализовать в кристалле заказной СБИС, используя современную кремниевую технологию. Поскольку затраты памяти для хранения данных весьма незначительны, вычислитель может быть использован при проектировании систем управления в таких областях человеческой деятельности, как промышленное производство, защита информации, медицина, искусственный интеллект, космонавтика, геология, метеорология. Особый интерес вычислитель представляет для цифровой обработки данных, распознавания образов и криптоанализа.

Однако основное назначение вычислителя — получение квазиоптимального решения в интегрированной задаче поиска и (или) распознавания с использованием компонентов архитектуры, ориентированных на выполнение векторных логических операций:

$$P(m,A)=\min Q_i\left(m\overset{n}{\underset{i=1}{\Delta}}A_i\right),\ \ m=\{m_a,m_b,m_c,m_d\}.$$

Интерфейс системы, соответствующий данному функционалу, представлен на рис. 2. Все компоненты $\{A,m_a,m_b,m_c,m_d\}$ могут быть как входными, так и выходными. Двунаправленная детализация интерфейса связана с инвариантностью отношения всех переменных, векторов, A-матрицы и компонентов к входам и (или) выходам архитектуры. Поэтому структурная модель системы вычислителя может быть использована для решения любых задач прямой и обратной импликации в дискретном логическом пространстве, в чем и заключается ее отличие от концепции автоматной модели вычислительного устройства с выраженными входами и выходами. Компоненты или регистры $m=\{m_a,m_b,m_c,m_d\}$ используются для получения решения в виде буферных, входных и выходных векторов, а также для идентификации оценки качества удовлетворения входного запроса.

Один из возможных вариантов архитектуры вычислителя представлен на рис. 3. Основным ее компонентом является матрица $P = [P_{ij}], \mathrm{card}\,(4 \times 4)$, содержащая 16 вектор-процессоров, каждый из которых предназначен для выполнения пяти логических векторных операций над памятью данных, представленной в виде таблицы, размерностью $A = \mathrm{card}\,(m \times n)$.

В блоке интерфейса происходит обмен данными и загрузка программы обработки данных в соответствующую память команд. Блок управления инициирует выполнение команд логической обработки данных и синхронизирует функционирование всех компонентов мультипроцессора. Блок *IP* [1] предназначен для сервисного обслуживания всех модулей, диагностирования дефектов и восстановления работоспособности компонентов и устройства в целом. Элементарный логический ассоциативный процессор или УПУ, входящий в состав вычислителя, содержит логический процессор *LP*, ассоциативную *A*-матрицу (память) для параллельного выполнения базовых операций, блок векторов *m*, предназначенный для параллельного обслуживания строк и столбцов *A*-матрицы, а также обмена данными в процессе вычислений, память прямого доступа *CM*, сохраняющую команды программы обработки информации, автомат управления *CU* выполнением логических операций, интерфейс *I* связи УПУ с другими элементами и устройствами вычислителя.

Логический процессор *LP* (рис. 4) выполняет пять операций (and, or not, xor, slc), являющихся базовыми для создания алгоритмов и процедур информационного поиска и оценивания решения. Модуль *LP* имеет

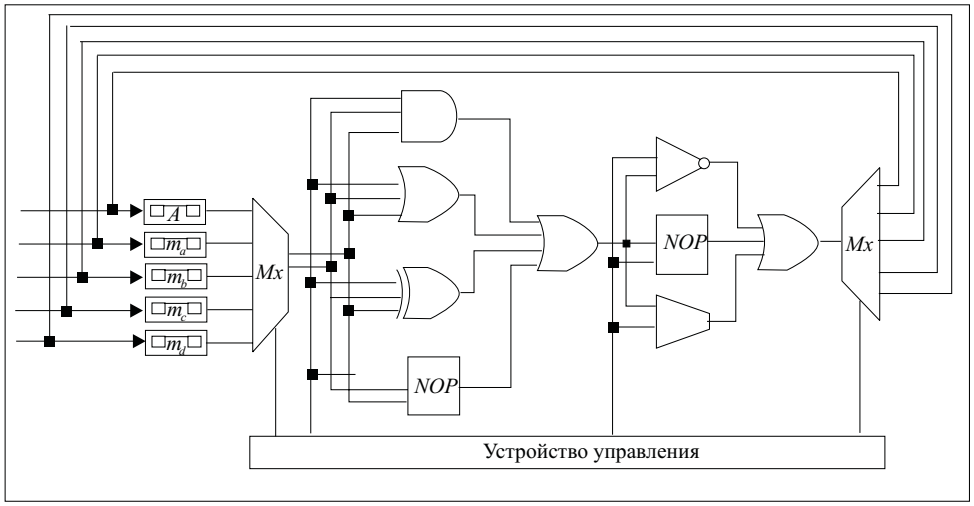


Рис. 4. Функциональная схема блока логических вычислений

мультиплексор на входе для выбора одного из пяти операндов, который подается на выбранный логический векторный оператор. Сформированный результат через мультиплексор (элемент or) заносится в один из четырех операндов, выбираемый соответствующим адресом.

Особенности реализации логического процессора заключаются в наличии трех бинарных (and, or, xor) и двух унарных (not, slc) операций. Последние можно присоединять к такту обработки регистровых данных, выбрав одну из трех операций (not, slc, nop — нет операции). Для повышения эффективности работы логического устройства вводятся два элемента с пустой операцией. Если необходимо выполнить только унарную операцию, то на уровне бинарных команд следует выбрать nop, что практически означает передачу данных через повторитель ко второму уровню унарных операций.

Все операции в *LP* — регистровые или регистрово-матричные. Последние предназначены для анализа вектор-строк таблицы при использовании входного *m*-вектора как запроса для точного поиска информации. В блоке логических вычислений допустимо следующее сочетание операций и операндов:

$$C = \begin{cases} \{m_a, m_b, m_c, m_d\} \Delta A_i; \\ \{m_a, m_b, m_c, m_d\} \Delta \{m_a, m_b, m_c, m_d\}; \\ \{\text{not}, \text{nop}, \text{slc}\} \{m_a, m_b, m_c, m_d, A_i\}. \end{cases}$$

$$\Delta = \{\text{and, or, xor}\}$$

Реализация всех векторных операций блока логических вычислений, выполняемых с тактовой частотой 100 МГц, для одного УПУ в среде Verilog с последующей послесинтезной реализацией в кристалле программируемой логики Virtex 4, Xilinx содержит 2400 эквивалентных вентилей.

МВП векторно-логического анализа. Метрика оценивания решения при использовании модели определена двоичным логическим вектором в дискретном булевом пространстве. Концептуальная модель вычислительного изделия представлена совокупностью управляющего и операционного автоматов. В модели функциональности использована технология создания иерархических цифровых систем с локальной синхронизацией отдельных модулей и одновременно глобальной асинхронностью функционирования всего устройства [15].

Для детализации структуры векторного процессора и УПУ рассмотрим аналитические и структурные МВП, выполняющие анализ *A*-матрицы по столбцам или строкам. Первая из них представлена на рис. 5, *а* и предназначена для определения множества допустимых решений отно-

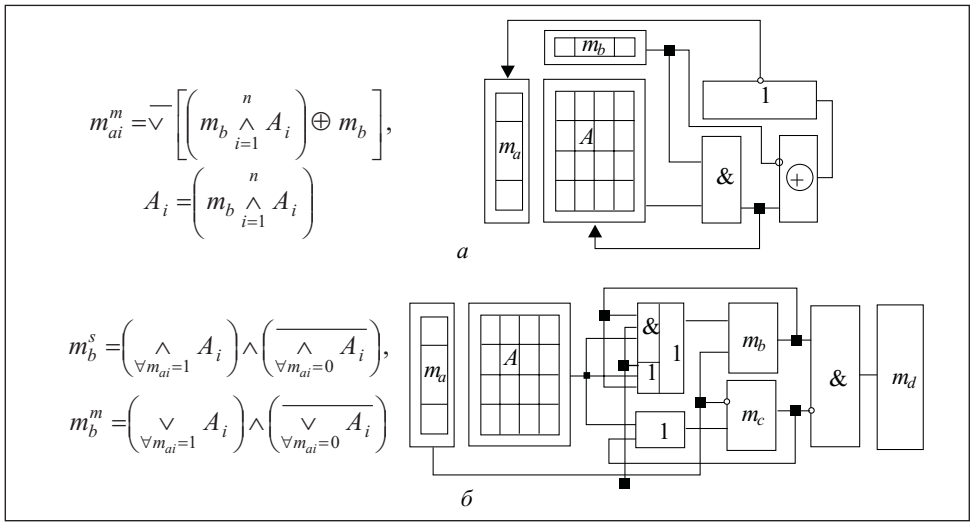


Рис. 5. Модели поиска всех допустимых решений (*а*) и выбора оптимального решения (*б*)

сительно входного запроса m_b, вторая (рис. 5, *б*) осуществляет поиск оптимального решения на множестве строк, найденных с помощью первой модели в результате их анализа. Возможно и самостоятельное применение второй модели, ориентированное на определение однозначного и многозначного решения при поиске дефектов в цифровой системе на кристалле.

Все операции, выполняемые двумя МВП — векторные. Модель анализа строк (см. рис. 5, *а*) формирует вектор m_a идентификации допустимых $m_{ai} = 1$ или противоречивых $m_{ai} = 0$ решений относительно входного условия m_b за n тактов обработки всех m-разрядных векторов таблицы $A = \text{card}\,(m \times n)$. Качество (допустимость) решения определяется для каждого взаимодействия входного вектора m_b и строки $A_i \in A$ на блоке (редукции) дизъюнкции. Матрица A может быть модифицирована ее пересечением с входным вектором на основе использования операции $A_i = (m_b \underset{i=1}{\overset{n}{\wedge}} A_i)$, если необходимо исключить из A-таблицы все незначимые для решения координаты и векторы, отмеченные единичными значениями вектора m_a.

Решение задач диагностирования посредством анализа строк таблицы (см. рис. 5, *б*) осуществляется так. После выполнения диагностического эксперимента формируется двоичный вектор экспериментальной проверки m_a, маскирующий A-таблицу неисправностей для поиска одиночных

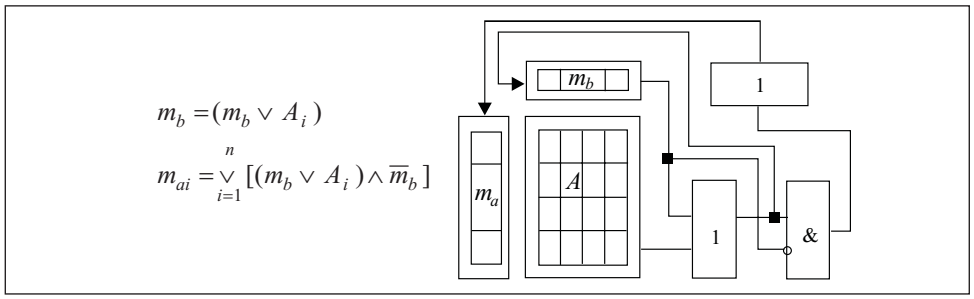


Рис. 6. МВП поиска квазиоптимального покрытия

или кратных дефектов. Векторы m_b и m_c используются для накопления результатов выполнения операций конъюнкции и дизъюнкции. Затем выполняется логическое вычитание (xor-операция) из первого регистра m_b содержимого второго вектора m_c с последующей записью результата в регистр m_d.

Для реализации второго уравнения, которое формирует множественное решение, элемент and заменяется функцией or. В схеме используется также переменная выбора режима поиска решения: single или multiple. В качестве входного условия в модели использован вектор m_a, управляющий выбором векторной операции and, or для обработки единичных $A_i(m_{ai} = 1) \in A$ или нулевых $A_i(m_{ai} = 0) \in A$ строк A-таблицы.

В результате выполнения n тактов осуществляется накопление единичных и нулевых относительно значений координат вектора m_a решений в регистрах A_1, A_0. Априори в указанные регистры заносится вектор единиц и нулей: $A_1 = 1$, $A_0 = 0$. После обработки всех n строк A-таблицы за n тактов выполняется векторная конъюнкция содержимого регистра A_1 с инверсией регистра A_0, которая формирует результат в виде вектора m_b, где единичные значения координат определяют решение. В таблице неисправностей цифрового изделия единичным координатам вектора m_b соответствуют столбцы, отождествляемые с номерами дефектов или неисправных блоков, подлежащих восстановлению или ремонту.

При сервисном обслуживании функциональных модулей можно на универсальной структуре системы векторного логического анализа решить оптимизационную задачу восстановления работоспособности. С помощью минимального числа ремонтных запасных строк и (или) столбцов, например памяти, необходимо обеспечить квазиоптимальное покрытие всех обнаруженных в ячейках неисправностей. Технологическая и математическая составляющие векторной логики в данном случае обусловливает простое схемотехническое решение для получения квазиоптимального покрытия (рис. 6), преимущества которого заключаются в следующем.

1. Вычислительная сложность процедуры: число векторных операций, равное числу строк таблицы, $Z = n$.

2. Минимум аппаратных затрат: таблица и два вектора (m_b, m_a) для хранения промежуточных покрытий и накопления результата в виде единичных координат, соответствующих строкам таблицы, которые составляют квазиоптимальное покрытие.

3. Отсутствие классического деления задачи покрытия на поиск ядра покрытия и дополнения.

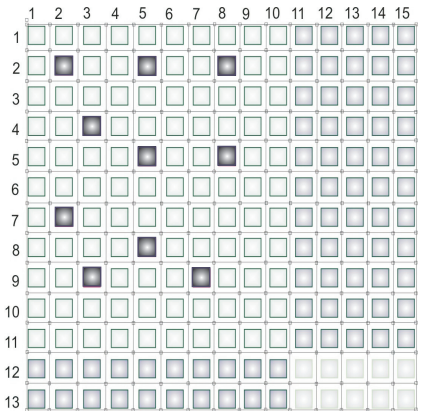
4. Отсутствие сложных процедур манипулирования ячейками строк и столбцов. При этом получение не всегда оптимального покрытия — недостаток, который компенсируется технологичностью векторной процедуры, представленной на рис. 6.

Операция редукции на последнем этапе превращает векторный результат в бит m_{ai} вектора m_a по функции or $m_{ai} = \vee [(m_b \vee A_i) \wedge \overline{m}_b]$. В общем случае операция редукции в алгебре векторных операций записывается в виде <бинарная операция><вектор>: $\vee A_i$, $\wedge m$, $\overline{\wedge}(m \vee A_i)$. Обратная процедура — векторизация есть конкатенация булевых переменных: $m_a(a, b, c, d, e, f, g, h)$. В процедуре поиска покрытия априори векторы m_b и m_a становятся равными нулю. Квазиоптимальное покрытие накапливается за n тактов в векторе m_a последовательным сдвигом. Биты, заносимые в регистр m_a, формируются схемой or, которая выполняет редукцию после анализа полученного результата $[(m_b \vee A_i) \wedge \overline{m}_b]$ на наличие единиц.

Представляет интерес функциональная законченность цикла диагностирования, когда после получения квазиоптимального покрытия данная информация используется для восстановления работоспособности дефектных ячеек памяти [19]. Размерность модуля памяти (13 × 15 ячеек) не влияет на вычислительную сложность получения покрытия десяти дефектных ячеек с помощью резервных строк и столбцов (рис. 7).

Для решения оптимизационной задачи выполняется построение таблицы покрытия (см. рис. 7) неисправных ячеек, в которой строки — резервные ресурсы для покрытия дефектов (C_2, C_3, C_5, C_7, C_8, R_2, R_4, R_5, R_7, R_8, R_9), а столбцы — дефекты ячеек ($F_{2,2}$, $F_{2,5}$, $F_{2,8}$, $F_{4,3}$, $F_{5,5}$, $F_{5,8}$, $F_{7,2}$, $F_{8,5}$, $F_{9,3}$, $F_{9,7}$), подлежащие ремонту. При этом столбцы соответствуют координатам дефектных ячеек, а строки идентифицируют резервные компоненты (строки и столбцы), которые могут восстановить работоспособность неисправных координат. Модель вычислительного процесса, представленная на рис. 5, *а*, дает возможность получить оптимальное решение в виде

$$m_a = 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0,$$



1	.	.	.	.	.	1	.	.	.
.	.	.	1	.	.	.	.	1	.
.	1	.	.	1	.	.	1	.	.
.	.	.	.	.	.	.	.	.	1
.	.	1	.	.	1	.	.	.	.
1	1	1	.	.	.	.	.	.	.
.	.	.	1	.	.	.	.	.	.
.	.	.	.	1	1	.	.	.	.
.	.	.	.	.	.	1	.	.	.
.	.	.	.	.	.	.	1	.	.
.	.	.	.	.	.	.	.	1	1

Рис. 7. Модуль памяти с резервом и таблица покрытия

которому соответствует покрытие $R=\{C_2,C_3,C_5,C_7,C_8\}$, как одно из трех возможных минимальных решений

$$R=C_2,C_3,C_5,C_7,C_8\vee C_2,C_3,C_5,C_8,R_9\vee C_2,C_5,C_8,R_4,R_9$$

для таблицы неисправностей.

Технологическая модель встроенного диагностирования и ремонта памяти (рис. 8) имеет четыре компонента:

1) тестирование модуля (Unit Under Test (UUT)) с использованием эталонной модели (Model Under Test (MUT)) для формирования вектора экспериментальной проверки m_a, размерность которого соответствует числу тестовых наборов;

2) поиск дефектов на основе анализа таблицы неисправностей A;

3) оптимизация покрытия дефектных ячеек ремонтными строками и столбцами на основе анализа таблицы A;

4) восстановление работоспособности памяти посредством замены адресов (Address Decoder (AD)) неисправных строк и столбцов, представленных вектором m_a, на адреса компонентов из ремонтного запаса (Spare Memory (SM)) [19].

МВП встроенного сервисного обслуживания работает в реальном масштабе времени и позволяет поддерживать в работоспособном состоянии, без вмешательства человека, цифровую систему на кристалле, что является целесообразным решением в случае применения технологий, связанных с дистанционной эксплуатацией изделия.

Предложенные МВП анализа ассоциативных таблиц, а также введенные критерии качества логических решений позволяют решать задачи

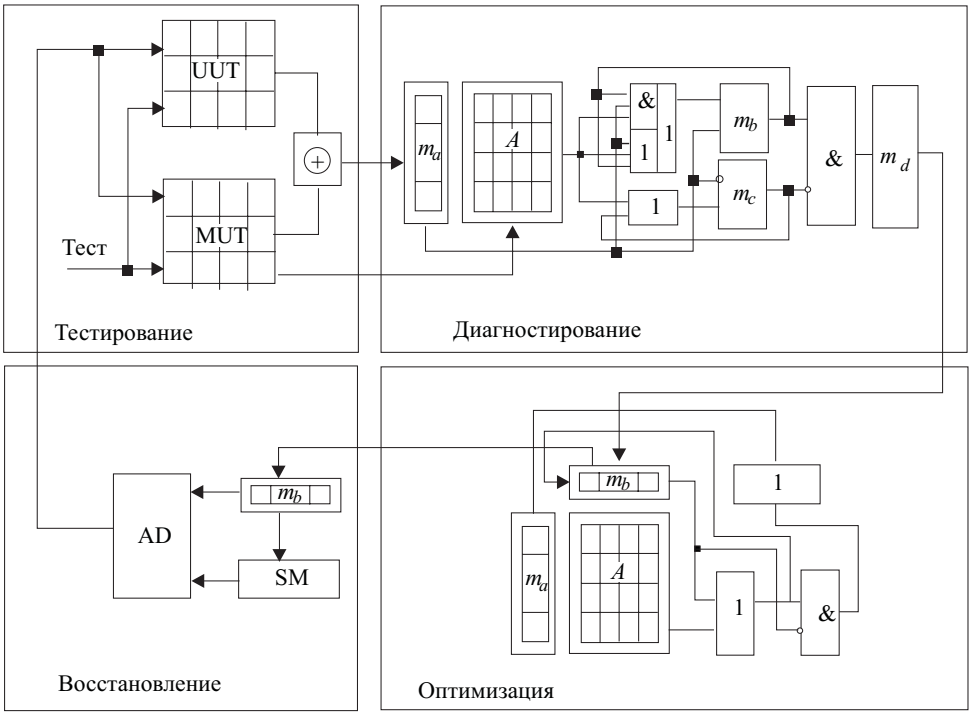


Рис. 8. Модель встроенного тестирования и восстановления памяти

$$E = F(L,T,H) = \min\left[\frac{1}{3}(L+T+H)\right],$$

$$Y = (1-P)^n,$$

$$L = 1 - Y^{(1-k)} = 1 - (1-P)^{n(1-k)},$$

$$T = \frac{(1-k)\times H^s}{H^s + H^a}, \quad H = \frac{H^a}{H^s + H^a}.$$

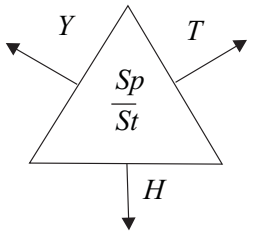


Рис. 9. Оценка эффективности МВП

квазиоптимального покрытия, диагностирования дефектов программных и (или) аппаратных блоков. Модель векторных вычислений стала основой для разработки специализированной мультипроцессорной архитектуры, ориентированной на поиск, распознавание и принятие решений об использовании структур ассоциативных таблиц.

Аналитическая оценка эффективности проектного решения, направленного на выполнение условий специализации *Sp* и стандартизации *St*

(рис. 9) определяется минимумом среднего значения следующих трех взаимно противоречивых относительных и безразмерных параметров: уровень ошибок проекта L, время верификации и (или) тестирования T, программно-аппаратная избыточность, определяемая механизмами асерций и (или) граничного сканирования H.

Параметр L, как дополнение к параметру Y, характеризующему выход годной продукции, зависит от тестопригодности проекта k, вероятности P существования неисправных компонентов и числа необнаруженных ошибок n. Время верификации определяется тестопригодностью проекта k, умноженной на структурную сложность аппаратно-программной функциональности, отнесенной к общей сложности проекта в строках кода или эквивалентных вентилях. Программно-аппаратная избыточность находится в функциональной зависимости от сложности ассерционного кода или механизма граничного сканирования, отнесенной к общей сложности проекта. При этом ассерционная, или сканирующая, избыточность должна обеспечивать заданную глубину диагностирования ошибок функциональности за время выхода изделия на рынок, определенное заказчиком.

Выводы. Существующие программные аналоги не предлагают ассоциативно-логических путей поиска, распознавания и принятия решений в дискретном информационном пространстве [21, 22, 25]. Практически все они используют универсальную систему команд современного дорогостоящего процессора с математическим сопроцессором. В то же время, аппаратные специализированные средства логического анализа, являющиеся их прототипами [4, 5], как правило, ориентированы на побитовую или невекторную обработку информации.

Предложенный новый подход векторно-логической обработки ассоциативных данных с полным исключением арифметических операций, влияющих на быстродействие и аппаратную сложность, может быть эффективно реализован на основе использования современной микроэлектронной аппаратуры в виде мультипроцессорной цифровой системы на кристалле.

Фактическая реализация подхода основана на следующем:

1. МВП анализа ассоциативных таблиц на основе векторных логических операций для поиска, распознавания образов, принятия и оценивания решений в векторном дискретном булевом пространстве, ориентированные на достижение высокого быстродействия параллельного векторного логического анализа информации и подсчета критериев качества решения.

2. Метод параллельного решения ассоциативно-логических задач с минимальным числом векторных логических операций и полным исключением арифметических команд, что обеспечивает высокое быстродействие, минимальную стоимость и незначительное энергопотребление вычислителя, реализованного на кристалле программируемой логики.

3. Новые векторно-логические МВП встроенного диагностирования цифровых систем на кристаллах, поиска квазиоптимального покрытия, использующие средства логического ассоциативного вычислителя, параллельные операции вычислительных процессов и подсчета критериев качества.

Практическая значимость полученных результатов подтверждена созданием вычислителя для диагностирования и восстановления работоспособности компонентов памяти в цифровой системе на кристалле. Дальнейшие исследования направлены на разработку прототипа логического ассоциативного процессора для решения актуальных задач поиска, распознавания и принятия решений с помощью векторного логического анализа.

Novel process-models for analyzing information in the tabular form based on using vector logical operations to solve the problems of search, diagnosis, pattern recognition and decision-making in the vector discrete Boolean space are proposed. The models are focused to realization of high-performance vector concurrent logical analysis of information that in the limit completely excludes the use of arithmetic operations.

1. *Zorian Y.* System-in-Package Test Strategies // Test Technology Educational Program. — 2009. — P. 6. — http://tab.computer.org/tttc/teg/ttep
2. *Smith L.* 3D Packaging Applications, Requirements, Infrastructure and Technologies // Fourth Annual International Wafer-Level Packaging Conference. — San Jose, California. September, 2007. — P. 4.
3. *The next* Step in Assembly and Packaging: System Level Integration in the package (SiP) Editors: W. Chen, W. R. Bottoms, K. Pressel, J. Wolf // SiP White Paper. International Technology Roadmap for Semiconductors. — 2007. — P. 17—23. — http://www.itrs.net/Links/2007ITRS/LinkedFiles/AP/AP_Paper.pdf
4. *А.с. 1439682*. Регистр сдвига. / Н. Я. Какурин, В. И. Хаханов, В. Г. Лобода, А. Н. Какурина. Опубл. 22.07.88, бюл. № 43.
5. *Бондаренко М. Ф. и др.* О мозгоподобных ЭВМ // Радиоэлектроника и информатика. — 2004. — № 2. — С. 89—105.
6. *Бондаренко М. Ф., Шабанов–Кушнаренко Ю. П.* Об алгебре предикатов // Бионика интеллекта. — 2004. — № 1. — С. 15—26.
7. *Бондаренко М. Ф., Шабанов–Кушнаренко Ю. П.* Теория интеллекта / Учебник. — Харьков: СМИТ, 2006. — 592 с.
8. *Бондаренко М. Ф., Шабанов–Кушнаренко Ю. П.* Модели языка // Бионика интеллекта. — 2004. — № 1. — С. 27—37.
9. *Акритас А.* Основы компьютерной алгебры с приложениями. Пер. с англ. — М. : Мир, 1994. — 544 с.
10. *Гилл Ф., Мюррей У., Райт М.* Практическая оптимизация — М. : Мир, 1985. — 509 с.
11. *Аттетков А. В., Галкин С. В., Зарубин В. С.* Методы оптимизации. — М. : Изд-во МГТУ им. Н. Э. Баумана, 2003. — 440 с.
12. *Дегтярев Ю. И.* Методы оптимизации / Учеб. пособие для вузов. — М. : Сов. радио, 1980. — 270 с.
13. *Bergeron J.* Writing Testbenches Using SystemVerilog. — NY : Springer Science and Business Media, Inc, 2006. — 414 p.
14. *Abramovici M., Breuer M. A., Friedman A. D.* Digital System Testing and Testable Design. — NY : Comp. Sc. Press, 1998. — 652 p.

15. *Densmore D., Passerone R., Sangiovanni–Vincentelli A.* A Platform–Based taxonomy for ESL Design // Design & Test of computers. — 2006. — P. 359—373.
16. *Хаханов В. И., Литвинова Е. И., Гузь О. А.* Проектирование и тестирование цифровых систем на кристаллах. — Харьков : Новое слово, 2009. — 484 с.
17. *Hahanov V. I., Gorbunov D. M., Miroshnichenko Y. V. et al.* SIGETEST — Test generation and fault simulation for digital design // Proc. of Conf. «Modern SoC Design Technology based on PLD». — Kharkov, 2003. — C. 50—53.
18. *Малышенко Ю. В. и др*. Автоматизация диагностирования электронных устройств. Под ред. В. П. Чипулиса. — М. : Энергоатомиздат, 1986. — 216 с.
19. *Хаханов В. И. и др.* Проектирование и верификация цифровых систем на кристаллах. — Харьков : Новое слово, 2010. — 528 с.
20. *Cohen A. A.* Addressing architecture for Brain-like Massively Parallel Computers. — Euromicro Symposium on Digital System Design (DSD'04). — 2004. — P. 594—597.
21. *Липаев В. В.* Программная инженерия. Методологические основы / Учебник. — М. : Теис, 2006. — 608 с.
22. *Трахтенгерц Э. А.* Компьютерные методы реализации экономических и информационных управленческих решений. — М. : СИНТЕГ. — 2009. — 396 с.
23. *Кузнецов О. П.* О моделировании быстрых интеллектуальных процессов обыденного мышления // Интеллектуальные системы. — 1997. — Т. 2, вып. 1—4. — С. 55—71.
24. *Кузнецов О. П.* Интеллектуализация поддержки управляющих решений и создание интеллектуальных систем // Проблемы управления.— 2009. — № 3.1. — С. 64—72.
25. *Васильев С. Н. и др*. Интеллектуальное управление динамическими системами. — М. : Физ.-мат. лит-ра, 2000. — 352 с.

Поступила 15.03.10;
после доработки 21.10.10

ХАХАНОВ Владимир Иванович, д-р техн. наук, декан факультета компьютерной инженерии и управления, профессор кафедры автоматизации проектирования вычислительной техники Харьковского национального университета радиоэлектроники. В 1978 г. окончил Харьковский ин-т радиоэлектроники. Область научных исследований — техническая диагностика цифровых систем, сетей и программных продуктов.

ЧУМАЧЕНКО Светлана Викторовна, д-р техн. наук, профессор кафедры автоматизации проектирования вычислительной техники Харьковского национального университета радиоэлектроники. В 1991 г. окончила Харьковский госуниверситет им. А. М. Горького. Область научных исследований — дискретная математика.

ЛИТВИНОВА Евгения Ивановна, д-р техн. наук, профессор кафедры автоматизации проектирования вычислительной техники Харьковского национального университета радиоэлектроники. В 1985 г. окончила Харьковский ин-т радиоэлектроники. Область научных исследований — проектирование и тестирование цифровых систем и сетей на кристаллах.

ГУЗЬ Олеся Алексеевна, канд. техн. наук, доцент, зав. кафедрой специализированных компьютерных систем Донецкой академии автомобильного транспорта. Область научных исследований — проектирование и диагностика цифровых систем и сетей на кристаллах.