

ЛОГИЧЕСКИЙ ВЕКТОРНЫЙ АНАЛИЗ АССОЦИАТИВНЫХ ТАБЛИЦ

Предлагается инфраструктура логического анализа ассоциативных таблиц, позволяющая выполнять обработку взаимодействия входного вектора с n -мерным алгебрологическим пространством, задаваемым с помощью упорядоченных и структурированных таблиц проблемно-ориентированных данных, которые представляют собой ассоциативные модели поведения логических объектов. Для оценки взаимодействия векторов в алгебрологическом пространстве разработан универсальный критерий качества, позволяющий находить и оценивать квазиоптимальное решение в задачах ассоциативно-логического поиска информации. Приводятся примеры использования инфраструктуры и алгебрологических процедур для решения традиционных задач логического анализа, подтверждающие эффективность и практическую направленность алгебраических моделей.

Цель – разработка элементов алгебры логического анализа ассоциативных таблиц для реализации инфраструктуры обработки больших объемов информации при решении задач поиска, принятия решений и диагностирования технических состояний.

Задачи исследования: 1. Мотивация и актуальность процедур анализа информационных массивов, сформатированных в графы ассоциативных таблиц. 2. Разработка алгебры логического анализа ассоциативных таблиц как эффективного математического аппарата, ориентированного на параллельно-конвейерную обработку векторных и табличных массивов. 3. Создание метрики для оценивания взаимодействия векторов и таблиц в процессе поиска оптимального решения в дискретном булевом пространстве. 4. Решение практических задач нахождения квазиоптимального покрытия и диагностирования неисправностей или ошибок в программных продуктах.

Источники исследования: 1. Аппаратные средства решения информационно-логических задач [1-4]. 2. Алгебраические системы для описания и анализа ассоциативно-логических структур данных [5-8]. 3. Критерии оценивания вариантов поиска решений в дискретном булевом пространстве [9-12]. 4. Методы решения задач покрытия и диагностирования на основе анализа таблиц [13-16].

1. Актуальность технологий векторного параллельного анализа

Современный рынок электронных и информационных технологий нуждается в новых носителях больших объемов информации, эффективных структурах данных для хранения и поиска требуемой информации, новых интеллектуальных и компьютерных средствах, точно и быстро обеспечивающих любые запросы пользователя. Современные компьютеры ориентированы в своей основе на арифметические вычисления, поэтому не всегда являются эффективным инструментом для решения логических, ассоциативных и поисковых задач в существующем океане информации. Поэтому актуальной представляется разработка новых специализированных технологий, ориентированных на использование ограниченной системы логических команд для анализа в реальном масштабе времени больших информационных массивов, организованных в эффективные структуры данных. Если исключить из рассмотрения арифметические операции, которые не свойственны человеческому мозгу, то все ресурсы компьютера можно переориентировать на повышение мощности и производительности вычислителя для решения чрезвычайно большого класса проблем логического, ассоциативного анализа, распознавания образов, диагностирования и поиска информации.

Создание ассоциативно-логического «движка» должно быть ориентировано только на параллельные вычисления, связанные с обработкой двоичных и многозначных векторов, таблиц, матриц на основе использования мультипроцессора, ассоциативной памяти, регистров. Для оценивания полученных решений необходимо разработать арифметически независимый критерий оценки качества в векторном логическом пространстве, определяемый

соотношением нулевых и единичных разрядов. Таким образом, лучшим решением при разработке упомянутого «движка» будет полное исключение всех арифметических операций и команд на их основе, что позволит на несколько порядков повысить быстродействие выполнения векторных логических операций над информационными массивами больших объемов. Теория должна быть ориентирована только на логические операции, оперировать не переменными, а векторами и матрицами (таблицами), максимально распараллеливать все вычислительные процессы, обеспечивать масштабируемость как размерности операндов, так и исполняемых программных приложений для эффективного достижения цели с требуемым качеством удовлетворения запроса.

Сущность данного исследования – создание проблемно ориентированной алгебры логического анализа ассоциативных матриц на основе минимального числа логических операций, ориентированных на параллельно-конвейерную обработку табличных и векторных форм данных для существенного повышения быстродействия анализа больших объемов информации при решении задач поиска, принятия решений, распознавания образов и диагностирования технических состояний объектов или процессов.

2. Инфраструктура логического анализа ассоциативных таблиц

Аналитическая модель описания и решения логических уравнений информационного анализа представлена системой выражений:

$$\begin{aligned}
 P(m, A) &= Q(m, A) = [0, 0; 1, 0]; \\
 Q(m, A) &= \frac{1}{3}[d(m, A) + \mu(m \in A) + \mu(A \in m)]; \\
 P(m, A) &= 1 \leftarrow (m = A); \\
 P(m, A) &= \max_i Q_i(m, A_i); \\
 Q(m, A_i) &= \{Q_1, Q_2, \dots, Q_i, \dots, Q_n\}; \\
 P(m, A) &= m \Delta (A_1, A_2, \dots, A_i, \dots, A_t); \\
 \Delta &= \{\neg, \wedge, \vee\}; \\
 A &= (A_1, A_2, \dots, A_i, \dots, A_t); \\
 A_i &= (A_{i1}, A_{i2}, \dots, A_{ij}, \dots, A_{is}); \\
 A_{ij} &= (A_{ij1}, A_{ij2}, \dots, A_{ijr}, \dots, A_{ijq}); \\
 m &= (m_1, m_2, \dots, m_r, \dots, m_q); \\
 P(m, A) &= m \wedge \{A_1, A_2, \dots, A_i, \dots, A_t\} = p; \quad p = \{p_1, p_2, \dots, p_r, \dots, p_n\}; \\
 m \wedge (A_1 \vee A_2 \vee \dots \vee A_i \vee \dots \vee A_t) &= \max_i Q_i(m, A_i); \\
 p_r = m \bigwedge_{i=1, t}^{j=1, s} A_{ij} &\leftarrow \begin{cases} \forall i \exists j (m \wedge A_{ij} \neq \emptyset); \\ \exists i \forall j (m \wedge A_{ij} = \emptyset) \wedge d(m, A_{ij}) = \max; \quad p_r \in p; \end{cases} \\
 m \wedge A_{ij} = m_r \bigwedge_{r=1}^q A_{ijr} &= \begin{cases} m \leftarrow m_r \bigwedge_{r=1}^q A_{ijr} = m_r; \\ A_{ij} \leftarrow m_r \bigwedge_{r=1}^q A_{ijr} = A_{ijr}; \\ m \wedge A_{ij} \leftarrow m_r \bigwedge_{r=1}^q A_{ijr} = m_r \vee A_{ijr}; \\ \emptyset \leftarrow \exists (m_r \bigwedge_{r=1}^q A_{ijr} = \emptyset); \end{cases} \quad (1) \\
 P(m, A) = p_r \leftarrow Q_i(m, p_r) &= \max, \quad r = \overline{1, n}, p_r \in p.
 \end{aligned}$$

Здесь определены следующие предикаты или высказывания. Выражение $P(m, A) = Q(m, A)$ с помощью операций объединения, пересечения и дополнения задает аналитическую модель вычислительного процесса в виде предиката, определенного на единичном интервале $Q(m, A) = [0, 0; 1, 0]$ интегральным критерием принадлежности входного вектора множеству дискретных булевых состояний. Упорядоченная совокупность $A = (A_1, A_2, \dots, A_i, \dots, A_t)$ взаимодействующих ассоциативных таблиц объединяется на системном уровне обобщенным графом целевой функциональности. Предикат $A_i = (A_{i1}, A_{i2}, \dots, A_{ij}, \dots, A_{is}) = 1$ представлен совокупностью векторов, формирующих ассоциативную таблицу. Вектор $A_{ij} = (A_{ij1}, A_{ij2}, \dots, A_{ijr}, \dots, A_{ijtsq})$ задает ассоциативное решение как совокупность состояний переменных, где каждая координата определена в троичном алфавите $A_{ijr} \in \{0, 1, x\} = \beta$. Взаимодействие $P(m, A)$ входного вектора запроса $m = (m_1, m_2, \dots, m_r, \dots, m_q)$ с графом ассоциативных таблиц $A = (A_1, A_2, \dots, A_i, \dots, A_t)$ формирует множество решений $p = \{p_1, p_2, \dots, p_r, \dots, p_n\}$, каждое из которых имеет интегральную оценку качества $Q(m, A) = \{Q_1, Q_2, \dots, Q_i, \dots, Q_n\}$. Выбор лучшего решения осуществляется путем вычисления функционала:

$$P(m, A) = p_r \leftarrow Q_i(m, p_r) = \max, r = \overline{1, n}, p_r \in p.$$

3. Элементы алгебры векторного анализа ассоциативных матриц

Матрица (таблица), размерностью $n \times m$, может быть представлена векторами строк или столбцов $[A_i] \vee [A_j]$, где каждая координата матрицы или вектора определена в алфавите $\{0, 1\}$:

$$A = [A_{ij}] = \{[A_i], [A_j]\}, [A_i] = (A_{i1}, A_{i2}, \dots, A_{in}), [A_j] = (A_{j1}, A_{j2}, \dots, A_{jt}), A_{ij} = \{0, 1\}, A_i = (A_{i1}, A_{i2}, \dots, A_{ij}, \dots, A_{in}), A_j = (A_{j1}, A_{j2}, \dots, A_{ji}, \dots, A_{jt}). \quad (2)$$

Матрица взаимодействует с одним из входных векторов, ориентированных на обработку строк или столбцов $m^a \vee m^b$:

$$m = m^a \vee m^b, m^a = (m_1, m_2, \dots, m_i, \dots, m_n), m^b = (m_1, m_2, \dots, m_j, \dots, m_t). \quad (3)$$

По существу матрица представляет собой любое ассоциативное отношение (таблицу истинности, таблицу предикатов), заданное в n - (m -) мерном векторном двоичном пространстве, где каждый вектор есть точка, принадлежащая данному пространству (т.е. элемент пространства). Взаимодействие m -вектора с матрицей A преследует цель определить отношение принадлежности каждого вектора матрицы или точки пространства к m -вектору или выполнить разделение пространства точек на два подмножества. Для решения данного класса задач вводятся предикаты, инвариантные по отношению к обработке строк и столбцов матрицы, основанные на использовании логических операций (И, ИЛИ, НЕ).

1) Дизъюнкция конъюнктивного взаимодействия между m -вектором и A -матрицей:

$$P_1(m, A) = m \wedge A = m \wedge [A_1 \vee A_2 \vee \dots \vee A_{i(j)} \vee \dots \vee A_{n(t)}] =$$

$$2) = \begin{cases} p^a = p^a \vee m^a \bigwedge_{i=1}^n A_i \leftarrow (m^a \bigwedge_{i=1}^n A_i \neq \emptyset); \\ p^b = p^b \vee m^b \bigwedge_{j=1}^t A_j \leftarrow (m^b \bigwedge_{j=1}^t A_j \neq \emptyset). \end{cases} \quad (4)$$

3) Логическое умножение конъюнкции единичных по m векторов матрицы A на отрицание дизъюнкции нулевых по m векторов A :

$$P_2(m, A) = \left(\bigwedge_{\forall m_{i(j)}=1} A_{i(j)} \right) \wedge \left(\overline{\bigvee_{\forall m_{i(j)}=0} A_{i(j)}} \right).$$

4) Логическое умножение дизъюнкции единичных по m векторов матрицы A на отрицание дизъюнкции нулевых по m векторов A :

$$P_3(m, A) = \left(\bigvee_{\forall m_{i(j)}=1} A_{i(j)} \right) \wedge \left(\bigvee_{\forall m_{i(j)}=0} A_{i(j)} \right). \quad (5)$$

Для всех трех операций качество взаимодействия между m -вектором и A -матрицей оценивается с помощью теоретико-множественной метрики:

$$\begin{aligned} Q &= \frac{1}{3} [d(m, A_{i(j)}) + \mu(m \in A_{i(j)}) + \mu(A_{i(j)} \in m)], \\ d(m, A_{i(j)}) &= \frac{1}{n} \{n - \text{card}[m \bigcap_{i(j)=1}^{n(t)} A_{i(j)} = \emptyset]\}; \\ \mu(m \in A_{i(j)}) &= 2^{\text{card}[m \cap A_{i(j)}] - \text{card}[A_{i(j)}]} \leftarrow \text{card}[m \cap A_{i(j)}] = \\ &= \text{card}[m \bigcap_{i(j)=1}^{n(t)} A_{i(j)} = x] \& \text{card}[A_{i(j)}] = \text{card}[\bigcup_{i(j)=1}^{n(t)} A_{i(j)} = x] \& [m \bigcap_{i(j)=1}^{n(t)} A_{i(j)} \neq \emptyset]; \\ \mu(A_{i(j)} \in m) &= 2^{\text{card}[m \cap A_{i(j)}] - \text{card}[m]} \leftarrow \text{card}[m \cap A_{i(j)}] = \\ &= \text{card}[m \bigcap_{i(j)=1}^{n(t)} A_{i(j)} = x] \& \text{card}[m] = \text{card}[\bigcup_{i(j)=1}^{n(t)} m = x] \& [m \bigcap_{i(j)=1}^{n(t)} A_{i(j)} \neq \emptyset]. \end{aligned} \quad (6)$$

Данные формулы определения качества взаимодействия m -вектора и A -матрицы изоморфны следующим алгебрологическим выражениям:

$$\begin{aligned} Q &= \frac{1}{3} [d(m, A_{i(j)}) + \mu(m \in A_{i(j)}) + \mu(A_{i(j)} \in m)], \\ d(m, A_{i(j)}) &= \frac{1}{n} \{n - \text{card}[m \bigwedge_{i(j)=1}^{n(t)} A_{i(j)} = 0]\}; \\ \mu(m \in A_{i(j)}) &= 2^{\text{card}[m \wedge A_{i(j)}] - \text{card}[A_{i(j)}]} \leftarrow \text{card}[m \wedge A_{i(j)}] = \\ &= \text{card}[m \bigwedge_{i(j)=1}^{n(t)} A_{i(j)} = x] \& \text{card}[A_{i(j)}] = \text{card}[\bigvee_{i(j)=1}^{n(t)} A_{i(j)} = x] \& [m \bigwedge_{i(j)=1}^{n(t)} A_{i(j)} \neq 0]; \\ \mu(A_{i(j)} \in m) &= 2^{\text{card}[m \wedge A_{i(j)}] - \text{card}[m]} \leftarrow \text{card}[m \wedge A_{i(j)}] = \\ &= \text{card}[m \bigwedge_{i(j)=1}^{n(t)} A_{i(j)} = x] \& \text{card}[m] = \text{card}[\bigvee_{i(j)=1}^{n(t)} m = x] \& [m \bigwedge_{i(j)=1}^{n(t)} A_{i(j)} \neq 0]. \end{aligned} \quad (7)$$

Здесь кодирование теоретико-множественных символов четырехзначного алфавита для выполнения операций в алгебре логики определяется в следующем виде:

$$T(\alpha \rightarrow \beta) = \begin{array}{c|cccc} \alpha & 0 & 1 & x & \emptyset \\ \beta & 01 & 10 & 11 & 00 \end{array}. \quad (8)$$

В таком алфавите все теоретико-множественные операции, согласно изоморфизму, имеют аналоги в булевой алгебре, что позволяет параллельно выполнять алгебрологические регистровые операции над векторами. Если векторы $A_{i(j)}$, m определены в двоичном алфавите, то формулы вычисления качества взаимодействия m -вектора и A -матрицы существенно упрощаются:

$$\begin{aligned}
Q &= \frac{1}{3} [d(m, A_{i(j)}) + \mu(m \in A_{i(j)}) + \mu(A_{i(j)} \in m)], \\
d(m, A_{i(j)}) &= \frac{1}{n} \{n - \text{card}[\bigwedge_{i(j)=1}^{n(t)} A_{i(j)} = 0]\}; \\
\mu(m \in A_{i(j)}) &= 2^{\text{card}[m \wedge A_{i(j)}] - \text{card}[A_{i(j)}]} \leftarrow \text{card}[m \wedge A_{i(j)}] = \\
&= \text{card}[\bigwedge_{i(j)=1}^{n(t)} A_{i(j)} = 1] \& \text{card}[A_{i(j)}] = \text{card}[\bigvee_{i(j)=1}^{n(t)} A_{i(j)} = 1]; \\
\mu(A_{i(j)} \in m) &= 2^{\text{card}[m \wedge A_{i(j)}] - \text{card}[m]} \leftarrow \text{card}[m \wedge A_{i(j)}] = \\
&= \text{card}[\bigwedge_{i(j)=1}^{n(t)} A_{i(j)} = 1] \& \text{card}[m] = \text{card}[\bigvee_{i(j)=1}^{n(t)} m = 1].
\end{aligned} \tag{9}$$

Фактически функция принадлежности определяется числом единиц, полученных после логического умножения векторов $A_{i(j)} \wedge m$, отнесенных к количеству единиц в векторах $A_{i(j)}$, m . Здесь следует понимать существенное отличие взаимодействия уже не пространств, а точек в виде вектор-маски m и двоичных векторов матрицы $A = [A_{ij}]$. Тем не менее, два упомянутых компонента успешно могут взаимодействовать в векторном алгебрологическом пространстве с помощью операций И, ИЛИ, НЕ. При этом следует учитывать, что нулевые векторы не могут взаимодействовать между собой, поскольку они соответствуют пустому множеству по определению. Следовательно, векторы матрицы $A = [A_{ij}]$ и маска m не могут быть определены только нулевыми значениями по всем координатам. Кодовое расстояние между двумя алгебрологическими векторами определяется числом нулевых значений при их конъюнкции минус количество единичных координат в результирующем векторе.

Здесь и далее матрица $A = [A_{ij}]$ будет записываться двумя способами – по строкам или столбцам:

$$\begin{aligned}
A &= [A] \vee [B] = [A_i] \vee [B_j], \\
A &= (A_1, A_2, \dots, A_i, \dots, A_n); \\
B &= (B_1, B_2, \dots, B_j, \dots, B_m).
\end{aligned} \tag{10}$$

Две формы записи служат для сокращения сложности представления формул, задающих процесс обработки матрицы, а также для улучшения понимания аналитически записанных вычислительных процессов. Соответственно входной вектор-маска имеет идентификаторы $m = m^a \vee m^b$ в зависимости от способа анализа матрицы, по строкам или столбцам. Четыре предикатные векторные операции (and, or, not, xor – раньше не упоминалась и не задавалась) позволяют решать многие, интересные для практики, задачи. Некоторые из них представлены ниже.

4. Логический анализ таблицы для поиска покрытия

Нахождение квазиоптимального покрытия единичными значениями вектор-строк всех вектор-столбцов таблицы. Технологическая инновация – параллельное выполнение операций над векторами матрицы или таблицы. Научная новизна – определение подмножества максимально непересекающихся или минимально пересекающихся строк. Стратегия процедуры. Выбор ядра – каждая строка, покрывающая некоторый столбец безальтернативно – одна единица в столбце. Для этого вектор-столбец m^b последовательно маскирует по одной координате для нахождения ядра в виде вектор-строк, которые покрывают безаль-

тернативно некоторый столбец. Определение дополнения к ядру – строки с максимальным числом единиц, имеющие максимально пустое пересечение с уже найденными векторами покрытия. Вербальному описанию модели процесса поиска покрытия соответствует следующая аналитическая форма поиска ядра и дополнения к нему:

$$P = P \vee_{i=1}^n [A_i \leftarrow (Q_j (m^b \bigwedge_{j=1}^t B_j) = 1) \leftarrow (m^b = 0) \vee (m_j^b = 1)];$$

$$P = P \vee_{i=1}^n [A_i \leftarrow \max[Q_i ((m^a = 1) \wedge A_i)) \times (1 - Q_i (P \wedge A_i))]]. \quad (11)$$

где $\times$ – арифметическое умножение критериев качества.

Здесь вектор m играет неоценимую роль маскирования строк, столбцов или отдельных переменных матрицы для создания идеальных условий вычислительного процесса в целях достижения результата. Без него матрица не несет информации. Для решения задачи покрытия необходимо последовательно задать для каждого столбца вектор-маску $(m^b = 0) \vee (m_j^b = 1)$, определенную всеми нулями, кроме одной координаты j , имеющей единичный сигнал; запись $m^a = 1$ означает, что строчный вектор-маска равен 1 по всем координатам. Произведение критериев качества $Q_i ((m^a = 1) \wedge A_i)) \times (1 - Q_i (P \wedge A_i))$ предпочтительнее суммы, поскольку оно дает более взвешенную оценку относительно покрывающей способности вектора и кодового расстояния с уже имеющимся решением при выборе очередной строки – кандидата в покрытие. Вычислительная сложность процедуры

поиска квазиоптимального покрытия равна $Z = n + \frac{1}{2}n^2$, где n – число строк в матрице, первое слагаемое – определение ядра покрытия, второе – поиск квазиоптимального дополнения к нему. Для примера таблицы, размерностью 5×7 , имеющей вид:

$$A = [A_i] \vee [B_j] =$$

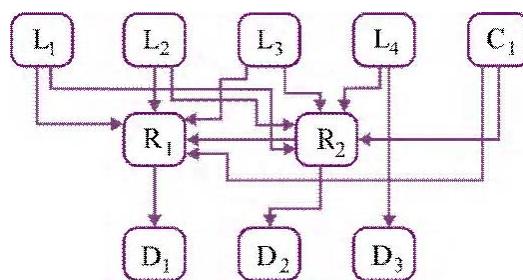
$A_i \setminus B_j$	B_1	B_2	B_3	B_4	B_5	B_6	B_7
A_1	0	0	0	0	1	0	1
A_2	0	0	1	1	0	1	0
A_3	0	1	0	1	1	0	0
A_4	0	1	1	0	0	1	0
A_5	1	0	0	1	0	0	0

(12)

модель поиска квазиоптимального покрытия на основе использования описанной выше процедуры дает минимальное покрытие: $P = \{A_1, A_4, A_5\}$.

5. Логический метод диагностирования дефектов по таблице неисправностей

Рассматривается на примере транзакционного графа одного из программных модулей Row_buffer, используемых при создании IP-core вейвлет-преобразования для стандарта JPEG 2000 (рисунок). Вершины графа представлены входными шинами и переменными, регистрами и выходными шинами. Дуги между ними означают существование транзакций между вершинами при выполнении операторов HDL-кода.



Row_buffer транзакционный граф

Для данного Row_buffer графа построена таблица, задающая поведение неисправных блоков на тесте путем использования системы моделирования и генерации тестов SIGETEST:

Test	B ₁	B ₂	B ₃	B ₄	B ₅	B ₆	B ₇	B ₈	B ₉	B ₁₀	m ₁ ^b	m ₂ ^b
	(L ₁)	(L ₂)	(L ₃)	(L ₄)	(C ₁)	(R ₁)	(R ₂)	(D ₁)	(D ₂)	(D ₃)		
A ₁ (t ₁)	1	0	0	0	0	1	0	1	0	1	0	1
A ₂ (t ₂)	0	1	0	0	0	0	1	0	1	0	1	1
A ₃ (t ₃)	0	0	1	0	0	1	1	1	0	1	0	1
A ₄ (t ₄)	0	0	0	1	0	0	1	0	1	0	1	0
A ₅ (t ₅)	0	0	0	0	1	1	1	1	0	1	0	0
A ₆ (t ₆)	1	0	0	0	0	0	1	0	1	0	1	1
A ₇ (t ₇)	0	1	0	0	0	1	0	1	0	0	0	1
A ₈ (t ₈)	0	0	1	0	0	1	0	1	0	0	0	1
A ₉ (t ₉)	0	0	1	0	0	0	1	0	1	0	1	0
A ₁₀ (t ₁₀)	0	0	0	1	0	1	1	1	0	0	0	0
A ₁₁ (t ₁₁)	0	0	0	1	0	0	0	0	0	1	0	0
A ₁₂ (t ₁₂)	0	0	0	0	1	0	1	0	1	0	1	0
A ₁₃ (t ₁₃)	0	0	0	0	1	1	1	1	0	0	0	0

5.1. Логический метод анализа столбцов таблицы неисправностей

Основан на применении операции логического умножения или конъюнкции вектора экспериментальной проверки, формально рассматриваемого в качестве входного вектор-столбца или маски m , на столбцы таблицы неисправностей $m^b \wedge (B_1 \vee B_2 \vee \dots \vee B_j \vee \dots \vee B_m)$ и подсчете качества взаимодействия векторов $Q_j(m^b \wedge B_j)$ в целях выбора лучшего из них при наличии максимальной оценки. При этом столбец $B_j \in A$ фактически идентифицирует метрику поведения неисправности или дефектного блока на тестовых наборах. Предикатная запись процесса получения решения в виде совокупности ошибок, присутствующих в HDL-коде, представлена в следующем виде:

$$P^s = P^s \bigvee_{i=1}^n [B_j \leftarrow \max(Q_j(m^b \bigwedge_{j=1}^t B_j))];$$

$$P^m = P^m \bigvee_{i=1}^n [B_j \leftarrow (m^b \wedge B_j = B_j)] \leftarrow Q[m^b \wedge (\bigvee_{j=1}^t B_j \in P^m)] = 1 \vee \max; \quad (13)$$

Здесь вектор экспериментальной проверки

$$m^b = f(T, F) \oplus f^*(T, F, L) \quad (14)$$

есть результат проведения тестового эксперимента – сравнение функционалов (состояний выходов) эталонного $f(T, F)$ и реального $f^*(T, F, L)$ устройства с дефектами L на тестовых наборах T .

Достоинство метода – выбор всегда лучшего решения из всех возможных как для одиночных, так и для кратных дефектов. В последнем случае, если одиночный дефект не идентифицируется оценкой, равной 1, выполняется дизъюнкция таких вектор-строк (главное отличие метода от существующих технологий), которые формируют оценку качества,

равную 1 или максимально близкую к единице $Q[m^b \wedge (\bigvee_{j=1}^t B_j \in P^t)] = 1 \vee \max$. По суще-

ству, в список кратных дефектов включаются такие одиночные неисправности, которые при логическом умножении на вектор экспериментальной проверки дают результат в виде соответствующего вектор-столбца. Дизъюнкция всех столбцов, составляющих решение,

$$\bigvee_{j=1}^t (B_j \in P^t) = m^b.$$

С использованием таблицы и процедуры диагностирования (13) определяются дефектные компоненты программного кода модуля Row_buffer методом логического умножения вектор-столбцов таблицы истинности на вектор экспериментальной проверки. Здесь векторы m_1^b, m_2^b формируют результаты диагностического эксперимента, выполненные по технологии (14). Результат диагностирования одиночных и кратных дефектов имеет следующий вид:

$$\begin{aligned} P^s(m_1^b) &= m_1^b \wedge (L_1 \vee L_2 \vee L_3 \vee L_4 \vee C_1 \vee R_1 \vee R_2 \vee D_1 \vee D_2 \vee D_3) = D_2; \\ P^m(m_2^b) &= m_2^b \wedge (L_1 \vee L_2 \vee L_3 \vee L_4 \vee C_1 \vee R_1 \vee R_2 \vee D_1 \vee D_2) = L_1 \vee L_2; \\ Q(m_1^b, D_2) &= 1; \\ Q[m_2^b, (L_1 \vee L_2)] &= \frac{1}{3}(\frac{4}{13} + \frac{1}{4} + 1) = 0,52. \end{aligned} \tag{15}$$

В первом случае диагноз определен в виде одного дефектного блока D_2 , присутствующего в транзакционном графе, качество решения равно 1. Во втором случае процедура диагностирования выявила наличие двух дефектных модулей $L_1 \vee L_2$, которые не смогли сформировать идеальную оценку качества. Тем не менее, решение является лучшим среди всех возможных, которое максимально приближено к вектору экспериментальной проверки по критерию принадлежности $Q[m_2^b, (L_1 \vee L_2)]$.

Вычислительная сложность метода анализа столбцов определяется следующей зависимостью: $Z^c = 3n^2 + n^2 = 4n^2$; $Z^r = 3n + n = 4n$. Здесь первая оценка учитывает выполнение координатных операций над матрицей размерностью $n \times n$. Вторая оценка определяет вычислительную сложность регистровых параллельных операций для подсчета критериев качества и обработки матрицы соответственно.

5.2. Логический метод анализа строк таблицы неисправностей

Стратегия определения ошибок программного кода по таблице неисправностей связана с анализом ее строк, состоящим из двух процедур: 1) вычисление логического произведения конъюнкции строк, отмеченных единичными значениями вектора $A_i(m_i^b = 1)$, на отрицание дизъюнкции нулевых строк $A_i(m_i^b = 0)$ для одиночных дефектных блоков; 2) вычисление логического произведения дизъюнкции единичных строк на отрицание дизъюнкции нулевых строк для кратных дефектных блоков:

$$\begin{aligned} P^s &= (\bigwedge_{\forall m_i^b=1} A_i) \wedge (\overline{\bigvee_{\forall m_i^b=0} A_i}); \\ P^m &= (\bigvee_{\forall m_i^b=1} A_i) \wedge (\overline{\bigvee_{\forall m_i^b=0} A_i}); \end{aligned} \tag{16}$$

Формулы интересны тем, что они не привязаны к критериям качества диагностирования, а оперируют лишь двумя компонентами, таблицей неисправностей и вектором экспериментальной проверки. Выполнение процедуры диагностирования по формулам (16) для вектора экспериментальной проверки $m^b = (0101010010010)$, заданного в последней таблице неисправностей, дает результат: $P^s(m_1^b, A) = D_2$, который не хуже, чем полученный ранее методом анализа столбцов. Для вектора экспериментальной проверки $m^b = (1110011100000)$, результат диагностирования имеет вид: $P^m(m_2^b, A) = L_1 \vee L_2$. Вычислительная сложность метода анализа строк определяется следующей зависимостью: $Z^c = n^2$; $Z^r = n$. Первая оценка предназначена для подсчета числа координатных операций, вторая – определяет вычислительную сложность процесса обработки на основе регистровых параллельных операций.

6. Векторно-логический критерий качества решения

Цель – использовать только логические операции и исключить арифметические вычисления из процедуры формирования критерия для существенного повышения быстродействия логического анализа информационных массивов.

Идея – качество решения логического анализа оценивается мощностью единиц в векторе качества и определяется путем взаимодействия входного вектора с таблицами данных на основе использования векторных логических операций.

Вектор качества формирует критерий принадлежности, определяемый следующими выражениями:

$$\begin{aligned}
 Q &= d[m, A_{i(j)}] + \mu[m \in A_{i(j)}] + \mu[A_{i(j)} \in m], \\
 d(m, A_{i(j)}) &= \text{card} \left[m \bigoplus_{i(j)=1}^{n(i)} A_{i(j)} = 1 \right]; \\
 \mu(m \in A_{i(j)}) &= \text{card}[A_{i(j)} = 1] - \text{card} \left[m \bigwedge_{i(j)=1}^{n(i)} A_{i(j)} = 1 \right]; \\
 \mu(A_{i(j)} \in m) &= \text{card}[m = 1] - \text{card} \left[m \bigwedge_{i(j)=1}^{n(i)} A_{i(j)} = 1 \right].
 \end{aligned} \tag{17}$$

Первый компонент, составляющий критерий, формирует степень несовпадения n -мерных векторов – кодовое расстояние, путем выполнения операции хог, второй и третий – определяют степень непринадлежности результата конъюнкции к числу единиц каждого из двух взаимодействующих векторов. Понятия принадлежности и непринадлежности являются взаимодополняющими, но в данном случае технологичнее высчитывать именно непринадлежность. Таким образом, идеальный критерий качества равен нулю, когда два вектора равны между собой. Оценка качества взаимодействия двух двоичных векторов убывает по мере роста критерия от 0 к 1.

Чтобы окончательно уйти от арифметических операций при подсчете критерия качества, необходимо выражения (17) преобразовать к следующему виду:

$$\begin{aligned}
 Q &= d(m, A) \vee \mu(m \in A) \vee \mu(A \in m), \\
 d(m, A) &= m \oplus A; \\
 \mu(m \in A) &= A \wedge \overline{m \wedge A}; \\
 \mu(A \in m) &= m \wedge \overline{m \wedge A}.
 \end{aligned} \tag{18}$$

Здесь критерии представлены уже не числами, а векторами, которые оценивают взаимодействие между компонентами m, A . При этом число нулей в трех оценках есть хорошо, а единицы ухудшают качество взаимодействия. Оптимизация решения логической задачи направлена на минимизацию числа единиц и максимизацию количества нулевых координат в векторах критерия качества. Для сравнения двух оценок необходимо определять мощность единиц в каждом векторе без выполнения операций суммирования, это можно сделать, например, с помощью регистра [4] уплотнения единиц и их сдвига влево. После процедуры сжатия номер последнего или правого единичного бита уплотненной серии единиц формирует индекс качества взаимодействия векторов.

Для двух векторов $m = (110011001100)$; $A = (000011110101)$ определение качества их взаимодействия в соответствии с моделью (18) представлено 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 не только оценка качества взаимодействия векторов, равная $Q(m, A) = (6,12)$, но, что самое главное, единичные координаты строки $Q = d(m, A) \vee \mu(m \in A) \vee \mu(A \in m)$ идентифицируют все те места или позиции, по которым существует некачественное взаимодействие векторов.

Другой пример иллюстрирует формирование максимального критерия качества для двоичных векторов $m = (110000110011)$; $A = (110000110011)$, имеющих совпадение по всем координатам:

m	1 1 1 1 . . 1 1
A	1 1 1 1 . . 1 1
$m \wedge A$	1 1 1 1 . . 1 1
$\overline{m \wedge A}$	. . 1 1 1 1 . . 1 1 . .
$d(m, A) = m \oplus A$	
$\mu(A \in m) = m \wedge (\overline{m \wedge A})$	
$\mu(m \in A) = A \wedge (\overline{m \wedge A})$	
$Q = d(m, A) \vee \mu(m \in A) \vee \mu(A \in m)$	
$Q = (0,12)$	

Здесь критерий качества, равный нулю во всех 12 разрядах $Q = (0,12)$, является максимальным или самым лучшим для взаимодействующих векторов $m = (110000110011)$; $A = (110000110011)$, поскольку он определен минимальным числом единиц на двенадцати координатах вектора. Для сравнения двух решений, полученных в результате логического анализа, следует использовать сжатые векторы качества Q , над которыми необходимо выполнить процедуру, включающую следующие векторные операции:

$$Q(m, A) = \begin{cases} Q_1(m, A) \leftarrow Q_1(m, A) \oplus Q_1(m, A) \wedge Q_2(m, A) = 0; \\ Q_2(m, A) \leftarrow Q_1(m, A) \oplus Q_1(m, A) \wedge Q_2(m, A) \neq 0. \end{cases} \quad (19)$$

Для двоичных векторов, представляющих собой критерии качества, выполнена процедура выбора лучшего из них на основе выражения, представленного в (19):

$Q_1(m, A) = (6,12)$	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

Предложенная алгебра и разработанные на ее основе аналитические модели и методы анализа таблиц на основе введенных критериев качества позволяют решать задачи поиска квазиоптимального покрытия, диагностирования одиночных и кратных дефектов программных и/или аппаратных блоков. Алгебра анализа матриц также может быть использована для решения следующих задач технической диагностики: 1. Моделирование исправного поведения цифрового устройства, заданного таблицей истинности. 2. Определение качества теста по таблице неисправностей. 3. Минимизация теста. 4. Поиск тестов, распознающих заданный дефект, на основе анализа таблицы неисправностей. 5. Идентификация кратного дефекта. 6. Поиск и исправление ошибок при наборе текста.

7. Выводы

1. Научная новизна представлена алгеброй логического анализа векторных и табличных форм задания информации для решения задач поиска, диагностирования, распознавания образов и принятия решений в векторном дискретном булевом пространстве. Представлены быстродействующие модели и методы параллельного векторного логического анализа информации, в пределе полностью исключающие использование арифметических операций, в том числе и для подсчета критерия качества решения, где применяются только логические операции. В качестве примеров описаны новые методы для решения задач диагностирования и нахождения квазиоптимального покрытия, использующие векторные операции, ориентированные на распараллеливание вычислительных процессов.

2. Практическая значимость заключается в ориентации алгебры ассоциативных таблиц и методов их анализа на создание логического мультипроцессора с ограниченной системой команд, ориентированной на высокое быстродействие параллельной обработки больших массивов информации, представленных графовыми структурами ассоциативных матриц.

3. Дальнейшие исследования будут направлены на разработку прототипа мультипроцессора и решение новых практических задач с помощью предложенной алгебры.

Список литературы: 1. Zorian Yervant. Test Strategies for System-in-Package // The Plenary Paper of IEEE East-West Design & Test Symposium (EWDTS'08). Lvov, Ukraine. October 9-12, 2008. 2. Smith L. 3D Packaging Applications, Requirements, Infrastructure and Technologies // Fourth Annual International Wafer-Level Packaging Conference. San Jose, California. September, 2007. 3. *The next Step in Assembly and Packaging: System Level Integration in the package (SiP)* / Editors: William Chen, W. R. Bottoms, Klaus Pressel, Juergen Wolf // SiP White Paper. International Technology Roadmap for Semiconductors. 2007. P. 17-23. 4. Какурин Н.Я., Хаханов В.И., Лобода В.Г., Какурина А.Н. / Регистр сдвига. А.С. №1439682. 22.07.88. 4с. 5. Бондаренко М.Ф., Дударь З.В., Ефимова И.А., Лецинский В.А., Шабанов-Кушнарченко С.Ю. О мозгоподобных ЭВМ // Радиотехника и информатика. 2004. № 2. С. 89–105. 6. Бондаренко М.Ф., Шабанов-Кушнарченко Ю.П. Об алгебре предикатов // Бионика интеллекта. 2004. № 1. С. 15–26. 7. Бондаренко М.Ф., Шабанов-Кушнарченко Ю.П. Теория интеллекта. Учебник. Харьков: СМИТ, 2006. 592 с. 8. Бондаренко М.Ф., Шабанов-Кушнарченко 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. 509с. 11. Амтетков 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: Учебное пособие для вузов. М.: Сов. радио, 1980. 270с. 13. Bergeron J. Writing Testbenches Using SystemVerilog / J. Bergeron // Springer Science and Business Media, Inc., 2006. 414 p. 14. Abramovici M., Breuer M.A. and Friedman A.D. Digital System Testing and Testable Design. Comp. Sc. Press. 1998. 652 p. 15. Densmore Douglas A Platform-Based taxonomy for ESL Design / Douglas Densmore, Roberto Passerone, Alberto Sangiovanni-Vincentelli // 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: ХНУРЭ, 2009. 484с.

Поступила в редколлегию 15.04.2010

Хаханов Владимир Иванович, декан факультета КИУ ХНУРЭ, д-р техн. наук, профессор кафедры АПВТ ХНУРЭ. Научные интересы: техническая диагностика цифровых систем, сетей и программных продуктов. Увлечения: баскетбол, футбол, горные лыжи. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-326. E-mail: hahanov@kture.kharkov.ua.

Чумаченко Светлана Викторовна, д-р техн. наук, профессор кафедры АПВТ ХНУРЭ. Научные интересы: математическое моделирование и вычислительные методы, методы дискретной оптимизации. Увлечения: путешествия, музыка, поэзия, спорт. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-326. E-mail: ri@kture.kharkov.ua.

Литвинова Евгения Ивановна, канд. техн. наук, доцент кафедры АПВТ ХНУРЭ. Научные интересы: техническая диагностика цифровых систем, сетей и программных продуктов. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-326. E-mail: kiu@kture.kharkov.ua.