НАУКОВО-ТЕХНІЧНИЙ ЖУРНАЛ *ISSN 1681-4886*

4 (83)' 2010

Виходить 6 разів на рік

Видається з 23 квітня 1996 р.

ІНФОРМАЦІЙНО -КЕРУЮЧІ СИСТЕМИ НА ЗАЛІЗНИЧНОМУ ТРАНСПОРТІ

INFORMACIJNO-KERUÛCI SISTEMI NA ZALIZNICNOMU TRANSPORTI

УДК 658.512.011:681.326:519.713

ХАХАНОВ В. И., д.т.н., профессор
ЧУМАЧЕНКО С. В., д.т.н., профессор
ХАХАНОВА А. В., к.т.н., доцент
TIECOURA YVES (ХНУРЭ)

Параллельные мультипроцессорные процесс-модели векторно-логического анализа

Введение

Проблема обработки больших массивов логических данных в процессе поиска, распознавания информации и принятия решений может быть решена путем организации вычислений на мультипроцессорной системе с использованием векторно-логического подхода, который позволяет существенно уменьшить время получения решения по сравнению с традиционной обработкой информации.

Целью данного исследования является существенное (x100) повышение быстродействия процедур поиска, распознавания и принятия решений путем мультипроцессорной и параллельной реализации ассоциативно-логических векторных операций для анализа графовых и табличных структур данных в дискретном булевом пространстве без использования арифметических операций.

Для достижения цели необходимы: 1) разработка структур данных и процесс-моделей; 2) создание метрики векторно-логического пространства и упрощение критерия качества векторно-логического анализа; 3) оценка эффективности проектного решения.

Объектом исследования является инфраструктура векторно-логического анализа на основе использования алгебры векторной логики, мультипроцессорной платформы анализа ассоциативно-логических структур данных и неарифметического интегрального критерия качества.

Предмет исследования – процесс-модели поиска, распознавания и выбора решения на основе неарифметического интегрального критерия качества путем использования мультипроцессорной системы на кристалле, оперирующей векторными логическими операциями.

Источники: модели и методы дискретного анализа и синтеза [1-4]; проектирование мультипроцессорных средств решения информационно-логических задач [5-8]; мозгоподобные и интеллектуальные логические вычисления [9-15].

Критерий качества векторно-логического анализа и β-метрика

Для оценки взаимодействия между двумя векторами (a, b) введен интегральный критерий качества в векторном булевом пространстве [15], который позволяет определять: близость по Хэммингу двух объектов в n-мерном пространстве как мощность множества пустых пересечений, а также векторно-логическую меру принадлежности объектов друг другу:

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

Данный критерий качества однозначно определяет три формы взаимодействия двух любых объектов в n-мерном векторном логическом пространстве: расстояние, и две функции принадлежности. Учитывая, что все три оценки входящие в интегральный критерий объединены по функции or, упрощение формы взаимодействия векторов дает результат

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

Критерий качества $Q = m \oplus A$ согласуется с метрикой оценивания расстояния или взаимодействия в векторно-логическом пространстве, а также имеет тривиальную вычислительную форму для оценивания многочисленных решений, связанных с анализом и синтезом информации. В самом деле, логическое век-

торное пространство не должно иметь метрического расстояния и критериев качества, включающих арифметические операции на скалярных величинах. ВЛК определяет не только расстояние между непересекающимися объектами векторно-логического пространства, но и их взаимную принадлежность: $d(a,b) \vee \mu(a,b) \vee \mu(b,a)$, если они пересекаются.

Векторное дискретное логическое (булево) пространство определяет взаимодействие объектов путем использования трех аксиом (тождественности, симметрии и треугольника), формирующих неарифметическую B-метрику векторного измерения:

$$B = \begin{cases} d(a,b) = a \oplus b = (a_i \oplus b_i), i = \overline{1,n}; \\ d(a,b) = [0 \leftarrow \forall i(d_i = 0)] \leftrightarrow a = b; \\ d(a,b) = d(b,a); \\ d(a,b) \oplus d(b,c) = d(a,c), \\ \oplus = [d(a,b) \wedge \bar{d}(b,c)] \vee [\bar{d}(a,b) \wedge d(b,c)]. \end{cases}$$

Вершины в транзитивном треугольнике есть векторы, идентифицирующие объекты в n-мерном булевом B-пространстве. Стороны треугольника d(a,b), d(b,c), d(a,c) есть расстояния между вершинами, которые также представлены векторами размерностью n, где каждый разряд определен в том же алфавите, что и координаты векторов-вершин.

Векторный транзитивный треугольник имеет полную аналогию численному измерению расстояния в метрическом M-пространстве, которое задается системой аксиом, определяющей взаимодействие одной, двух и трех точек в любом пространстве:

$$M = \begin{cases} d(a,b) = 0 \leftrightarrow a = b; \\ d(a,b) = d(b,a); \\ d(a,b) + d(b,c) \geq d(a,c). \end{cases}$$

Специфика аксиомы метрического треугольника заключается в численном (скалярном) сравнении расстояний трех объектов, когда интервальная неопределенность ответа – две стороны треугольника могут быть больше либо равны третьей – малопригодна для определения точной длины последней стороны. Устранить данный недостаток можно только в логическом векторном пространстве, которое может иметь детерминированное представление о каждом параметре состояния процесса или явления. Тогда численная неопределенность третьей стороны треугольника в векторном логическом пространстве приобретает форму точного двоичного вектора, который характеризует расстояние между двумя объектами и вычисляется на основе знания расстояний двух других сторон треугольника:

$$d(a,b) \oplus d(b,c) = d(a,c).$$

Можно перенести правый компонент в левую часть, что дает возможность сворачивать любое замкнутое пространство в нуль вектор $d(a,b) \oplus d(b,c) = d(a,c) \rightarrow d(a,b) \oplus d(b,c) \oplus d(a,c) = 0$.

Свертка пространства в нуль вектор представляет интерес для многих практических задач, включая: 1) Диагностирование и исправление ошибок при передаче информации по каналам связи. 2) Поиск дефектов в цифровых изделиях на основе двузначных и многозначных таблиц неисправностей. Далее представлено теоретическое обоснование свертывания пространства путем доказательства корректности использования метрики векторного булевого пространства в целях измерения взаимодействия логических конструкций, включая точку, линию, плоскость.

Аксиома 1. Для булевых переменных действительно логическое выражение: $a = b \rightarrow a \oplus b = 0$. В самом деле, сумма по модулю 2 есть функция неравнозначности, которая принимает истинное или единичное значение при несовпадении значений переменных $a \neq b \rightarrow a \oplus b = 1$ и нулевое – при равенстве аргументов.

Определение 1. Расстояние между двумя точками n-мерного пространства есть вектор, вычисленный на основе сложения по модулю 2 (далее xor-сумма) одноименных координат:

$$d(a,b) = a_i \overset{n}{\underset{i=1}{\oplus}} b_i .$$

Определение 2. Вектор-расстояние между двумя точками (объектами) n-мерного логического пространства равно нулю (единице), если все координаты вектора равны нулю (единице):

$$d(a,b) = 0(1) \leftarrow \forall i[a_i \overset{n}{\underset{i=1}{\oplus}} b_i = 0(1)].$$

Определение 3. Простая цепь – последовательность векторных расстояний и точек пространства, среди которых нет повторяющихся компонентов: расстояний, точек.

Определение 4. Векторно-логический цикл D есть совокупность векторных расстояний $d_i \in D$ между точками пространства, создающими замкнутую простую цепь, в которой первая и последняя точки пространства совпадают.

Теорема 1. Xor-сумма векторных расстояний, образующих цикл, между двумя точками n-мерного пространства равна нулю: $d(a,b) \oplus d(b,a) = 0$.

Это следует из аксиомы симметрии (коммутативности), которая предполагает, что вектор-расстояния между двумя любыми точками n-мерного пространства $d(a,b) = d(b,a)$ равны между собой. Но, согласно аксиоме 1, перенос правой части равенства в левую сопровождается регулированием отношений между компонентами с помощью xor-операции, результат которой равен нулю: $d(a,b) \oplus d(b,a) = 0$ ввиду равенства векторных расстояний.

Теорема 2. Xor-сумма векторных расстояний или двух сторон в транзитивном треугольнике равна третьей.

Доказательство. В общем случае метрика определяет взаимодействие трех точек {a, b, c} (сторон треугольника) в пространстве путем формирования трех расстояний $d(a,b), d(b,c), d(a,c)$. Допустим, что имеются два расстояния $d(a,b), d(b,c)$, которые, согласно теореме 1, создают равенство $d(a,b) \oplus d(b,c) = 0$, имеющее три варианта взаимодействия точек: 1) $d(a,b) = d(b,c) = 0$ – имеется одна точка, обозначенная в данном случае идентификаторами a,b,c, расстояние между которыми равно нулю. 2) $d(a,b) = d(b,c) \rightarrow d(a,b) \oplus d(b,c) = 0$ – имеются две точки {b, a=c}, которые формируют два одинаковых расстояния, создающих цикл, в соответствии с принципом симметрии или коммутативности. 3) $d(a,b) \neq d(b,c) \rightarrow d(a,b) \oplus d(b,c) \neq 0 \rightarrow d(a,b) \oplus d(b,c) = d(a,c)$ – имеются два неравных между собой расстояния $d(a,b) \neq d(b,c)$, которые возможны лишь при взаимодействии трех точек пространства a,b,c в целях определения третьего расстояния с помощью векторной операции $d(a,b) \oplus d(b,c) = d(a,c)$. При этом вектор, заданный в правой части равенства, никогда не будет равен одному из слагаемых левой части ввиду того, что $d(a,b) \neq d(b,c)$. Таким образом, отношения трех любых точек векторного логического пространства могут быть сведены к формальному взаимодействию, записанному с помощью равенства $d(a,b) \oplus d(b,c) = d(a,c)$, которое, вырождаясь, также регулирует соотношения двух и одной точки, самой на себя.

Теорема 3. Xor-сумма векторных расстояний в транзитивном треугольнике равна нулю: $d(a,b) \oplus d(b,c) = d(a,c) \rightarrow d(a,b) \oplus d(b,c) \oplus d(a,c) = 0$. Доказательство основывается на применении аксиомы 1 к выражению транзитивного замыкания, полученному в теореме 2.

Теорема 4. Xor-сумма $\bigoplus_{i=1}^{n} d_i = 0$ векторных расстояний $d_i \in D$ в цикле D, составленном из конечного числа вершин (n), равна нулю:

1) $d_1 = 0$;
2) $d_1 \oplus d_2 = 0$;
3) $d_1 \oplus d_2 \oplus d_3 = 0$;
4) $d_1 \oplus d_2 \oplus d_3 \oplus d_4 = 0$;
5) $d_1 \oplus d_2 \oplus \ldots \oplus d_i \oplus \ldots \oplus d_n = 0$.

Доказательство следует из применения теорем 1-3 к расстояниям между точками (вершинами), составляющим замкнутые циклы. В первом случае фиксируется расстояние транзитивного замыкания точки самой на себя. Во втором – расстояние между двумя транзитивно замкнутыми точками. В третьем – между тремя точками пространства. В четвертом – между четырьмя точками. Пятый случай обобщает наличие нулевого расстояния в сумме транзитивных замыканий любого числа точек в n-мерном векторном пространстве.

Следствие 1. Xor-сумма любых двоичных кодов одинаковой длины равна нулю, если они составляют цикл.

Следствие 2. Метрика β векторного логического пространства определяется единственным равенством, которое формирует нулевую xor-сумму расстояний между ненулевым и конечным числом точек, замкнутых в цикл:

$$\beta = \bigoplus_{i=1}^{n} d_i = 0.$$

Определение 5. Метрика β векторного логического пространства есть равная нулю xor-сумма расстояний между конечным числом точек цикла.

Метрика β векторного логического пространства ставит во главу угла не элементы множества, но отношения, что позволяет сократить аксиоматику с трех до одной формулы и распространить ее действие на сколь угодно сложные конструкции n-мерного пространства.

Пример. Имеется пять точек в векторном пространстве: (000111, 111000, 101010, 010101, 110011). Замыкание этих точек в цикл дает следующие стороны-расстояния в пятиугольнике: (111111, 010010, 111111, 100110, 110100). Покоординатное сложение всех векторов дает результат: (000000). Практическая значимость данного факта заключается в возможности восстановления любого расстояния в замкнутом цикле, если известны (n-1) сторон фигуры. Для треугольника это означает восстановление третьей стороны по известным двум. Если же создать логическое пространство, составленное из треугольников, то можно сэкономить 66% от объема данных, который формирует все расстояния в логическом пространстве.

Процесс-модели векторно-логического анализа

Инфраструктура представляет собой совокупность моделей, методов и средств описания, анализа и синтеза структур данных для решения функциональных задач. Модель (системная) – это совокупность взаимосвязанных, определенных в пространстве и времени компонентов, с заданной адекватностью описывающая процесс или явление и используемая для достижения поставленной цели при наличии ограничений и метрики оценивания качества решения. Здесь ограничения есть аппаратурные затраты, время разработки и производства до появления изделия на рынке (time-to-market), подлежащие минимизации. Метрика оценивания решения при использовании модели опре-

делена двоичным логическим вектором в дискретном булевом пространстве. Концептуальная модель вычислительного изделия представлена совокупностью управляющего и операционного автоматов. Системная модель LAMP функциональности использует GALS (Global Asynhronus Local Synchronus) [8] технологию создания иерархических цифровых систем с локальной синхронизацией отдельных модулей и одновременно глобальной асинхронностью функционирования всего устройства.

В целях детализации структуры векторного процессора и секвенсора ниже представлены аналитические и структурные процесс-модели, которые приводятся к анализу А-матрицы по столбцам или строкам. Первая из них представлена на рис. 1 и предназначена для определения множества допустимых решений относительно входного запроса m_b

$$m_{ai}^{m} = \bar{\vee}\,[(m_b \overset{n}{\underset{i=1}{\wedge}} A_i) \oplus m_b];$$

$$A_i = (m_b \overset{n}{\underset{i=1}{\wedge}} A_i).$$

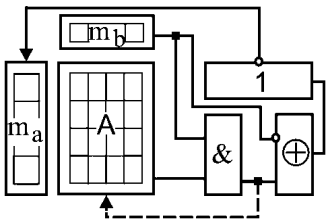

Рисунок 1 – Поиск всех допустимых решений

Вторая структура (рис. 2) осуществляет поиск оптимального решения на множестве найденных в первой процессной модели путем анализа строк. Кроме того, вторая модель имеет и самостоятельное применение, ориентированное на определение однозначного и многозначного решения при поиске дефектов в цифровой системе на кристалле.

$$m_b^{s} = (\underset{\forall m_{ai}=1}{\wedge} A_i) \wedge (\overline{\underset{\forall m_{ai}=0}{\vee} A_i})$$

$$m_b^{m} = (\underset{\forall m_{ai}=1}{\vee} A_i) \wedge (\overline{\underset{\forall m_{ai}=0}{\vee} A_i})$$

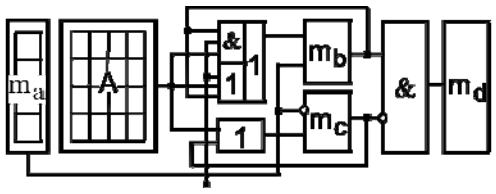

Рисунок 2 – Структура выбора оптимального решения

Все операции, представленные в двух моделях процессов, являются векторными. Процесс-модель анализа строк (рис. 1) 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 $A = \mathrm{card}(m \times n)$. Качество (допустимость) решения определяется для каждого взаимодействия входного вектора m_b и строки $A_i \in A$ на блоке (девекторизации) дизъюнкции. Матрица A может быть модифицирована путем ее пересечения с входным вектором на основе использования операции $A_i = (m_b \overset{n}{\underset{i=1}{\wedge}} A_i)$, если необходимо исключить из А-таблицы все незначимые для решения координаты и векторы, отмеченные единичными значениями вектора m_a. Решение задачи диагностирования путем анализа строк таблицы, представленное на рис. 2, необходимо интерпретировать следующим образом. После выполнения диагностического эксперимента формируется двоичный вектор экспериментальной проверки m_a, который маскирует А-таблицу неисправностей для поиска одиночных или кратных дефектов. Векторы m_b и m_c используются для накопления результатов выполнения операций конъюнкции и дизъюнкции. Затем осуществляется логическое вычитание 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. Схема имеет также переменную выбора режима поиска решения: single или multiple. Процесс-модель использует в качестве входного условия вектор m_a, который управляет 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 А-таблицы. В результате выполнения n тактов осуществляется накопление единичных и нулевых относительно значений координат вектора m_a решений в регистрах A_1, A_0 соответственно. Априори в указанные регистры заносится вектор единиц и нулей: $A_1 = 1, A_0 = 0$. После обработки всех n строк А-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. При анализе таблицы неисправностей цифрового изделия единичным координатам вектора m_b соответствуют столбцы, отождествляемые с номерами дефектов или неисправных

блоков, подлежащих восстановлению или ремонту.

Можно пойти еще дальше в части сервисного обслуживания функциональных модулей: 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 покрыть все обнаруженные в ячейках неисправности. Технологическая и математическая культура векторной логики в данном случае предлагает простое схемотехническое решение для получения квазиоптимального покрытия, рис. 3. Преимущества: 1) Вычислительная сложность процедуры: $Z = n$ векторных операций, равное числу строк таблицы. 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. Недостаток – получение не всегда оптимального покрытия, что является платой за технологичность векторной процедуры, представленной на рис. 3.

Здесь имеется операция девекторизации, которая 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 девекторизации 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 $m_b = 0, m_a = 0$ обнуляются. Квазиоптимальное покрытие накапливается за n тактов в векторе m_a путем последовательного сдвига. Биты, заносимые в регистр m_a, формируются схемой or, которая выполняет девекторизацию путем анализа входного полученного результата $[(m_b \vee A_i) \wedge \overline{m}_b]$ на присутствие единиц.

$$\begin{cases} m_b = (m_b \vee A_i); \\ m_{ai} = \bigvee\limits_{i=1}^{n} [(m_b \vee A_i) \wedge \overline{m}_b]. \end{cases}$$

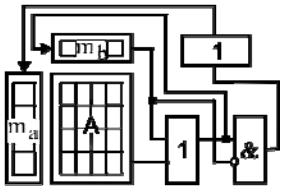

Рисунок 3 – Процесс-модель поиска квазиоптимального покрытия

Следующий пример интересен функциональной законченностью цикла диагностирования, когда после получения квазиоптимального покрытия данная информация используется для восстановления работоспособности дефектных ячеек памяти [8]. Размерность модуля памяти – 13x15 ячеек не влияет на вычислительную сложность получения покрытия десяти дефектных ячеек с помощью резервных строк (2) и столбцов (5) (рис. 4).

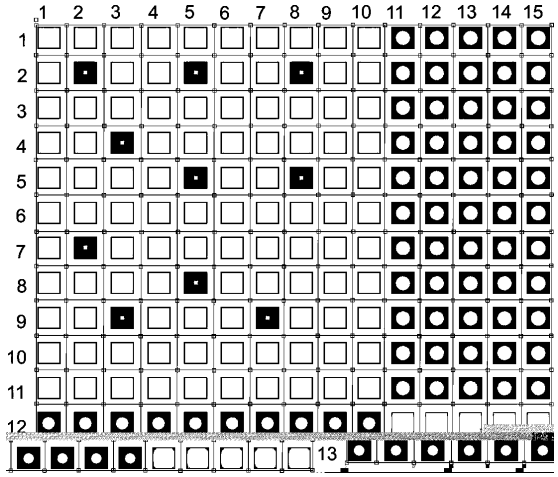

Рисунок 4 – Модуль памяти с резервом и таблица покрытия

Для решения оптимизационной задачи выполняется построение таблицы покрытия (рис. 4) неисправных ячеек, которая имеет строки – резервные ресурсы для покрытия дефектов: $(C_2, C_3, C_5, C_7, C_8, C_2, R_2, R_4, R_5, R_7, R_8, R_9)$, а столбцы представляют собой 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. Здесь столбцы соответствуют координатам дефектных ячеек, а строки идентифицируют резервные компоненты (строки и столбцы), которые могут восстановить работоспособность неисправных координат. Процесс-модель, рис. 1, дает возможность получить оптимальное решение в виде $m_a = \boxed{1\ 1\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0}$, которому соответствует покрытие: $R = \{C_2, C_3, C_5, C_7, C_8\}$, как одно из трех $R = C_2, C_3, C_5, C_7, C_8 \vee C_2, C_3, C_5, C_8, R_9 \vee C_2, C_5, C_8, R_4, R_9$ возможных минимальных решений для таблицы неисправностей. Технологическая модель встроенного диагностирования и ремонта памяти представлена на рис. 5. Она имеет четыре компонента: 1) Testing – тестирование модуля (UUT – Unit Under Test – тестируемое устройство) с использованием эталонной модели (MUT – Model Under Test) для формирования вектора экспериментальной проверки m_a, размерность которого соответствует числу тестовых наборов. 2) Diagnosis – поиск дефектов на основе анализа таблицы

неисправностей A. 3) Optimization – оптимизация покрытия дефектных ячеек ремонтными строками и столбцами на основе анализа таблицы A. 4) Repairing – восстановление работоспособности памяти путем замены адресов (AD – Address Decoder) неисправных строк и столбцов, представленных вектором m_a, на адреса компонентов из резервной памяти SM – Spare Memory [8].

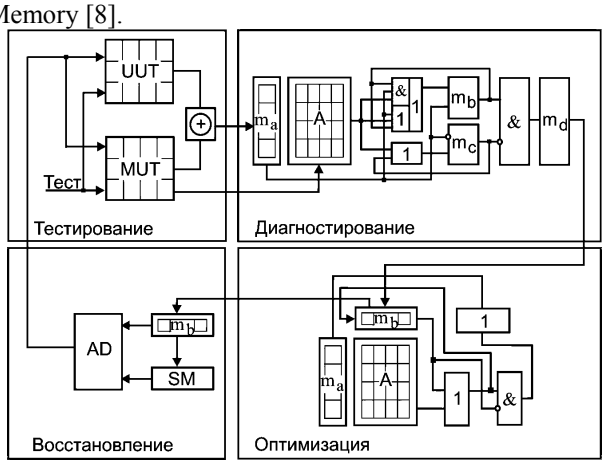

Рисунок 5 – Модель встроенного тестирования и восстановления памяти

Процесс-модель встроенного сервисного обслуживания работает в реальном масштабе времени и позволяет поддерживать в работоспособном состоянии, без вмешательства человека, цифровую систему на кристалле, что является оригинальным решением для критических технологий, связанных с дистанционной эксплуатацией изделия.

Предложенные процесс-модели анализа (графа) ассоциативных таблиц, а также введенные критерии качества логических решений позволяют решать задачи квазиоптимального покрытия, диагностирования дефектов программных и/или аппаратных блоков. Модель векторных вычислений послужила основой для разработки специализированной мультипроцессорной архитектуры, ориентированной на поиск, распознавание и принятие решений при использовании структур ассоциативных таблиц.

Оценка эффективности (рис. 6) проектного решения под эгидой специализации Sp и стандартизации St основывается на совместном использовании трех взаимно противоречивых параметров: качество Y, быстродействие T, аппаратурные затраты H:

$$E = F(Y, T, H),$$
$$Y = (1-P)^{n(1-Q)};$$
$$T = \frac{1}{f} \times S \times d;$$
$$H = 2(H^s \times n).$$

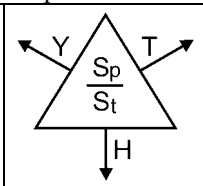

Рисунок 6 – Оценка эффективности процесс-модели

Параметр Y зависит от тестопригодности проекта Q, вероятности P существования в кристалле неисправных областей и числа n необнаруженных дефектов. Время решения задачи определяется: $S \times d$ структурной глубиной схемы, умноженной на среднюю задержку примитива, принадлежащего максимальному логическому пути, отнесенной к тактовой частоте f устройства. Аппаратурные затраты находятся в функциональной зависимости от сложности: $H^s, n, H^u, H^m, H^d, H^i$ – секвенсера, числа логических процессоров, устройства управления, памяти для хранения команд и данных, сервисных средств диагностирования и интерфейса связи. В целях упрощения формулы эффективности можно считать, что матрица логических процессоров равна по сложности остальной части LAMP $(H^s \times n) = H^u + H^m + H^d + H^i$.

Сравнительный анализ эффективности E^n предложенной процесс-модели диагностирования и восстановления работоспособности блока памяти (рис. 5) на основе LAMP по отношению к базовой E^b реализации на универсальном компьютере имеет оценку:

$$\eta = \frac{E^b}{E^n} = \frac{T^b}{T^n} + \frac{H^b}{H^n} = \frac{(S \times d)^b}{(S \times d)^n} + \frac{(H^s \times n)^b}{(H^s \times n)^n} =$$
$$= \frac{200}{20} + \frac{1000000 \text{ gates}}{2(800 \times 16) \text{ gates}} = 10 + 40 = 50.$$

Здесь тактовая частота базового и нового изделия, их качество, а также задержки примитивов считаются равными и поэтому выведены из формулы подсчета результата. Структурная глубина аппаратурных вариантов равна соответственно 200 и 20, число эквивалентных вентилей – 1000000 и 25600. Аддитивная оценка эффективности использования инфраструктуры для решения задачи диагностирования и ремонта памяти дает результат, равный 50. Мультипликативная оценка практически будет на порядок выше.

Выводы

Параллельные векторные мультипроцессорные неарифметические процесс-модели, представленные в исследованиях, ориентированы на новые эффективные решения практических задач синтеза и анализа: минимизация булевых функций, поиск дефектов, восстановление работоспособности, распознавание образов, принятие решений, разработка цифровых фильтров, создание дружественных серверов, сайтов и порталов. Метрика и упрощенный критерий качества векторно-логического пространства позволяют эффективно оценивать взаимодействие объектов в векторном пространстве и существенно экономить на объемах информации, представляющей расстояния между объектами.

Предложенная инфраструктура векторно-логического анализа, позволяет существенно (х100) повысить быстродействие вычислений в информационно-компьютерном пространстве планеты, что определяет рыночную привлекательность исследований.

Литература

1. *Акритас А.* Основы компьютерной алгебры с приложениями: Пер. с англ. / А. Акритас.– М.: Мир.– 1994.– 544 с.
2. *Гилл Ф.* Практическая оптимизация. / Ф. Гилл, У. Мюррей, М. Райт.– М.: Мир.– 1985.– 509 с.
3. *Аттетков А.В.* Методы оптимизации / А.В. Аттетков, С.В. Галкин, В.С. Зарубин.– М.: Издательство МГТУ им. Н.Э. Баумана, 2003.– 440 с.
4. *Дегтярев Ю. И.* Методы оптимизации: Учебное пособие для вузов / Ю. И. Дегтярев. – М.: Сов. Радио, 1980.– 270 с.
5. *Bergeron J.* Writing Testbenches Using SystemVerilog / J. Bergeron // Springer Science and Business Media, Inc, 2006. – 414 p.
6. *Abramovici M.* Digital System Testing and Testable Design / M. Abramovici, M.A. Breuer and A.D. Friedman.– Comp. Sc. Press, 1998.– 652 p.
7. *Densmore D.* A Platform–Based taxonomy for ESL Design / Douglas Densmore, Roberto Passerone, Alberto Sangiovanni–Vincentelli // Design & Test of computers.– 2006. – P. 359–373.
8. *Хаханов В.И.* Проектирование и тестирование цифровых систем на кристаллах / В.И. Хаханов, Е.И. Литвинова, О.А. Гузь. – Харьков: ХНУРЭ, 2009.– 484с.
9. *Cohen A.A.* Addressing architecture for Brain-like Massively Parallel Computers / Euromicro Symposium on Digital System Design (DSD'04).– 2004.– P. 594-597.
10. *Липаев В.В.* Программная инженерия. Методологические основы. Учебник.– М.: Теис.– 2006.– 608 с.
11. *Трахтенгерц Э.А.* Компьютерные методы реализации экономических и информационных управленческих решений. – М.: СИНТЕГ, 2009. – 396 с.
12. *Кузнецов О.П.* О моделировании быстрых интеллектуальных процессов обыденного мышления // Интеллектуальные системы.– 1997.– Т.2, Вып. 1-4.
13. *Кузнецов О.П.* Быстрые процессы мозга и обработка образов // Новости искусственного интеллекта.– 1998.– №2.
14. *Васильев С.Н., Жерлов А.К., Федосов Е.А., Федунов Б.Е.* Интеллектуальное управление динамическими системами. – М.: Физ.-мат. лит-ра, 2000. – 352 с.
15. *Хаханов В.И., Литвинова Е.И., Побеженко И.А., Ngene Christopher Umerah.* Мультипроцессорная инфраструктура анализа информационного пространства // In the Proc. Intellectual Systems for Decision Making and Problems of Computational Intelligence, Yevpatoria, 2010. – P.409-413.

Резюме

Предложены векторно-логические процесс-модели актуальных прикладных задач, качество решения которых оценивается введенной интегральной неарифметической метрикой взаимодействия булевых векторов. Реализация процесс-моделей в инфраструктуре векторно-логического анализа информации позволяет существенно увеличить скорость процедур поиска, распознавания и принятия решений

Запропоновано векторно-логічні процес-моделі актуальних прикладних задач, якість рішення яких оцінюється введеною інтегральною неарифметичною метрикою взаємодії булевих векторів. Реалізація процес-моделей в інфраструктурі векторно-логічного аналізу інформації дозволяє суттєво збільшити швидкодію процедур пошуку, розпізнавання та прийняття рішень

The models are focused to realization of high-performance vector parallel logical analysis of information that in the limit completely excludes the use of arithmetic operations. Implementation of the process models in the infrastructure for vector logic analyzing information allows to increase the speed of searching, pattern recognition and decision making

Ключові слова: мультипроцессор, анализ информации, логическое ассоциативное отношение, процесс-модель

Поступила 16.07.2010 г.