

УДК 658.512.011:681.326:519.713

В.И. Хаханов, С.В. Чумаченко**МУЛЬТИПРОЦЕССОР ДЛЯ АНАЛИЗА
ИНФОРМАЦИОННОГО ПРОСТРАНСТВА. 2. ИНФРАСТРУКТУРА
ВЕКТОРНО-ЛОГИЧЕСКОГО АНАЛИЗА**

Представлены параллельные векторные мультипроцессорные неарифметические процесс-модели, ориентированные 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.

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

Современный рынок электронных и информационных технологий очень 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, а поэтому не всегда являются эффективным инструментом для решения логических, ассоциативных и поисковых задач в существующем океане информации. Поэтому актуальным представляется разработка новых специализированных технологий, ориентированных на создание цифровой системы на кристалле с ограниченной системой логических команд для анализа в реальном масштабе времени больших информационных массивов, организованных в эффективные структуры данных. Такие информационно-ориентированные процессоры могут поддерживать как серверную, так и клиентскую стороны компьютерной сети путем встраивания в системную плату или их соединения через один из портов, например с использованием USB. Если исключить из рассмотрения арифметические операции, которые не свойственны человеческому мозгу, то все ресурсы мозгоподобного компьютера можно переориентировать на повышение мощности и производительности вычислителя для чрезвычайно большого класса проблем логического, ассоциативного анализа, распознавания образов, диагностирования и поиска информации.

Цель данного исследования – существенное ($\times 100$) повышение быстродействия процедур решения актуальных прикладных задач 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 необходимо решить задачи, связанные с разработкой векторно-логических процесс-моделей встроенного диагностирования цифровых систем на кристаллах, поиска квазиоптимального покрытия, использующие средства логического ассоциативного мультипроцессора, параллельные операции вычислительных процессов и подсчета критериев качества.

Источники. 1. Ассоциативно-логические структуры данных для мозгоподобных вычислительных процессов [1 – 5]. 2. Мультипроцессорная аппаратная платформа информационно-логического анализа [6 – 9]. 3. Модели и методы дискретного анализа, синтеза и диагностирования цифровых изделий [10 – 15].

1. Инфраструктура векторно-логического анализа

Инфраструктура – совокупность моделей, методов и средств описания, анализа и синтеза структур данных для решения функциональных задач. Модель (системная) – 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 аппаратные затраты, время разработки и производства до появления изделия на рынке (time-to-market), подлежащие минимизации. Метрика оценивания решения при использовании модели определена 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)-технологию создания иерархических цифровых систем с локальной синхронизацией отдельных модулей и одновременно глобальной асинхронностью функционирования всего устройства [13].

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 .

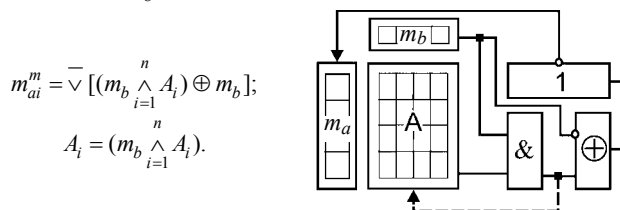


Рис. 1. Поиск всех допустимых решений

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, например, при поиске дефектов в цифровой системе на кристалле.

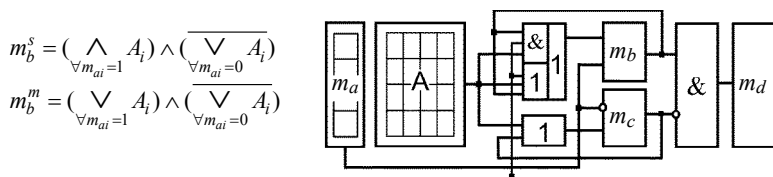


Рис. 2. Структура выбора оптимального решения

Все операции, представленные в двух моделях процессов, являются векторными. Процесс-модель анализа строк (см. 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 $A = \text{card}(m \times n)$. 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 $A_i = (m_b \bigwedge_{i=1}^n A_i)$, если необходимо исключить из A -таблицы все незначимые для решения координаты и векторы, отмеченные единичными значениями вектора m_a . Интересное решение для задач 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 A -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 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 соответственно. 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. 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 и интересное 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 класси-

ческого 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.

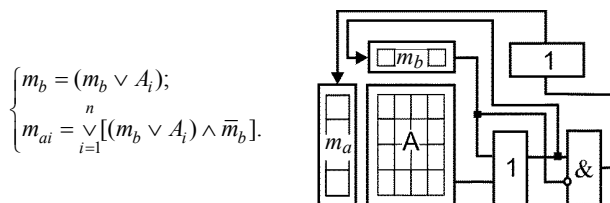


Рис. 3. Процесс-модель поиска квазиоптимального покрытия

Здесь имеется операция девекторизации, которая на последнем этапе превращает векторный результат в бит m_{ai} вектора m_a по функции $m_{ai} = \bigvee [(m_b \vee A_i) \wedge \bar{m}_b]$. В общем случае операция девекторизации в алгебре векторных операций записывается в виде <бинарная операция> <вектор>: $\vee A_i, \wedge m, \bar{\wedge}(m \vee A_i)$. 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 ог, которая выполняет девекторизацию, путем анализа входного полученного результата $[(m_b \vee A_i) \wedge \bar{m}_b]$ на присутствие единиц.

Следующий пример интересен функциональной законченностью цикла диагностирования, когда после получения квазиоптимального покрытия данная информация используется для восстановления работоспособности дефектных ячеек памяти [9]. Размерность модуля памяти – 13×15 ячеек не влияет на вычислительную сложность получения покрытия десяти дефектных ячеек с помощью резервных строк (2) и столбцов (5) (рис. 4).

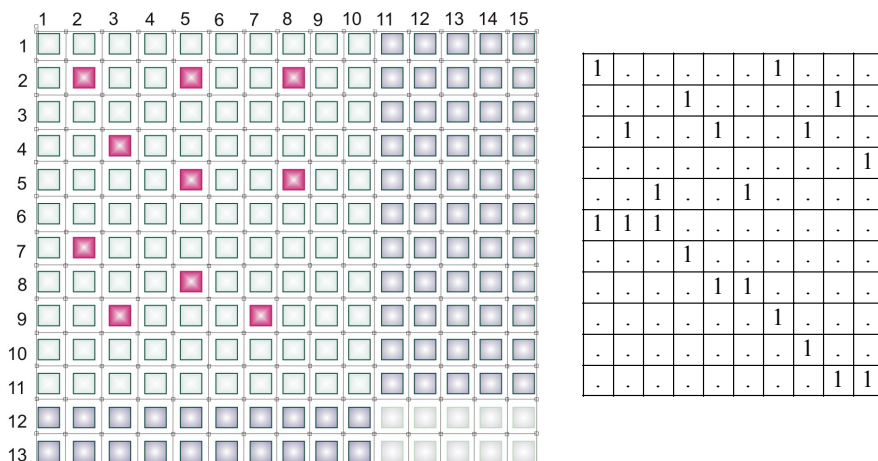


Рис. 4. Модуль памяти с резервом и таблица покрытия

Для решения оптимизационной задачи выполняется построение таблицы покрытия (см. рис. 10) 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. 2, дает возможность получить оптимальное решение в виде

$$m_a = [1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0],$$

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) 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 ремонтного запаса SM – Spare Memory [9].

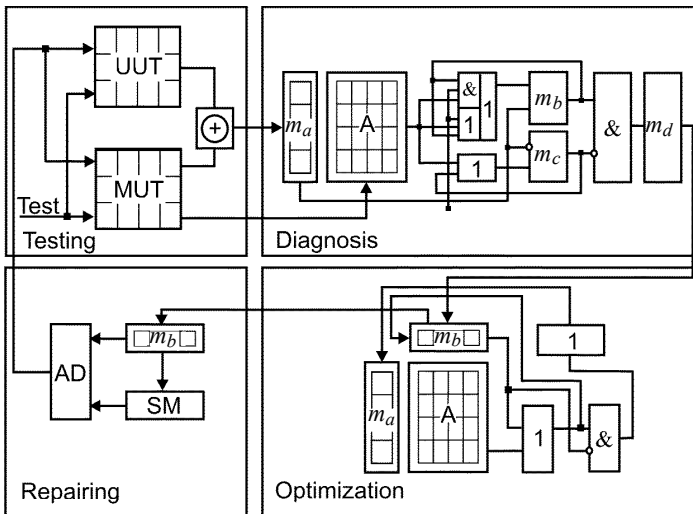


Рис. 5. Модель встроенного тестирования и восстановления памяти

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 интересным 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 на использовании структур ассоциативных таблиц.

Заключение

Существующие программные аналоги не предлагают чисто ассоциативно-логических маршрутов поиска, распознавания и принятия решений в дискретном информационном пространстве [4, 5, 15]. Практически все они используют универсальную систему команд современного дорогостоящего процессора с математическим сопроцессором. Кроме того, аппаратные специализированные средства логического анализа, которые могут выступать прототипами [1, 6, 7], как правило, ориентированы на побитовую или не векторную обработку информации.

Для устранения недостатков программных аналогов и аппаратных прототипов в работе предложен новый подход векторно-логической обработки ассоциативных данных с полным исключением арифметических операций, влияющих на быстродействие и аппаратную сложность, который оказался удачно реализуемым на основе использования современной микроэлектронной аппаратуры в виде мультипроцессорной цифровой системы на кристалле.

Фактическая реализация подхода основывается на предложении инфраструктуры, которая включает следующие компоненты: 1. Процесс-модели анализа ассоциативных таблиц на основе использования векторных логических операций для поиска, распознавания образов, принятия и оценивания решений в векторном дискретном булевом пространстве. Модели ориентированы на достижение высокого быстродействия параллельного векторного логического анализа информации и подсчета критериев качества решения. 2. Мультипроцессорная архитектура параллельного решения ассоциативно-логических задач с минимальным множеством векторных логических операций и полным исключением арифметических команд, что обеспечивает высокое быстродействие, минимальную стоимость и незначительное энергопотребление LAMP, имплементированного в кристалл программируемой логики. 3. Новые векторно-логические процесс-модели встроенного диагностирования цифровых систем на кристаллах, поиска квазиоптимального покрытия, использующие средства логического ассоциативного мультипроцессора, параллельные операции вычислительных процессов и подсчета критериев качества.

Достоверность и практическая значимость полученных результатов подтверждается созданием инфраструктуры мультипроцессора для диагностирования и восстановления работоспособности компонентов памяти в цифровой системе на кристалле. Дальнейшие исследования направлены на разработку прототипа логического ассоциативного мультипроцессора в целях решения актуальных задач по-

иска, распознавания и принятия решений с помощью предложенной инфраструктуры векторного логического анализа.

Статья является продолжением работы [16].

ЛИТЕРАТУРА

1. Бондаренко М.Ф., Дударь З.В., Ефимова И.А. и др. О мозгоподобных ЭВМ // Радиоэлектроника и информатика. Харьков: ХНУРЭ. 2004. № 2. С. 89–105.
2. Cohen A.A. Addressing architecture for Brain-like Massively Parallel Computers // Euromicro Symposium on Digital System Design (DSD'04). 2004. P. 594–597.
3. Кузнецов О.П. Быстрые процессы мозга и обработка образов // Новости искусственного интеллекта. 1998. № 2.
4. Васильев С.Н., Жерлов А.К., Федосов Е.А., Федунев Б.Е. Интеллектуальное управление динамическими системами. М.: Физико-математическая литература, 2000. 352 с.
5. Липаев В.В. Программная инженерия. Методологические основы: учебник. М.: Теис, 2006. 608 с.
6. А.С. №1439682. 22.07.88. Регистр сдвига / Какурин Н.Я., Хаханов В.И., Лобода В.Г., Какурина А.Н. 4с.
7. Гайдук С.М., Хаханов В.И., Обризан В.И., Каменюка Е.А. Сферический мультипроцессор PRUS для решения булевых уравнений // Радиоэлектроника и информатика. 2004. № 4(29). С. 107–116.
8. Хаханов В.И., Литвинова Е.И., Гузь О.А. Проектирование и тестирование цифровых систем на кристаллах. Харьков: ХНУРЭ, 2009. 484 с.
9. Хаханов В.И., Хаханова И.В., Литвинова Е.И., Гузь О.А. Проектирование и верификация цифровых систем на кристаллах. Verilog & System Verilog. Харьков: Новое слово, 2010. 528 с.
10. Акритас А. Основы компьютерной алгебры с приложениями: пер. с англ. М.: Мир, 1994. 544 с.
11. Амтетков А.В., Галкин С.В., Зарубин В.С. Методы оптимизации. М.: Изд-во МГТУ им. Н.Э. Баумана, 2003. 440 с.
12. Abramovici M., Breuer M.A. and Friedman A.D. Digital System Testing and Testable Design // Comp. Sc. Press. 1998. 652 p.
13. Densmore D., Passerone R., Sangiovanni-Vincentelli A. A Platform-Based taxonomy for ESL Design // Design & Test of computers. 2006. P. 359–373.
14. Мальшиенко Ю.В. и др. Автоматизация диагностирования электронных устройств / под ред. В.П. Чипулиса. М.: Энергоатомиздат, 1986. 216 с.
15. Трахтенгерц Э.А. Компьютерные методы реализации экономических и информационных управленческих решений. М.: СИНТЕГ, 2009. 396 с.
16. Хаханов В.И., Литвинова Е.И. Мультипроцессор для анализа информационного пространства. 1. Архитектура логического ассоциативного мультипроцессора // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2010. № 1(14). С. 95–108.

Хаханов Владимир Иванович

Чумаченко Светлана Викторовна

Харьковский национальный университет радиоэлектроники (Украина)

E-mail: hahanov@kture.kharkov.ua; ri@kture.kharkov.ua

Поступила в редакцию 14 сентября 2010 г.