На правах рукописи

Колосов Герман Геннадьевич

ОРГАНИЗАЦИЯ И ПРОЕКТИРОВАНИЕ ФУНКЦИОНАЛЬНО-ОРИЕНТИРОВАННЫХ ПРОЦЕССОРОВ ОБРАБОТКИ ПРОДУКЦИОННЫХ ЗНАНИЙ НА ОСНОВЕ RETE-СЕТИ ДЛЯ ИНТЕЛЛЕКТУАЛЬНЫХ АГЕНТОВ РЕАЛЬНОГО ВРЕМЕНИ

Специальность: 05.13.15 – Вычислительные машины и системы

АВТОРЕФЕРАТ
диссертации на соискание ученой степени
кандидата технических наук

Санкт-Петербург – 2005

Работа выполнена в Санкт-Петербургском государственном электротехническом университете «ЛЭТИ» им. В.И. Ульянова (Ленина)

Научный руководитель –
кандидат технических наук, доцент Пантелеев М.Г.

Официальные оппоненты:
доктор технических наук, доцент Сергеев М. Б.,
кандидат технических наук, доцент Крылов Б. А.

Ведущая организация – Санкт-Петербургский институт информатики и автоматизации РАН

Защита диссертации состоится « 1 » ноября 2005г. в 10[00] часов на заседании диссертационного совета Д 212.238.01 Санкт-Петербургского государственного электротехнического университета «ЛЭТИ» им. В.И. Ульянова (Ленина) по адресу: 197376, Санкт-Петербург, ул. Проф. Попова, 5.

С диссертацией можно ознакомиться в библиотеке университета.

Автореферат разослан « 28 » сентября 2005г.

Ученый секретарь
диссертационного совета Пантелеев М.Г.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы. Системы искусственного интеллекта (СИИ) в последние годы находят все более широкое применение в различных областях, в том числе в бортовых и встраиваемых приложениях. Современный этап развития СИИ связан с разработкой теории, методов и средств построения *интеллектуальных агентов* (ИА) – нового класса систем, способных автономно целенаправленно функционировать в открытых, динамических и неопределенных средах.

Важнейшим требованием к таким системам является обеспечение режима реального времени (РВ). При этом ИА, согласно современным представлениям, являются системами «ограниченной эффективности», т. е. не могут в общем случае оптимально решать за отведенное время все стоящие перед ними задачи, а должны максимально эффективно использовать все имеющиеся у них время и ресурсы для принятия решений. Очевидно, что эффективность функционирования таких систем в приложениях жесткого реального времени может быть существенно повышена за счет использования средств аппаратной поддержки алгоритмов, составляющих значительный удельный вес в их архитектуре.

Как показал анализ различных архитектур и приложений ИА, при построении агентов широко используются продукционные системы (ПС), задаваемые совокупностью правил «Если ..., то ...». Данная модель используется в разных подсистемах агента для представления как предметных, так и управляющих знаний и в ряде случаев составляет до 50% алгоритмического обеспечения ИА. Таким образом, эффективность функционирования ИА в открытых динамических средах может быть существенно повышена за счет средств аппаратной поддержки алгоритмов данного класса.

Существующие подходы к аппаратной поддержке ПС либо недостаточно универсальны, либо приводят к резкому увеличению аппаратных затрат с ростом объема баз знаний (БЗ). Наиболее универсальной и эффективной с вычислительной точки зрения формой представления продукционных БЗ (ПБЗ) является RETE-сеть. Именно такая форма представления реализована в широко распространенных программных средствах построения ИА (CLIPS, JESS).

Быстрый прогресс технологии программируемых логических интегральных схем (ПЛИС) создал возможность реализации высокопроизводительных функционально-ориентированных процессоров (ФОП) обработки знаний в конструктиве типовых плат расширения, подключаемых к стандартным магистральным интерфейсам базовых вычислительных платформ. Перспективным направлением реализации ФОП обработки знаний во встроенных системах является использование «систем на кристалле» (System-on-programmable-chip), получивших значительное развитие в последние годы. Такой подход позволяет повысить функциональность и производительность ФОП за счет снижения затрат на взаимодействие с базовым процессорным ядром и возможности оперативной реконфигурации дополнительных аппаратных средств.

Цель и задачи исследования. Целью диссертационной работы является разработка принципов организации, архитектур и методов проектирования функционально-ориентированных процессоров обработки продукционных баз знаний на основе RETE-сети.

В соответствии с поставленной целью, в работе формулируются и решаются следующие основные задачи:

1. Анализ известных подходов к построению средств аппаратной поддержки ПС на основе RETE-сетей с точки зрения эффективности их использования при построении ИА РВ.

2. Разработка виртуальной машины логического вывода для продукционных баз знаний на основе RETE-сети, ориентированной на аппаратную интерпретацию.

3. Анализ и разработка эффективных с точки зрения аппаратной реализации способов внутреннего представления и методов оптимизации RETE-сети;

4. Разработка базовых архитектур и методики проектирования ФОП аппаратной поддержки ПС на основе RETE-сети;

5. Разработка методов и алгоритмов эффективной компиляции ПБЗ в форматы внутреннего представления оптимизированной RETE-сети;

6. Разработка методики и экспериментальная оценка производительности ФОП с использованием макетного образца.

Предмет и методы исследования

Предметом исследования являются способы представления и обработки ПБЗ на основе RETE-сети и принципы организации ФОП аппаратной поддержки алгоритмов логического вывода для ПС на базе RETE-сетей. При решении поставленных задач использовались методы теории оптимизации, в частности целочисленное линейное программирование; теории автоматов; методы алгоритмизации и программирования на языках C++, Fortran; методы проектирования и разработки цифровой аппаратуры на базе ПЛИС с использованием языков описания дискретных устройств (VHDL).

Научную новизну работы составляют:

1. Модификации RETE-сети, ориентированные на аппаратную реализацию продукционных систем, отличающиеся от известных использованием статических структур данных, что позволяет при незначительном росте объема памяти БЗ существенно упростить архитектуру и повысить производительность ФОП.

2. Базовая архитектура и варианты структурной организации ФОП, ориентированные на предложенный способ представления RETE-сети и учитывающие представление и обработку в продукционных БЗ неопределенной информации.

3. Постановка и решение задачи определения оптимальной структуры подсетей β-узлов RETE-сети по критерию минимального времени логического вывода в терминах целочисленного линейного программирования.

4. Метод и алгоритм быстрой компиляции БЗ в модифицированную RETE-сеть с оптимизацией за счет склеивания β-узлов, отличающийся использованием матричных и побитовых преобразований над антецедентами правил. Метод позволяет исключить комбинаторный перебор конфигураций сети и обеспечивает предсказуемое время ее построения.

Практическая значимость работы заключается в следующем:

1. Реализован алгоритм быстрой компиляции исходной БЗ в оптимальную RETE-сеть с квадратичной временной сложностью и предсказуемым временем обработки, позволяющий использовать ФОП в ИА РВ с динамическим формированием БЗ.

2. Разработан пакет VHDL-модулей масштабируемого ядра ФОП обработки ПБЗ на основе RETE-сети, позволяющий проектировать подобные устройства с учетом различных форм неопределенности и требований по параметрам обрабатываемых БЗ, таких как число объектов, глубина сети, ширина входного токена, и т.п.

3. Разработан и реализован экспериментальный образец ФОП на основе ПЛИС, позволяющий достоверно оценить эффективность предложенного подхода к построению средств аппаратной поддержки ПС.

4. Написан ряд программных модулей для взаимодействия ФОП с базовой машиной и экспериментальной оценки его производительности. Разработанное ПО может в дальнейшем использоваться в составе инструментальной среды проектирования устройств подобного класса.

5. С использованием созданного аппаратно-программного комплекса проведено экспериментальное исследование и установлено, что выигрыш ФОП в производительности по сравнению с программными интерпретаторами на процессорах общего назначения составляет от 10 до 40 раз в зависимости от характеристик БЗ и БД. Показано, что время обработки одного токена и всей БЗ являются предсказуемыми величинами, что имеет существенное значение при построении ИА РВ.

Достоверность результатов исследования подтверждается корректным использованием математического аппарата и результатами экспериментальных исследований производительности ФОП.

Внедрение результатов работы. Результаты диссертационной работы, полученные в ходе выполнения НИР 5937/ВТ-210, использовались на предприятии ФГУП НПП «Рубин» при разработке средств аппаратной поддержки экспертных систем реального времени для автоматизированных систем оценки и прогнозирования воздушной обстановки.

Работа также была поддержана грантом Минобразования РФ "Организация и проектирование процессоров аппаратной поддержки объектно-продукционных моделей представления и обработки знаний для интеллектуальных агентов реального времени" (НИР ВТ-39/ГТАТ, проект ТОО-3.3-2672) и двумя персональными грантами СПбГЭТУ за 2001 и 2002 гг.

Апробация результатов исследования. Основные положения и результаты работы докладывались и обсуждались на: 7-ой национальной конференции по искусственному интеллекту с международным участием «КИИ'2000» (Переславль-Залесский, 2000); 4-й и 5-й международных симпозиумах «Интеллектуальные системы» ИНТЕЛС (Москва, 2000; Калуга, 2002); 4-й международной конференции по мягким вычислениям и измерениям SCM'2001 (Санкт-Петербург, 2001); международном конгрессе «Искусственный интеллект в XXI веке» ICAI'2001 (Двиноморское, 2001); международной НТК «СуперЭВМ и многопроцессорные вычислительные системы» MCS'2002 (Таганрог, 2002); международной конференции «Искусственные интеллектуальные системы» IEEE ICAIS'2002 (Двиноморское, 2002); 4-й и 5-й международных НТК «Искусственный интеллект. Интеллектуальные и многопроцессорные системы» ИМС (Двиноморское, 2003; Таганрог, 2004); НТК ППС СПбГЭТУ за 2001, 2002 и 2003 гг.

Публикации. По теме диссертации опубликовано 15 научных работ, в том числе 4 статьи в журналах и 1 авторское свидетельство на полезную модель.

Структура и объем работы.

Диссертация состоит из введения, четырех глав, заключения, трех приложений и списка литературы, включающего 202 наименования. Основная часть работы изложена на 146 страницах машинописного текста. Работа содержит 68 рисунков и 2 таблицы.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Во введении обосновывается актуальность диссертационной работы, определяются цель и задачи исследования, формулируются научная новизна и практическая ценность результатов.

В первой главе рассмотрены основные архитектуры ИА РВ, показаны место и роль в них ПС, описана классическая RETE-сеть и ее модификации, проанализированы известные подходы к построению средств аппаратной поддержки ПС.

На примере ряда известных архитектур ИА РВ (SOAR, Guardian, REX и др.) показано, что агенты представляют собой гибридные системы ИИ, т.е. строятся с использованием разных моделей представления знаний и традиционных вычислительных алгоритмов. Значительный удельный вес при этом составляют ПС, используемые для представления как предметных, так и управляющих знаний.

Ядро ПС включает: *базу знаний*, представленную набором продукционных правил вида ЕСЛИ=>ТО; *базу данных* (рабочую память), содержащую множество фактов (элементов рабочей памяти – ЭРП), и *машину логического вывода* (МЛВ), называемую также интерпретатором ПС. Левая часть правил задается набором условных элементов (УЭ) или предикатов, а правая часть – группой операторов. МЛВ реализует процесс вывода, включающий фазы: *сопоставления (matching), разрешения конфликтов (conflict resolution)*, и *выполнения действий (action)*. В процессе вывода МЛВ сопоставляет факты БД с УЭ правил. Известно, что вычислительные затраты на фазу сопоставления составляют до 90% общих вычислительных затрат при обработке ПС.

Известные на сегодняшний день способы представления ПБЗ являются разновидностями трех базовых методов: списочная форма, деревья решений (ДР) и RETE-сети. Область применения ДР ограничивается задачами описания данных, классификации и регрессии. Ключевой недостаток списочной формы заключается в необходимости сопоставления на каждой итерации всех фактов, независимо от того, изменились они или нет за время предыдущей итерации. Наиболее эффективным способом представления ПБЗ является RETE-сеть — граф потока данных специального вида, позволяющий исключить избыточность вычислений при обработке. На рис. 1 представлена RETE-сеть, соответствующая правилу:

Самолет ?P (Высота *малая*, Скорость *высокая*, Курс *?X*)
И Объект *?K* (Направление *?X*, Скорость *средняя*)
=> Цель *?P* = атака объекта *?K*

Сеть содержит три типа узлов с памятью: α-узлы, β-узлы и терминальные. Добавление или удаление факта в БД влечет подачу соответствующего токена на вход RETE-сети. Токен попадает в α-узел, определяемый набором констант. Далее, он подается на вход β-узла, где сопоставляется с поступившими ранее на другой вход по значению одной из переменных. При совпадении результат в виде кортежа ЭРП поступает на следующий ярус сети. В терминальный узел поступает набор

ЭРП, удовлетворяющий условию правила. В конфликтное множество при этом добавляется пара правило и набор удовлетворяющих ему объектов.

Существующие подходы к аппаратной поддержке RETE-сетей (DADO, Oflazer, Non-Von и др.) ориентированы на отображение состава БЗ в аппаратную структуру вычислителя, что влечет существенный рост аппаратных затрат и затрудняет возможность оперативной модификации БЗ или обработки разных БЗ одним аппаратным модулем. Системы с массовым параллелизмом и программируе-

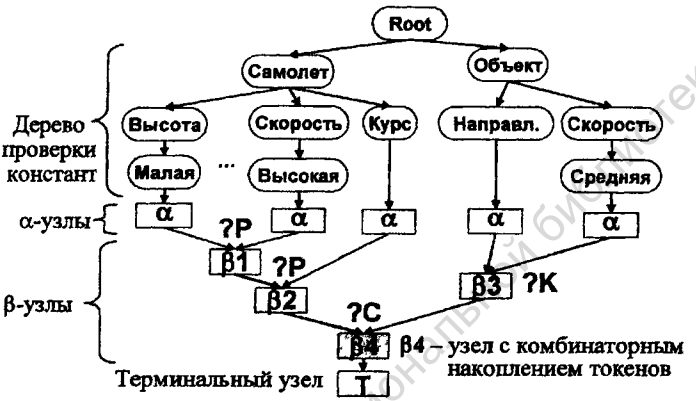

Рисунок 1 – Пример RETE-сети

мой архитектурой требуют существенных затрат на этапе компиляции, т.е. перехода от состава БЗ к конфигурации аппаратуры. Целесообразным подходом к организации средств аппаратной поддержки ПС для ИА являются ФОП, способные обрабатывать разные ПБЗ с определенной формой внутреннего представления.

В ряде архитектур ИА РВ допускается модификация/формирование БЗ в процессе работы агента (CIRCA, REX, и др.). Использование ФОП в таких системах предполагает решение задачи эффективной компиляции RETE-сети из исходной БЗ за относительно малое и предсказуемое время. Построение оптимальной RETE-сети в общем случае является NP-полной задачей, поэтому необходима разработка методов эффективной компиляции, исключающих комбинаторный перебор вариантов сети.

Во второй главе рассмотрены вопросы организации ФОП: построена виртуальная машина интерпретатора RETE-сети, разработан алгоритм обработки токенов, предложен и обоснован ряд модификаций RETE-сети, разработаны форматы представления элементов RETE-сети в памяти ФОП с учетом обработки неопределенной информации.

Основным препятствием аппаратной реализации RETE-сети является комбинаторное накопление токенов в β-памяти при увеличении глубины сети и количества объектов в БД. Это влечет необходимость использования динамических структур данных для организации памяти узлов и большие затраты на хранение и обработку вспомогательных данных. Аппаратная интерпретация в такой форме, как показывает анализ, не дает существенного выигрыша. Прямой переход к статиче-

ским структурам для классической RETE-сети приводит к резкому росту затрат памяти, определяемому выражением:

$$MEM_{класс} = \sum_{n=2}^{M}(1 + N \cdot n) + \sum_{k=2}^{P}\frac{kM}{P}\left(\frac{N}{k}\right)^{k}, \qquad (1)$$

где N – число объектов, P – число классов, M – количество УЭ в правиле. Наибольший вклад в рост объема памяти вносят β-узлы, выполняющие сопоставление предикатов по разным классам при наличии переменных одновременно в полях объекта и значения. Такие узлы названы «критическими».

Исходя из этого, в работе предложен ряд модификаций RETE-сети, позволяющих использовать фиксированные форматы в структуре β-узлов при умеренных затратах памяти:

1. Декомпозиция сложных УЭ правила во множества элементарных УЭ с фиксированным набором полей <*класс*, *объект*, *атрибут*, *значение*>.

2. Декомпозиция УЭ с двумя переменными во множества УЭ с фиксированным полем одной переменной.

3 Разбиение β-подсети на отдельные подсети для каждого класса.

4. Замена терминальных узлов на новый тип узлов – γ-узлы для систем, обрабатывающих отношения объектов разных классов.

5. Использование многовходовых β-узлов с учетом оптимального числа входов по критерию производительности.

6. Склеивание β-узлов разных продукций, выполняющих сопоставление по одним и тем же подмножествам предикатов.

Первые три модификации требуют однозначного задания состава УЭ и порядка расположения его полей. Структура сети, построенная с учетом данных требований, позволяет разбить терминальные узлы таким образом, чтобы результатом были обязательно все перечисления входящих токенов. Таким образом, каждой продукции соответствует несколько терминальных узлов вместо одного. При этом исключается необходимость искать выборки в критических узлах, а перечисление выполняется непосредственно при выдаче готового выходного токена в конфликтное множество. Такие новые узлы названы γ-узлами, поскольку они отличаются от классических хранением исходных множеств токенов. Многовходовые β-узлы и склеивание позволяют, помимо повышения производительности, существенно компенсировать рост числа β-узлов, обусловленный первыми тремя модификациями.

Затраты памяти для модифицированной RETE-сети оцениваются выражением:

$$MEM_{модиф} = PV_{ср}^{PVAR} + V_{ср}\sum_{j=1}^{P}V_{ср}^{VAR_j}, \qquad (2)$$

где P – число классов, Vcp – среднее число логических значений переменной, PVAR – число разных переменных в правиле, VAR_j – число разных переменных в классе j. Например, для БЗ из 1000 правил со следующими характеристиками: N = 100; P = 7; M = 30; PVAR=5; Vcp = 4; VAR_j = 3, RETE-сеть с фиксированной памятью в классическом варианте занимала бы $4*10^3$ Гслов (1), тогда как в модифицированной форме – порядка 10 МСлов (2).

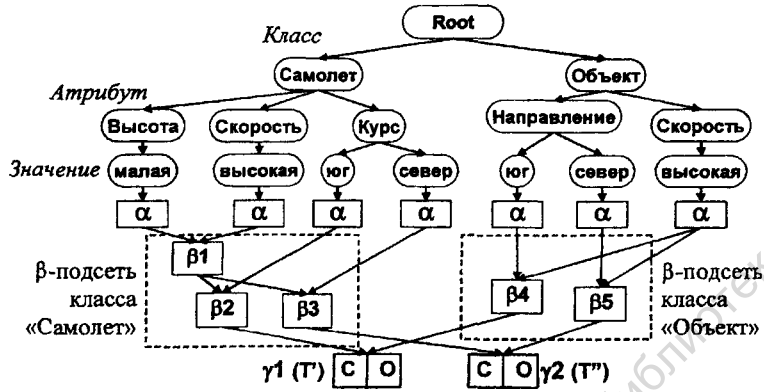

Рисунок 2 – Пример модифицированной RETE-сети

Пример модифицированной RETE-сети представлен на рис. 2. Количество узлов в нем выросло, но их структура существенно упрощена, что позволяет упростить архитектуру и повысить производительность процессора.

Проектирование ФОП аппаратной поддержки ПС предполагает на первом этапе построение виртуальной машины обработки RETE-сети как совокупности абстрактных элементов данных и укрупненных операций над ними.

ЭРП (факты) отображаются в виде WME = <Cls, Obj, Attr, Val>, где Cls – класс объекта, Obj – идентификатор объекта, Attr – атрибут, Val – значение. На вход RETE-сети подаются входные токены TKN_{in} = <TG, WME>, где TG – тэг, передающий знак токена (добавление/удаление).

Конфликтное множество CS представляет собой набор упорядоченных пар, каждая из которых содержит продукцию и список удовлетворяющих ей объектов: CS = < (PR_1, OL_1), (PR_2, OL_2), …, (PR_P, OL_P) >, где PR_1 … PR_P – продукции; OL_1 … OL_P – списки удовлетворяющих им объектов; P – количество продукций. При появлении в БД набора объектов, свойства которых удовлетворяют условию продукции, соответствующий набор вносится в конфликтное множество. Выходной токен RETE-сети содержит пару (PR_i, OL_i) конфликтного множества. Операции над конфликтным множеством можно формально записать следующим образом:

$$CS_{t+1} = CS_t \cup \{(PR_i, OL_i)\}, \text{ при положительном тэге токена,}$$

$$CS_{t+1} = CS_t \setminus \{(PR_i, OL_i)\}, \text{ при отрицательном тэге токена,}$$

где t – время до итерации; t+1 – время после итерации.

Терминальный узел хранит решения по продукции. На его вход поступает токен, содержащий полный набор ЭРП, удовлетворяющий условию продукции. На выходе формируется пара (PR_i, OL_i), где i – идентификатор продукции.

Базовой операцией алгоритма является операция сопоставления в β-узле:

$$Match\ (YM, TKN_{\beta IN}):\ \ \forall TKN_i \in YM,\ \ TKN_{\beta IN} \circ TKN_i,$$

где TKN_{IN} = < TG_{IN}, WME_{IN_1}, WME_{IN_2}, …, WME_{IN_L} > – входной токен, поступающий на вход X β-узла (в память XM); YM – память другого входа данного узла;

$a \circ b$ – операция сравнения значений переменной в токенах a и b. Результатом операции является один или несколько выходных токенов $TKN_{OUT} = <TKN_{IN}, TKN_{MT}>$, $TKN_{MT} \in XM$, где TKN_{MT} – найденный токен из памяти YM, удовлетворяющий условию сопоставления. Если тэг входного токена TG_{IN} положительный, в β-узле выполняется операция добавления токена к памяти XM:

$$ADD\ (XM, TKN_{IN}):\ XM = XM \cup \{TKN_{IN}\}.$$

Если тэг отрицательный, то выполняется удаление токена из памяти XM:

$$DEL\ (XM, TKN_{IN}):\ XM = XM \setminus \{TKN_{IN}\}.$$

При поступлении входного токена интерпретатор RETE-сети выполняет обход графа с добавлением или удалением соответствующих промежуточных токенов в β-памяти. Производительность интерпретатора определяется количеством входных токенов, обрабатываемых в единицу времени. Время обработки токена вычисляется по формуле $t_{TKN} = t_{const} + t_{\alpha} + n_S \cdot t_{\beta} + t_T$, где t_{const} – время обработки констант; t_{α} – время обработки α-узла; t_{β} – время обработки β-узла; t_T – время обработки терминального узла; n_S – количество слоев β-подсети. Время обработки β-узла определяет временную сложность алгоритма в целом.

Для представления RETE-сети в памяти ФОП в работе разработаны эффективные форматы указателей, подсети констант и слов α-, β и γ-памяти. Служебная информация (конец списка, число входов узла и т.п.) располагается в одном слове с указателем, что исключает дополнительные затраты на ее чтение и хранение. Общий формат сети включает дескриптор, определяющий параметры настройки ФОП на конкретную БЗ (длина сети, адрес массива γ-узлов, тип неопределенности, и пр.).

В третьей главе предложена и обоснована базовая архитектура ФОП, ориентированная на предложенную модифицированную форму RETE-сети, уточнен алгоритм работы интерпретатора, разработаны методы оптимизации и компиляции RETE-сети, рассмотрена методика проектирования устройств данного класса.

В соответствии с принятым подходом к организации ФОП, RETE-сеть представлена в виде структуры, загружаемой в его ОЗУ. Базовая архитектура ФОП, по-

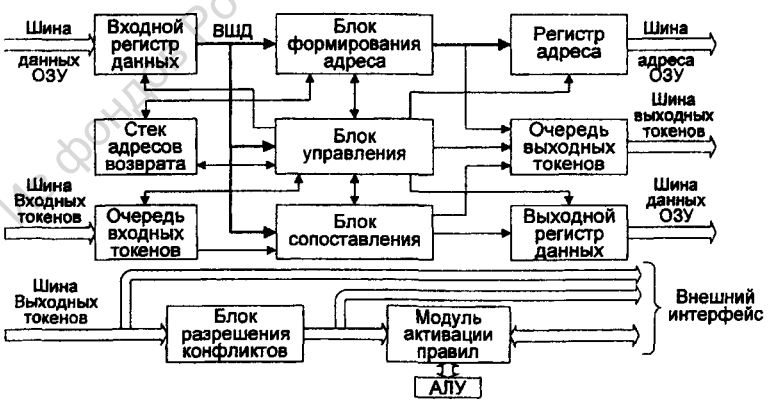

Рисунок 3 – Базовая архитектура ФОП

казанная на рис. 3, содержит аппаратный стек для реализации обхода сети в глубину, модуль сопоставления, используемый при обработке β-узлов, и блок управления, реализующий интерпретатор RETE-сети на микропрограммном уровне в терминах сигналов управления устройствами ФОП. Т.о., исключены этапы выборки и дешифрации команд, обычно имеющие место в процессорах. Ряд модулей разработан для вариантов четких и нечетких значений предикатов правил.

Поступление токена во входную очередь запускает обработку, в процессе которой производятся изменения в памяти узлов RETE-сети. В результате номер активируемой продукции, если таковая имеется, поступает в выходную очередь вместе с набором объектов, удовлетворяющих ее условию. Цикл работы аппаратного интерпретатора включает выборку очередного слова из памяти во входной регистр (либо запись слова), сопоставление его с текущим токеном, если это слово β-памяти, вычисление адреса следующего слова. Сопоставление выполняется параллельно с вычислением и выдачей следующего адреса. Таким образом, одноступенчатый конвейер обеспечивает почти полную загрузку канала обмена с памятью (80-90%), и дальнейшее дробление конвейера при данной организации памяти БЗ нецелесообразно. Память узлов располагается непосредственно в структуре RETE-сети вместе с указателями на последующие узлы, поэтому обращение к ней не требует отдельных операций адресной арифметики.

Ядро ФОП реализует основной этап вывода – сопоставление. Блок разрешения конфликтов и модуль активации правил реализуют соответствующие операции по алгоритмам, выбранным разработчиком.

Для предложенной модификации RETE-сети в работе обоснована целесообразность использования многовходовых β-узлов. Показано, что существует оптимальное число входов β-узла, при котором достигается максимальный прирост производительности по сравнению с двухвходовыми. Подсеть β-узлов с одним выходом рассматривается как структура, имеющая M слоев. При этом количество входов каждого промежуточного слоя равно числу выходов предыдущего:

(1): $1\times X(1,1) + 2\times X(2,1) + \ldots + N\times X(N,1) = N$

(2): $1\times X(1,2) + 2\times X(2,2) + \ldots + N\times X(N,2) = X(1,1) + X(2,1) + \ldots + X(N,1)$

...

(M): $1\times X(1,M) + 2\times X(2,M) + \ldots + N\times X(N,M) = X(1,M-1) + X(2,M-1) + \ldots + X(N,M-1)$

(M+1): $X(1,M) + X(2,M) + \ldots + X(N,M) = 1$

Зависимость среднего времени обработки β-узла от количества его входов выведена для уточненного алгоритма и названа ценой обработки узла:

$$T_n(n) = 19n - 7 + 2\frac{\sum_{m=2}^{n-1}\left((n-m)\sum_{k=1}^{m-1} k A_{m-1}^{k} P_{n-k-2}\right)}{P_{n-1}}, n \geq 2.$$

Для каждого слоя β-подсети суммарная цена равна:

$S1 = T(2)\times X(2,1) + T(3)\times X(3,1) + \ldots + T(N)\times X(N,1)$,

$S2 = T(2)\times X(2,2) + T(3)\times X(3,2) + \ldots + T(N)\times X(N,2)$,

...

$SM = T(2)\times X(2,M) + T(3)\times X(3,M) + \ldots + T(N)\times X(N,M)$.

Тогда задача минимизации общей суммарной цены, т.е. значения функции $S=S1 + S2 +\ldots + SM$, от NxM переменных $X(1,1),\ldots,X(N,M)$ является задачей целочисленного линейного программирования. Ее решение позволяет найти оптимальную по критерию времени обработки структуру β-подсети для любого числа входов.

Еще одна возможность оптимизации RETE-сети заключается в склеивании β-узлов из разных продукций, выполняющих сопоставление одних и тех же подмножеств УЭ. Нахождение оптимальных склеиваний требует в общем случае полного перебора и является NP-полной задачей. Для предложенной формы RETE-сети разработан способ склеивания, основанный на представлении продукций в виде строк двоичной матрицы, названной матрицей вхождений. С применением побитовых операций по транспонированной матрице вхождений находятся все непустые пересечения подмножеств УЭ за C_N^2 итераций, в то время как нахождение пересечений классическим методом потребовало бы экспоненциальной сложности. Найденные пересечения выбираются в порядке убывания степени и ранга (числа входов). При этом модификация матрицы позволяет отслеживать зависимые пересечения, понижать их ранг или исключать из рассмотрения.

На основе предложенных методов разработан и реализован алгоритм построения оптимальной RETE-сети. Экспериментально подтверждена его квадратичная временная сложность. Например, для количества входов сети порядка 100 на платформе PIII-700 он выполняется не более чем за 40 мс. (рис. 4). Такая невысокая и, что очень важно, предсказуемая сложность позволяет использовать данный алгоритм в системах с динамическим формированием БЗ.

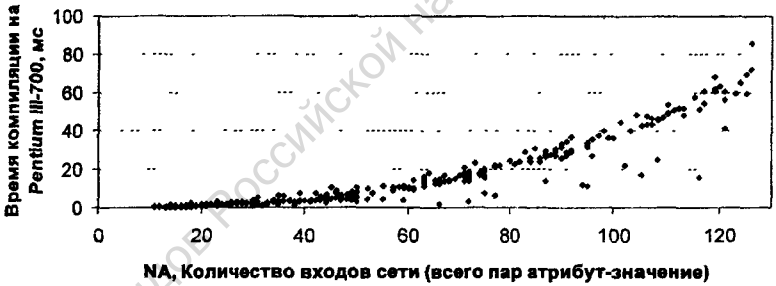

Рисунок 4 – Время компиляции RETE-сети

На рис. 5 отражены результаты экспериментального сравнения производительности ФОП при использовании одного метода оптимизации и обоих методов одновременно по сравнению с неоптимизированной RETE-сетью. Подтверждается теоретический расчет 18%-го прироста при использовании многовходовых β-узлов, а суммарный выигрыш в производительности с ростом объема БЗ стремится к величине порядка 40%. Кроме того, оба метода позволяют сократить объем памяти БЗ.

В работе предложена методика проектирования, базирующаяся на представлении ядра ФОП в виде готовой HDL-библиотеки мегафункций. Такой подход обеспечивает переносимость и масштабируемость. Проектировщик может выбирать ограничения на количество объектов, глубину сети, размеры полей токена и прочие параметры, влияющие на объем аппаратуры ФОП и памяти. Процесс осно-

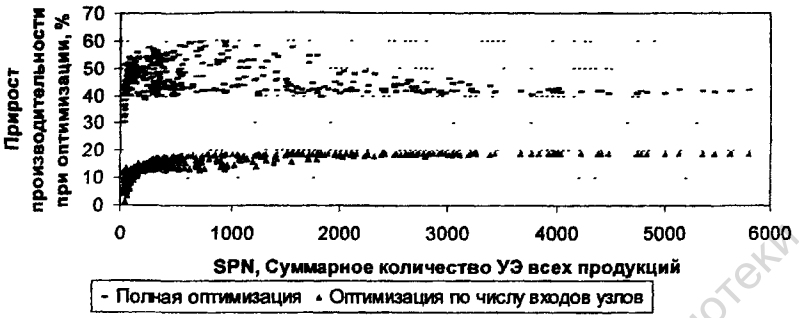

Рисунок 5 – Сравнение производительности при использовании оптимизации

ван на использовании инструментальной среды проектирования и включает решение задач: определения подмножества БЗ в архитектуре ИА, для которых разрабатываются средства аппаратной поддержки; преобразования правил к унифицированному виду; определения дополнительных аппаратных модулей для обработки неопределенных знаний и вспомогательных вычислительных алгоритмов; уточнения масштабируемых параметров базовой архитектуры; описания полной спецификации ФОП на HDL-языке (VHDL, Verilog); расчета параметров, влияющих на оптимальное число входов β-узлов; реализации алгоритма компиляции БЗ заданного типа в оптимальную RETE-сеть в формат ФОП.

В четвертой главе проведен анализ текущего состояния и перспектив в области ПЛИС, описан разработанный экспериментальный образец ФОП с сопутствующим ПО, и методика оценки его производительности.

ФОП с предложенной архитектурой был реализован на базе ПЛИС Altera ACEX 1K100. Ядро ФОП реализовано на VHDL с помощью пакета FPGA Advantage. В качестве основного сравниваемого варианта выбрана реализация ПБЗ в среде CLIPS, широко используемой для построения ИА РВ. С целью повышения достоверности экспериментальных оценок был написан программный интерпретатор RETE-сети на C++, функционально аналогичный ФОП. В CLIPS введен ряд средств, позволяющих точно измерить время этапа сопоставления фактов. Результаты сравнения производительности получены на основании измерений по 400 сгенерированным БЗ различного объема. Методика оценки включает ряд способов минимизации влияния побочных эффектов на точность измерений для программных реализаций (пятикратное повторение измерений с выбором лучшего результата, повтор эксперимента на разных ПЭВМ, и т.п.).

Для взаимодействия с хост-машиной и экспериментальной оценки производительности написан ряд программных модулей, включающих генераторы БЗ и БД, низкоуровневый программный интерпретатор RETE-сети, преобразователи форматов (в том числе в CLIPS), загрузчики БЗ и токенов, дополненный CLIPS, и т.п.

С точки зрения достоверности, важным результатом является то, что с ростом объема БЗ время обработки разными программными интерпретаторами приближается к одной и той же величине (рис. 6), что говорит об их одинаковой вы-

числительной сложности. Другим важным результатом является то, что время обработки одного токена и всей RETE-сети в целом является предсказуемой величиной и практически линейно зависит от параметров БЗ (рис. 7). Это позволяет агенту планировать время обработки в соответствии с текущим состоянием.

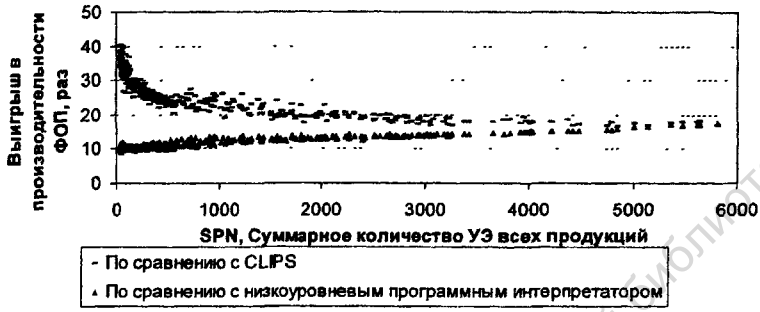

Рисунок 6 – Выигрыш в производительности

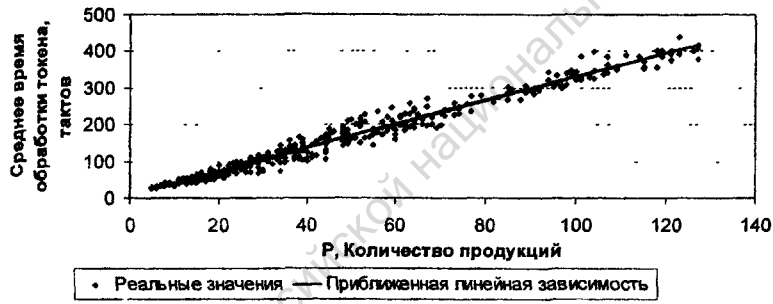

Рисунок 7 – Время обработки токена

В заключении сформулированы основные научные и практические результаты, обсуждаются перспективные направления дальнейших исследований.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ РАБОТЫ

В диссертации решена задача организации и проектирования функционально-ориентированных процессоров обработки продукционных баз знаний на основе RETE-сети. При этом получены следующие научные и практические результаты:

1. Виртуальная машина обработки RETE-сетей, представляющая собой совокупность абстрактных элементов данных и укрупненных операций над ними. Виртуальная машина позволяет систематизировать процесс проектирования ФОП.

2. Модифицированная форма RETE-сети, ориентированная на использование статических структур данных, и позволяющая существенно упростить архитектуру и повысить производительность ФОП. На основе данной формы разработан эффективный формат внутреннего представления RETE-сети в памяти ФОП для случаев четких и нечетких значений предикатов, а также различной разрядности памяти.

3. Базовая архитектура ФОП и варианты структур основных функциональных узлов, ориентированные на предложенный способ представления RETE-сети и учитывающие представление и обработку в ПБЗ неопределенной информации.

4. Метод оптимизации RETE-сети за счет использования многовходовых β-узлов. Задача нахождения оптимальной структуры β-подсети с любым числом входов поставлена и решена в терминах целочисленного линейного программирования. Разработаны способы оценки эффективности данного метода оптимизации.

5. Метод и алгоритм быстрой компиляции БЗ в модифицированную RETE-сеть с учетом склеивания β-узлов, отличающийся использованием матричных и побитовых преобразований над антецедентами правил. Метод исключает комбинаторный перебор конфигураций сети и обеспечивает предсказуемое время ее построения с алгоритмической сложностью $O(N^2)$. Это дает возможность использовать ФОП в системах с динамическим формированием ПБЗ.

6. Методика проектирования ФОП аппаратной поддержки ПС на основе RETE-сети, базирующаяся на представлении ядра в виде VHDL-библиотеки мегафункций, что обеспечивает переносимость и масштабируемость. Уточнен состав инструментальной среды проектирования, позволяющей автоматизировать процесс проектирования устройств данного класса.

7. Пакет VHDL-модулей масштабируемого ядра ФОП обработки ПБЗ на основе RETE-сети. Данный пакет позволяет проектировать подобные ФОП с учетом различных требований и ограничений по параметрам обрабатываемых БЗ, таких как максимальное число объектов, максимальная глубина RETE-сети, ширина входного токена, и т.п.

8. Действующий образец ФОП на основе ПЛИС, позволяющий оценить эффективность предложенного подхода к построению средств аппаратной поддержки ПС. Предложена методика экспериментальной оценки производительности ФОП в сравнении с программными интерпретаторами RETE-сети, учитывающая различные подходы к их построению и позволяющая минимизировать влияние побочных эффектов на точность измерений.

9. Программное обеспечение экспериментального комплекса, включающее: генераторы БЗ и БД, программный интерпретатор RETE-сети, компилятор БЗ в оптимизированную RETE-сеть, конверторы форматов (в том числе в CLIPS), программы взаимодействия с ФОП со сбором статистики, тестовые, и другие вспомогательные модули. Разработанное ПО позволяет отладить спроектированный ФОП и собрать статистику для экспериментальной оценки его эффективности. Оно также может использоваться как часть инструментальной среды проектирования.

ПУБЛИКАЦИИ ПО ТЕМЕ ДИССЕРТАЦИИ

1. Методы и средства построения интеллектуальных агентов реального времени / В.В. Денисов, Г.Г. Колосов, М.Г. Пантелеев, Д.В Пузанков // Труды 7-й нац. конф. по искусств. интеллекту с междунар. уч. КИИ'2000, г. Переславль-Залесский. – М.: Изд-во Физ.-мат. лит., 2000. – Т.2. – С. 805–813.

2. Пантелеев М.Г. Организация и проектирование средств аппаратной поддержки интеллектуальных агентов / М.Г. Пантелеев, В.В. Денисов, Г.Г. Колосов // Интеллектуальные системы (INTELS'2000): Труды 4-го междунар. симп. – М.: РУСАКИ, 2000. – С. 188–190.

3 Пантелеев М.Г. Процессоры аппаратной поддержки интеллектуальных агентов реального времени: разработка и реализация / М.Г. Пантелеев, В. В. Денисов, Г.Г. Колосов // Труды IV Междунар. конф. по мягким вычислениям и измерениям SCM'01. – СПб: СПбГЭТУ, 2001. – Т. 2. – С. 119–123.

4. Процессоры аппаратной поддержки интеллектуальных агентов реального времени: организация, проектирование, реализация / В.В. Денисов, Г.Г. Колосов, М Г. Пантелеев, Д.В. Пузанков // Труды междунар. конгресса Искусственный Интеллект в XXI веке ICAI'01. – М.: Изд-во Физ.-мат. лит., 2001. – С. 273–280.

5. Пантелеев М.Г Архитектура процессора логического вывода для продукционных БЗ на основе RETE-сети / М.Г. Пантелеев, Г.Г Колосов // Изв. СПбГЭТУ "ЛЭТИ".– СПб.: Изд-во СПбГЭТУ, 2002г. – Вып. 2. – С. 35–43.

6 Проектирование и реализация процессоров аппаратной поддержки интеллектуальных агентов реального времени / В.В. Денисов, Г.Г. Колосов, М.Г. Пантелеев, Д.В. Пузанков // Материалы междунар. науч.-техн. конф «СуперЭВМ и многопроцессорные вычислительные системы (MCS'2002). – Таганрог: Изд-во ТРТУ, 2002. – С. 216–220.

7. Проектирование и реализация средств аппаратной поддержки интеллектуальных агентов реального времени / Д.В. Пузанков, М.Г. Пантелеев, Г.Г. Колосов, И.Б. Говорухин // Труды междунар. конф. «Искусственные интеллектуальные системы» (IEEE ICAIS'02) и «Интеллектуальные САПР» (CAD-2002). – М.: Изд-во Физ.-мат. лит., 2002. – С. 236–243.

8. Design and Implementation of Hardware for Real-Time Intelligent Agents (Проектирование и реализация средств аппаратной поддержки интеллектуальных агентов реального времени) / M.G. Panteleev, D.V. Puzankov, G.G. Kolosov, I.B. Govorukhin. // Proceedings of 2002 IEEE Intl. Conference on Artificial Intelligence Systems (ICAIS 2002). – California, 2002. – P. 6–11.

9. Колосов Г. Г. Процессор аппаратной поддержки продукционных систем на основе RETE-сети / Г.Г. Колосов // Труды 5-го междунар. симп. «Интеллектуальные системы» (INTELS'2002). – М.: МГТУ им. Н. Э. Баумана, 2002. – С. 306–308.

10. Организация и проектирование функционально-ориентированных процессоров аппаратной поддержки продукционных баз знаний / Д.В. Пузанков, М.Г Пантелеев, В.В. Денисов, Г.Г. Колосов // Изв. ВУЗов. Приборостроение. – СПб., 2003. – Т.46. №2. – С. 18–23.

11 Пантелеев М. Г Проектирование процессоров обработки продукционных баз знаний на основе RETE-сети / М.Г. Пантелеев, Г.Г. Колосов // Материалы меж-

дунар. науч.-техн. конф. «Интеллектуальные и многопроцессорные системы» ИМС'2003. – Таганрог: Изд-во ТРТУ, 2003. – Т.2. – С. 102–105.

12. Пантелеев М.Г. Проектирование процессоров обработки продукционных баз знаний на основе RETE-сети / М.Г. Пантелеев, Г.Г. Колосов // Искусственный интеллект. – 2003. – №3. – С. 465–473.

13. Пантелеев М.Г. Виртуальная машина интерпретатора RETE-сетей в задачах проектирования функционально-ориентированных процессоров / М.Г. Пантелеев, Г.Г. Колосов // Материалы 5-й междунар. науч.-техн. конф. – Таганрог: Изд-во ТРТУ, 2004. – Т.1. – С. 292–296.

14. Пантелеев М.Г. Виртуальная машина интерпретатора RETE-сетей в задачах проектирования функционально-ориентированных процессоров / М.Г. Пантелеев, Г.Г. Колосов // Искусственный интеллект. – 2004. – №4. – С. 726–732.

15. А. с. 25951 РФ, МКИ G 06F 7/00. Функционально-ориентированный процессор обработки продукционных знаний / М.Г. Пантелеев, В.В. Денисов, Г.Г. Колосов (РФ). – № 2002115379/20; заявл. 11.06.02; опубл. 27.10.02, Бюл. № 30. – 3 с.: ил.

Подписано в печать 06.09.05. Формат 60*84 1/16.
Бумага офсетная. Печать офсетная. Печ. л. 1,0.
Тираж 100 экз. Заказ 73.

Отпечатано с готового оригинал-макета
в типографии Издательства СПбГЭТУ "ЛЭТИ"

Издательство СПбГЭТУ "ЛЭТИ"
197376, С.-Петербург, ул. Проф. Попова, 5