

Е.А. КАМЕНИЮКА

PRUS – СРЕДСТВО МОДЕЛИРОВАНИЯ СИСТЕМ БУЛЕВЫХ УРАВНЕНИЙ

Предлагается высокопроизводительная технология решения булевых уравнений, основанная на использовании сферического однобитного мультипроцессора PRUS (Programmable Unlimited Systems), реализуемого на кристалле ASIC. Он позволяет осуществлять параллельную, последовательную и конвейерную обработку булевых уравнений, записанных в базе операций AND, OR, NOT, XOR. Мультипроцессор экономичен в аппаратном исполнении – для обработки системы уравнений, насчитывающей 20 миллионов вентилей, необходимо иметь всего 256 Мбайт оперативной памяти.

1. Актуальность создания мультипроцессора

Чтобы решить систему булевых уравнений, содержащую порядка миллиона линий или эквивалентных вентилей, существует несколько практически ориентированных технологий.

1. Можно использовать персональный компьютер или рабочую станцию на основе микропроцессора фирмы Intel. Здесь каждое уравнение будет решаться программным путем и последовательно, поскольку существует только один процессор, хотя и достаточно мощный. Стоимость решения проблемы, а также временные затраты очень высоки.

2. Реализовать специализированный параллельный процессор на основе PLD. В данном случае высокий параллелизм обработки уравнений компенсирует сравнительно низкую (по сравнению с CPU) тактовую частоту. Такое схемотехническое решение, к тому же с возможностью репрограммирования, является по производительности абсолютно выигрышным. Но здесь существенным недостатком будет выступать отсутствие гибкости, присущей программным методам решения уравнений. Кроме того, реализация системы на кристалле PLD имеет высокую стоимость, если объем будущих продаж составит десятки тысяч штук.

3. Третье решение связано с объединением достоинств CPU, PLD и ASIC, таких как:

- гибкость программирования системы уравнений, которая позволяет оперативно корректировать спецификацию в виде исходных кодов;
- минимально возможная мощность системы команд, приводящая к простым схемотехническим решениям аппаратурной реализации процессора;
- распараллеливание процесса решения уравнений, благодаря идеологии PLD, но с элементами CPU, что означает иметь много простых (однобитовых) взаимосвязанных между собой процессоров с собственной системой команд для параллельного программирования;

– имплементация мультипроцессора в кристалл ASIC, что позволит иметь максимально возможную тактовую частоту, минимальную стоимость одной микросхемы при большом объеме (более 10 000) выпуска продукции, низкое энергопотребление;

– конвейеризация решения системы уравнений есть исключительное свойство, присущее только для мультипроцессорной системы, где конвейерная обработка данных есть один из основных режимов наряду с параллельным и последовательным способами.

Таким образом, необходимость появления мультипроцессора для решения булевых уравнений со свойствами, упомянутыми выше, обусловлена существующими тенденциями на рынке электронных технологий. Что имеется здесь ввиду? В ближайшем будущем актуальные и вычислительно сложные проблемы должны будут решаться специализированными вычислительными микро- и мульти-процессорами. На рынке микроэлектроники возрастает показатель специализированной ориентации на решение конкретных задач – customization. Когда возможности универсальных компьютеров начинают превышать потребности данной области рынка, конкуренция смещает акценты с повышения общей производительности на усовершенствование свойств, за которые пользователь готов платить дополнительные деньги. Это – не только надежность и удобство, но и быстрдействие решения сложной проблемы, заказные свойства, энергосбережение, автономность, миниатюризация, гибкость и репрограммируемость.

2. Постановка задач проектирования мультипроцессора PRUS

PRUS (Programmable Unlimited Systems) есть ad hoc технология и специализированное вычислительное устройство, реализуемое в кристалле ASIC, для быстрого решения булевых уравнений. Архитектура PRUS представляет собой матрицу параллельных процессоров, каждый из которых связан с восемью другими линиями передачи данных. Структура PRUS и модель обработки уравнений разработаны доктором Stanley Hyduke и носят название “Compiler Synchronized Parallel-processor Network-based Logic Device” – сеть параллельных компилятивно синхронизированных процессоров. PRUS-технология и логический процессор, имеющий порядка 90 вентиляей, защищены патентами США, что подтверждает оригинальность архитектурных и процессных решений, ориентированных на эмуляцию системы булевых уравнений большой размерности.

Проектирование на основе PRUS подобно классической схеме (design flow) за исключением того, что стадия place and route заменяется фазой распределения булевых уравнений между сотнями (тысячами) логических процессоров, работающих параллельно. Boolean Equation Compiler™ (BECOM) обеспечивает размещение уравнений по процессорам, задает время формирования решения на выходе каждого из них, а также планирует передачу полученных результатов другому процессору.

PRUS есть эффективная сеть процессоров, которая обрабатывает систему логических уравнений и обеспечивает обмен данными между компонентами сети в процессе их решения. Простая схемотехника каждого процессора, требующая 48 бит памяти для эмуляции двухвходового вентиля, позволяет обрабатывать сверхбольшие проекты, насчитывающие более 20 миллионов вентилях, затрачивая на это 256 Мбайт оперативной памяти. При этом ее затраты на описание триггера находятся в интервале 48 - 64 бит. Триггерные структуры могут располагаться в любой части комбинационной схемы, обеспечивая полную свободу проектирования.

Базовый вариант процессора PRUS может быть синтезирован на 90 вентилях, что дает возможность легко имплементировать, например, сеть, содержащую 4096 вычислителей, в ASIC, используя современную силиконовую технологию. Учитывая, что затраты памяти для эмуляции вентилях и триггеров весьма незначительны, то PRUS может представлять интерес для проектирования систем управления в таких областях человеческой деятельности, как: индустрия, медицина, защита информации, геология, прогнозирование погоды, искусственный интеллект, космонавтика. Это представляет особый интерес для цифровой обработки данных, распознавания образов, криптоанализа. Одним из основных приложений PRUS в EDA (Electronic Design Automation) технологиях является эмуляция больших проектов, имплементируемых в ASICs и FPGA. Учитывая вышеизложенное, далее необходимо сформулировать проблему, цель и задачи исследования с учетом основного предназначения мультипроцессора.

Сферический мультипроцессор PRUS предназначен для решения системы булевых уравнений большой размерности за приемлемое для пользователя время.

Проблема обработки булевых уравнений, насчитывающих порядка миллиона вентилях, связана с организацией вычислений на мультипроцессорной системе путем введения параллелизма и конвейеризации. Это позволяет уменьшить время получения решения в сотни раз по сравнению с последовательной обработкой данных.

Цель исследования – разработать однобитовый мультипроцессор, имплементируемый в ASIC, с матрично-сферической организацией взаимных связей и минимальным набором команд для повышения быстродействия решения систем булевых уравнений большой размерности в сотни раз.

Задачи исследования:

- 1) Разработать структуру сферического мультипроцессора PRUS, имплементируемого в кристалл ASIC и ориентированного на решение системы булевых уравнений большой размерности;
- 2) Спроектировать структуру секвенсера, как базовой ячейки PRUS, предназначенной для эмуляции примитивного элемента, выполняющего одну из следующих функций: AND, OR, NOT, XOR;

3) Разработать систему команд, достаточную для параллельного программирования и решения булевых уравнений на основе мультипроцессора PRUS;

4) Реализовать компилятор для преобразования HDL-описания системы булевых уравнений в язык машинных кодов мультипроцессора PRUS;

5) Разработать программный модуль оптимизации распределения системы булевых уравнений между секвенсерами PRUS, в целях максимального использования параллельного и конвейерного режимов работы мультипроцессора для получения решения за минимально возможное время;

6) Выполнить верификацию и тестирование мультипроцессора PRUS на доказательных примерах систем булевых уравнений, представляющих промышленный и научный интерес в целях оценивания быстродействия и аппаратных затрат на реализацию проекта.

3. Структура и функции мультипроцессора PRUS

Мультипроцессор PRUS в качестве основного составляющего компонента включает секвенсер, как hardware модуль, содержащий однобитовый процессор (Single Bit Processor – SBP) и оболочку (wrapper) в виде: памяти для хранения программного кода и данных, а также декодеров и мультиплексора. Собственно процессор представлен в виде модуля для реализации операций {AND, OR, NOT, XOR}. Далее мы будем считать секвенсер и процессор синонимами, если это не приведет к неверному толкованию излагаемого материала.

Для оптимизации распределения булевых уравнений по частям мультипроцессора необходимо иметь представление о его топологической структуре. Она отчасти подобна карте Карно (Karnaugh), поскольку каждый секвенсер имеет восемь соседей (в карте Карно – 4). Планарная структура PRUS представлена на рис. 1. Такое представление удобно для решения

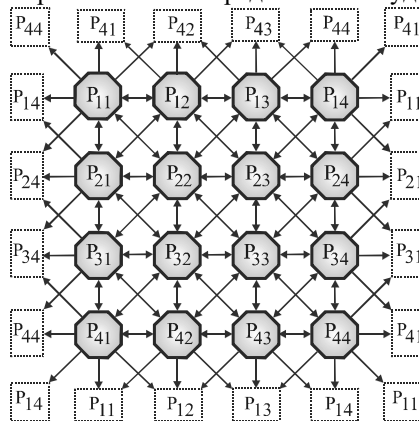


Рис. 1. Планарная структура процессора PRUS

оптимизационных задач на плоскости, к которым можно отнести: размещение булевых уравнений по процессорам, минимизацию функциональных связей между секвенсерами при эмуляции уравнений.

4. Архитектура секвенсера PRUS

Основным компонентом мультипроцессора является секвенсер, структура которого представлена на рис. 2.

Секвенсер эмулирует булевы уравнения. Все секвенсеры в сети функционируют синхронно и параллельно, обрабатывая однобитовый поток логических данных путем выполнения операций AND, OR, NOT, XOR. Результат моделирования сохраняется в памяти данных. В целях синхронизации всей вычислительной системы программная память каждого секвенсера в сети адресуется общим счетчиком адресов команд.

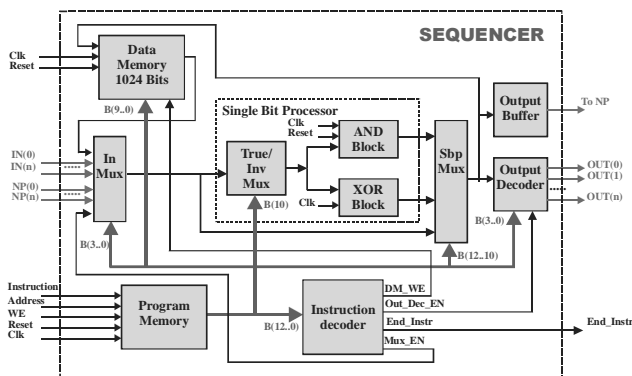


Рис. 2. Функциональная структура секвенсера

Program Memory предназначена для хранения программы моделирования уравнений в течение всего цикла, необходимого для их решения. Очередная команда выбирается с помощью адреса, формируемого на соответствующей шине. Код команды управляет работой всех блоков секвенсера.

Data Memory используется для хранения промежуточных результатов обработки уравнений на одном секвенсере. Здесь 10 младших разрядов командного слова используются как адрес памяти данных, если сигнал Data Memory Write Enable = 1. При таком количестве разрядов возможна адресация к 1024 ячейкам памяти данных.

InMux – используется для ввода внешних данных в секвенсер. Он может коммутировать один из восьми соседних процессоров ($Mux_En = 1$) или выбирать информацию из памяти данных.

Single Bit Processor (SBP) – выполняет две параллельные операции AND и OR над прямым или инверсным значением входной переменной.

SBP Mux – коммутирует выходы одного из четырех компонентов: AND block; инвертор содержимого AND block (operation ANDI); XOR block, InMux для передачи их на выход секвенсера. При этом коммутация блока InMux означает режим by pass для данного секвенсера в целях передачи информации ближайшему соседу.

Output Buffer – синхронизирует работу двух соседних процессоров и используется для передачи результирующей информации с выхода к соседу, после чего последний продолжает дальнейшую обработку уравнения.

Output Decoder – используется для выбора одного из n внешних выходов мультипроцессора PRUS в целях последующей передачи информации о решении системы уравнений.

Instruction Decoder – используется для декодирования адресов операндов и специальных команд, типа NOP, END. Он обеспечивает подачу управляющих сигналов для инициирования работы блоков: Output Decoder, Data Memory, Input Multiplexer и Single Bit Processor. Кроме того, он активизирует сигнал END_INSTR, который управляет работой программного счетчика адресов. При $END_INSTR = 1$ он обнуляется.

Каждый секвенсер имеет собственную программную память для хранения двоичных инструкций моделирования логических уравнений и память данных для сохранения результатов их решения. Во время программирования системы уравнений в память каждого секвенсера загружается собственное множество машинных команд, предназначенных для обработки логических уравнений. Поскольку все секвенсеры и соответствующие им памяти управляются одним общим регистром адреса, то они функционируют и обрабатывают логические уравнения синхронно и параллельно (рис. 3).

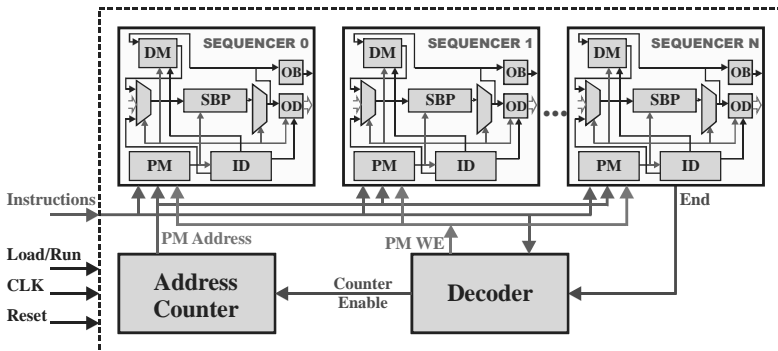


Рис. 3. Соединение секвенсеров

Счетчик адреса инкрементирует свое значение под управлением сигналов синхронизации системы CLK и Counter Enable – выходной сигнал декодера (Decoder), который равен 1 при достижении последней команды в программной памяти. Счетчик адреса может выбирать из памяти только последовательно идущие команды в программной памяти.

5. Структура и функции SBP

Однобитовый процессор SBP – основной компонент секвенсера. Его базовая структура представлена на рис. 4.

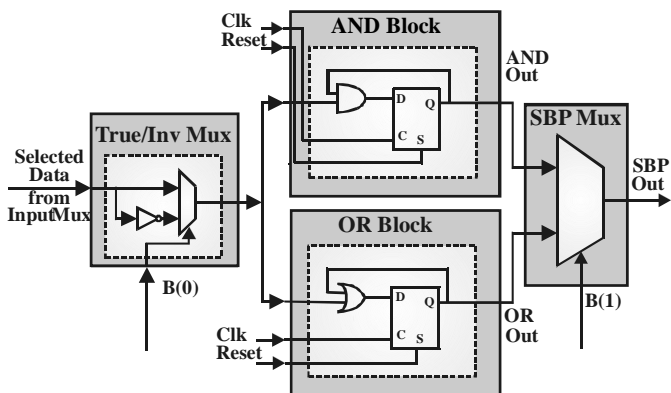


Рис. 4. Функциональная структура SBP

SBP предназначен для выполнения логических операций AND, OR, NOT, которых достаточно для решения любых сколь угодно сложных уравнений. В состав SBP входят также однобитовые OR- и AND-регистры для хранения промежуточных данных. SBP выполняет две операции над содержимым входа и регистра в параллельном режиме. Для выполнения операции AND (OR) на любом из блоков необходимо 2 временных такта: на первом выполняется загрузка первого операнда в однобитовый регистр, второй такт формирует результат на выходе D-триггера, как функцию, где X – вход данных для AND- или OR-блока. Один из полученных результатов передается на выход процессора с помощью мультиплексора SBP Mux. Функция мультиплексора True/Inv Mux заключается в передаче значения логической переменной в прямом или инверсном коде, что необходимо для обработки функций И-НЕ, ИЛИ-НЕ, а также простого инвертора или повторителя. Управление мультиплексором осуществляется одним битом командного слова. Данные для SBP могут быть доставлены со входа мультиплексора, от одного из соседних процессоров или из памяти данных. Что касается результата обработки уравнения, то он может быть передан в

память данных, соседнему процессору или на внешний выход мультипроцессора.

Базовая структура (см. рис.4) может быть модифицирована путем замены одного из функциональных блоков. Поскольку правила де Моргана позволяют **преобразовать** любую функцию к базису И-НЕ или ИЛИ-НЕ, то, естественно, что один из блоков AND- или OR-блок можно считать избыточным. Поэтому OR-блок предлагается исключить из SBP, поставив вместо него, модуль, выполняющий логическую операцию XOR logic operation (рис.5). Какие преимущества это дает? В базовой схеме SBP (см. рис.4) AND(OR)-регистр устанавливается в 1(0) перед началом обработки уравнения и он будет иметь в первом такте 1, если на вход поступит сигнал, равный 1. В модифицированной структуре SBP (см. рис.5) регистр XOR не требует первоначальной установки. Его инициализация выполняется единичным сигналом старшего бита командного слова. Обратная связь с выхода XOR на вход регистра сдвига позволяет вычислять XOR-операцию много раз. Такое расширение функций SBP оказывает положительное влияние на быстрдействие обработки булевых уравнений, имеющих операции XOR.

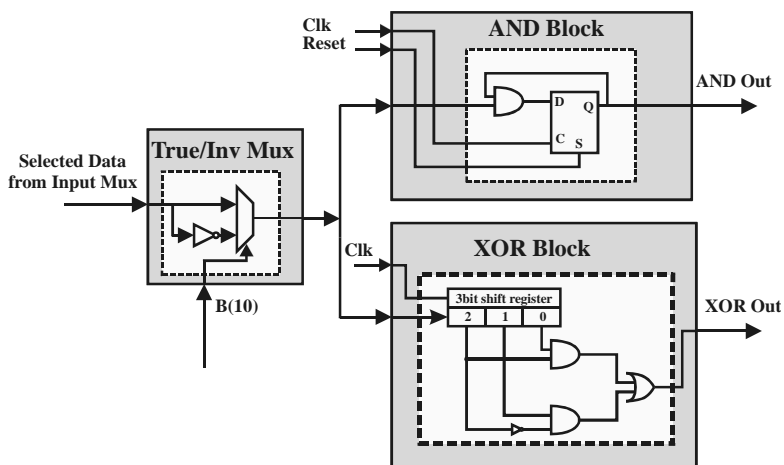


Рис. 5. Модифицированная структура SBP

6. Заключение

В качестве основных результатов, фигурирующих в предложенной PRUS-технологии, следует выделить два основных компонента: научную и практическую значимость. Первый компонент – научная новизна – определяется формулировкой следующих тезисов:

1. Предложена новая высокопроизводительная PRUS технология параллельного решения систем булевых уравнений, насчитывающих сотни миллионов эквивалентных вентиляей.

2. Разработана архитектура сферического мультипроцессора, где каждый однобитный процессор имеет непосредственные связи с восемью другими.

3. Представлена схемотехнически простая структура однобитного процессорного элемента, ориентированная на выполнение операций AND, OR, NOT, XOR, а также секвенсера, в состав которого входит память программы и данных, а также соответствующие схемы управления.

4. Предложена модель организации вычислительных процессов, заключающаяся в равномерном распределении системы булевых уравнений между вычислительными ресурсами мультипроцессора на основе применения метода половинного деления.

Литература: 1. *Design Automation Conference. Proceedings.* San Diego, California. 2004. 970p. 2. *Bass M.J. and Cristensen C.M.* The future of the Microprocessor Business.- IEEE Spectrum, April, 2002, P. 34-39. 3. *Don Edenfeld, Andrew B Kahng, Mike Rodgers, and Yervant Zorian,* "2003 Technology Roadmap for Semiconductors", Computer, January 2004, pp. 47-56. 4. *Nick Tredennock, Brion Shimamoto.* "Go Reconfigure". IEEE Spectrum. December 2003. P. 37-40. 5. *B. Gupta and M. Borgatti.* Different Approaches to Add Reconfigurability in a SoC Architecture // Design Automation and Test in Eurpe (DATE 03).- 2003.- P. 398-402. 6. *M. Borgatti, L. Calm, G. De Sandre, B. Forst, D. Iezzi, F. Lertora, G. Muzzi, M. Pasotti, M. Poles, P.L. Rolandi* (STMicroelectronics) A Reconfigurable Signal Processing IC with Embedded FPGA and Multi-Port Flash Memory // Design Automation Conference (DAC 03).- 2003.- P. 691-696. 7. *Linda Geppert.* "The New Indelible Memories". IEEE Spectrum. March 2003. P. 49-54. 8. UML-SOC 2004. UML for SOC Design. DAC 2004. Workshop. 58 p. 9. *Nikolaos S. Voros, Luis Sanchez, Alejandro Alonso, Alexios N. Birbas, Michael Birbas, Ahmed Jerraya.* Hardware-Software Co-Design of Complex Embedded Systems. Design Automation for Embedded Systems.- Boston: Kluwer Academic Publishers.- 2003.- P. 5-34. 10. *Yoo S. and Jerraya A.* Introduction to Hardware Abstraction Layers for SoC // Design Automation and Test in Europe (DATE 03).-2003.-P. 336-338. 11. *Vercauteren,S., J.V.D. Steen, and D Verkest.* Combining Software Synthesis and Hardware-Software Interface Generation to Meet Hard Real-Time Constraints.- DATE'99.- 1999.- P. 177-187. 12. *Boehm, B.* A Spiral model of Software Development and Enhancement. Science of Computer Programming.- Vol. 8.- 1988.- P. 231-274. 13. *Steve Farber.* ARM System-on-Chip architecture. Addison - Wesley. 2000. 420p. 14. *David Seal.* Architecture Reference Manual. Addison - Wesley. 2001. 560p. 15. *Active-HDL User's Guid.* Second Edition.- Copyright.- Aldec Inc.- 2003.- 213p. 16. *Samir Palnitkar.* Verilog HDL. A Guide to digital design and synthesis. Sunsoft Press. A prentice Hall Title. 2002. 396p.

Поступила в редакцию 16.06.2005