

Исследование работы процессора сопоставления параллельной потоковой вычислительной системы «Буран»

Н.Н. Левченко, А.С. Окунев, Д.Е. Яхонтов

Институт проблем проектирования в микроэлектронике РАН, nick@ippm.ru, oku@ippm.ru, yahontov@burcom.ru

Аннотация — В статье рассматривается архитектура параллельной потоковой вычислительной системы «Буран», описана работа одного из основных модулей системы – процессора сопоставления. Описывается конвейерная работа процессора сопоставления, приводятся различные варианты построения такого конвейера и его оптимизации. Приведена оценка увеличения пропускной способности, связанной с введением конвейера.

Ключевые слова — параллельная потоковая вычислительная система, процессор сопоставления, конвейерная обработка, пропускная способность.

I. ВВЕДЕНИЕ

Одним из основных методов увеличения производительности вычислительных средств является метод распараллеливания вычислительных процессов, так как вычислительная производительность одного ядра на используемой элементной базе близка к пределу. На аппаратном уровне основным методом повышения производительности вычислительных систем в настоящее время является увеличение числа процессоров в системе. Однако существенным фактом является то обстоятельство, что для большинства задач с увеличением числа процессоров вычислительной системы падает реальная производительность каждого процессора. К причинам падения реальной производительности многопроцессорных систем можно отнести следующие:

- высокие накладные расходы на организацию параллельных вычислений;
- низкую пропускную способность подсистем оперативной памяти при нелинейной организации данных;
- необходимость синхронизации вычислительных процессов по данным.

Вследствие вышесказанного представляют интерес принципы построения вычислительной системы, архитектура которой основывается на нетрадиционной модели вычислений, отличной от общепринятой – фон-неймановской. В настоящей работе описана перспективная гибридная архитектура вычислительной системы, сочетающая как элементы архитектуры потока данных, так и традиционной фон-неймановской архитектуры – модель вычислений потока данных с динамически формируемым контекстом.

II. ОПИСАНИЕ АРХИТЕКТУРЫ ПАРАЛЛЕЛЬНОЙ ПОТОВОЙ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ

В рамках модели вычислений потока данных задача представляется в виде вычислительного графа. Граф состоит из узлов (каждый узел имеет входы и выходы) и ветвей, которые отражают зависимость узлов по данным. В узле графа выполняется программа обработки поступивших на него операндов. Результат работы программы узла направляется по ветвям графа к следующим узлам. С каждым узлом связана программа, которая называется программой узла.

Сообщение, передаваемое из узла в узел, называется токеном. Токен представляет собой структуру, в состав которой входят поле данных с той информацией, которую токен поставляет в узел и ключ, определяющий узел назначения. Ключ состоит из двух частей: указатель на программу узла и контекст – поле, однозначно определяющее конкретный экземпляр данного узла на вычислительном графе. Результатом выполнения программы узла является посылка одного или многих токенов в другие узлы, либо посылка токенов с результатами вычислений на хост-машину.

Запуск программы узла осуществляется по готовности всех ее входных данных.

Параллельная потоковая вычислительная система (ППВС) «Буран» состоит из вычислительных модулей, которые соединяются между собой коммуникационной сетью (рис. 1).

Вычислительный модуль состоит из набора процессоров сопоставления (ПС), набора исполнительных устройств (ИУ), коммутатора токенов и коммутатора пакетов.

Каждое ИУ имеет в своем составе память команд, в которой хранятся программы всех узлов задачи. Поток программа скопирована в машине столько раз, сколько имеется исполнительных устройств. Это позволяет, во-первых, запускать программу узла на любом ИУ и, во-вторых, выполнять несколько «экземпляров» программы узла одновременно. Каждое ИУ содержит также память констант, в которой записаны все константы потоковой программы. Одним из основных преимуществ рассматриваемой архитектуры является динамически формируемый контекст, то есть возможность произвольным образом задавать контекст вы-

ходных токенов в программе узла, что дает программисту новые возможности по управлению вычислительным процессом.

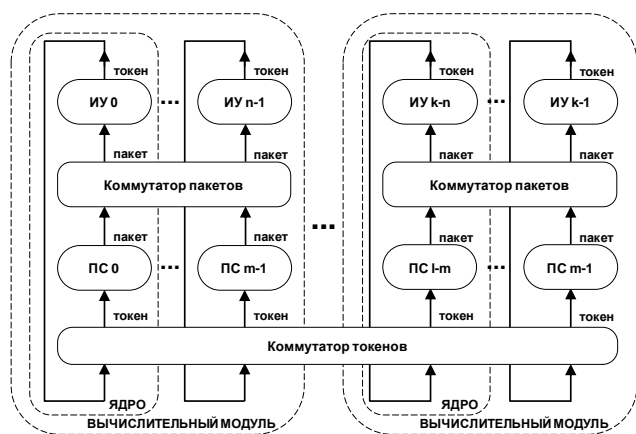


Рис. 1. Структура ППВС «Буря»

ПС выполняет функции сопоставления данных различных узлов, а также хранения токенов, которые ожидают сопоставления с другими токенами. Поступающий на вход ПС токен запускает ассоциативный поиск среди токенов, хранящихся в данном процессоре сопоставления, используя ключ в качестве аргумента поиска. При обнаружении совпадения, ПС формирует пакет и направляет его в ИУ. При отсутствии совпадений, токен хранится в ПС. Формирование пакета – операция, которая создает новую структуру данных – пакет, готовый для обработки на ИУ.

Коммутатор токенов ведет распределение согласно хэш-функции, аргументом которой является ключ токена, таким образом, чтобы токены с одинаковыми ключами попадали в один процессор сопоставления.

Коммутатор пакетов выполняет коммутацию на

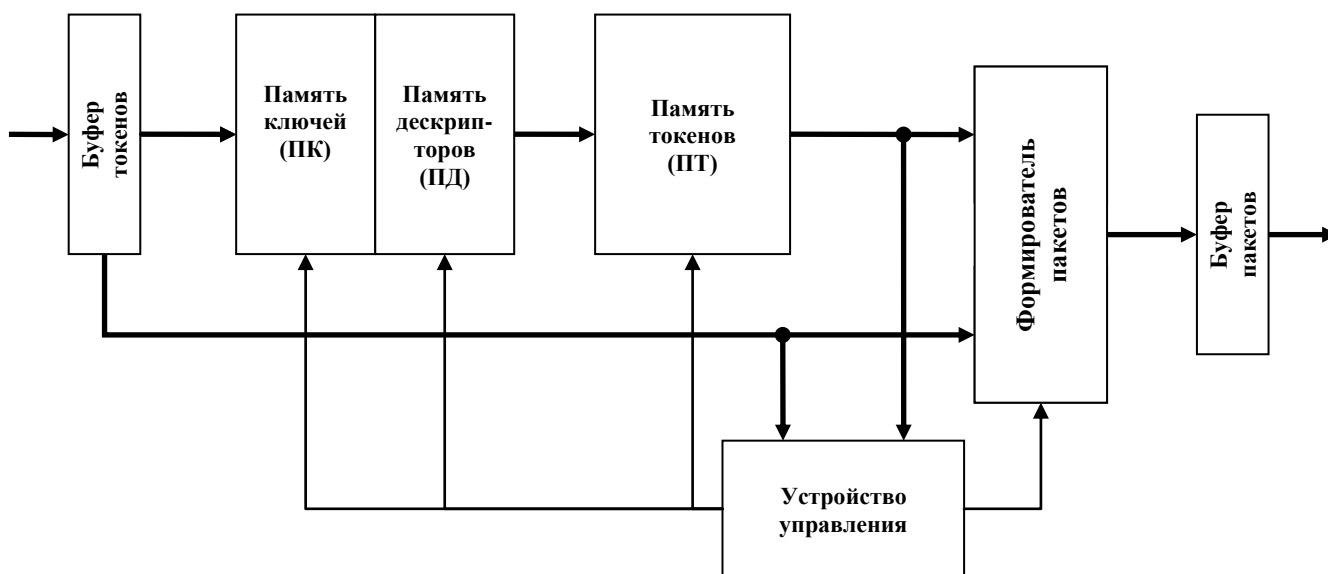


Рис. 2. Структурная схема процессора сопоставления

любое свободное ИУ. Он служит для более равномерной загрузки ИУ в пределах вычислительного модуля, фактически являясь аппаратным балансировочным устройством.

III. ПРОЦЕССОР СОПОСТАВЛЕНИЯ

Структурная схема ПС показана на рис. 2. Токен, поступивший на вход ПС (будем называть его “верхним” токеном) записывается во входной буфер, затем выполняется ассоциативный поиск в памяти ключей (ПК) по ключу “верхнего” токена. При наличии совпадения с ключом одного из ранее записанных токенов (будем называть его “нижним” токеном) из памяти дескрипторов (ПД) считывается его дескриптор. “Нижний” токен хранится в памяти токенов (ПТ), а дескриптор представляет собой адрес, по которому токен записан в ПТ.

После считывания “нижнего” токена выполняется анализ кодов операций (КОП) и кратностей обоих токенов, на основании результатов которого устройство управления (УУ) формирует сигналы разрешения операций записи, удаления и формирования пакета. Если кратность “верхнего” токена не исчерпана, то он записывается в ПТ, а его ключ и дескриптор – в ПК и ПД соответственно. Кратность токена определяет число токенов, с которыми данный токен может взаимодействовать прежде, чем будет удален. Фактически, N-кратный токен эквивалентен последовательности из N однократных токенов. Если кратность “нижнего” токена исчерпана, он удаляется, если нет – обновляется записанное значение кратности.

Поиск может быть начат после загрузки первых двух слов (заголовка и ключа) “верхнего” токена. Чтение “нижнего” токена – после получения дескриптора. Анализ кодов операций возможен, когда доступны заголовки “верхнего” и “нижнего” токена, то есть после чтения первого слова “нижнего” токена. Формирование пакета можно начинать после получения хотя бы



Рис. 3. Временная диаграмма работы ПС без применения конвейера

первых слов “верхнего” и “нижнего” токена и после завершения анализа. Передача пакета может быть начата после того, как сформировано хотя бы одно слово пакета. Запись в память “верхнего” токена возможна после завершения анализа.

С учетом вышесказанного построим временную диаграмму обработки токена в процессоре сопоставления без конвейерной работы (рис. 3).

IV. СТАДИИ КОНВЕЙЕРА

Обработку токена можно разделить на следующие стадии:

- загрузка “верхнего” токена во входной буфер ПС (стадия занимает три такта);
- поиск в ПК по ключу “верхнего” токена и выборка дескриптора “нижнего” токена (стадия занимает два такта);
- чтение “нижнего” токена из ПТ (стадия занимает три такта);
- анализ кодов операций и кратностей “верхнего” и “нижнего” токена (стадия занимает один такт);
- формирование пакета;
- передача пакета в исполнительное устройство (стадия занимает 4 такта);
- запись в память “верхнего” токена и/или удаление “нижнего” (если необходимо). Запись в ПК и ПД – один такт, в ПТ – три такта.
 - Пропускная способность ПС в данном случае составит 1 пакет за 11 тактов, при этом большую часть времени устройства, входящие в состав процессора, находятся в режиме ожидания.

Рассмотрим работу устройства в конвейерном режиме: прием второго “верхнего” токена можно начинать не на 12-ом такте, а уже на 7-ом (рис 4).

Время обработки одного токена по-прежнему со-



Рис. 4. Временная диаграмма работы ПС с применением конвейера

ставляет 11 тактов, но пропускная способность ПС после “разгона” конвейера достигает 1 пакета за 6 тактов.

V. СПОСОБЫ УВЕЛИЧЕНИЯ ПРОПУСКНОЙ СПОСОБНОСТИ КОНВЕЙЕРА

A. Применение двухпортовой памяти токенов

Анализ приведенной диаграммы позволяет определить устройство, лимитирующее скорость работы конвейера – память токенов. После разгона конвейера память токенов выполняет операцию (чтение или запись) каждый такт, тогда как остальные устройства часть времени простаивают. Ускорения работы памяти токенов можно достигнуть, если выполнять чтение и запись одновременно, для чего память должна быть двухпортовой. Первый порт обеспечивает чтение “нижнего” токена, второй – запись “верхнего”.

При работе с многопортовой памятью следует исключить возникновение конфликтов при доступе к одному адресу по разным портам одновременно.

Конфликты вида “запись-запись” не могут возникнуть принципиально, так как запись возможна только через первый порт, второй поддерживает только чтение. Конфликты вида “запись-чтение” также невозможны, так как операция чтения начинается раньше операции записи. Таким образом, на момент записи первого слова токена, чтение этого слова уже завершено, то есть физические адреса, по которым одновременно выполняется доступ к памяти по разным портам, различаются.

Пропускная способность ПС при использовании двухпортовой памяти токенов составляет 1 пакет за 5 тактов (рис. 5).

B. Хранение кода операции в памяти дескрипторов

После замены однопортовой памяти токенов на двухпортовую пропускная способность конвейера увеличилась, но все еще не достигает теоретической максимально возможной. Теперь лимитирующим устройством становится память ключей и дескрипторов.

Такты	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Вход.	прием																
ПК		поиск															
ПТ(1)			поиск														
ПТ(2)				чтение													
УУ					чтение												
Форм.						запись											
Вых.							ан.										
							формирование										
							передача										

Рис. 5. Временная диаграмма работы ПС с двухпортовой памятью токенов

Такты	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Вход.	прием																
ПК		поиск															
ПТ(1)			поиск														
ПТ(2)				чтение													
УУ					чтение												
Форм.						запись											
Вых.							ан.										
							формирование										
							передача										

Рис. 6. Временная диаграмма работы ПС при хранении КОП и кратности в памяти дескрипторов

Операция поиска (N+1)-го токена следует сразу за операцией записи N-го, именно этот участок не дает “сдвинуть” обработку (N+1)-го токена влево по оси времени, чтобы уменьшить задержки конвейера.

Можно добиться уменьшения простоев ассоциативной памяти за счет сокращения времени между чтением дескриптора и записью токена. Между данными стадиями должны выполняться чтение заголовка “нижнего” токена и анализ кодов операции и кратностей. Для анализа нужен не весь заголовок токена (одно слово, 64 бита), а только поля “код операции” (КОП) и “кратность” (суммарно 18 бит).

Если хранить КОП и кратность нижнего токена вместе с дескриптором в памяти дескрипторов, можно выполнять анализ, не дожидаясь чтения из памяти токенов, а сразу после начала обработки отклика. Таким образом, интервал времени между поиском и записью в ПК сокращается на один такт, соответственно на один такт раньше можно запускать загрузку и поиск следующего токена. Временная диаграмма работы ПС для данного решения показана на рис. 6.

Пропускная способность конвейера достигает максимально возможной при данной аппаратной реализации ПС – 1 пакет каждые 4 такта.

VI. ЗАКЛЮЧЕНИЕ

В настоящей статье описана архитектура параллельной потоковой вычислительной системы, рассмотрены механизмы работы одного из ключевых модулей системы – процессора сопоставления. Предложен путь повышения пропускной способности ПС без больших аппаратных затрат – использование конвейерной обработки увеличивает пропускную способность почти в три раза: от 1 пакета в 11 тактов до 1 пакета в 4 такта. Показан процесс пошаговой оптимизации конвейера, при котором на каждом шаге определяется устройство, являющееся “узким местом”, и предлагаются решения, устраняющие ограничение производительности.

В ближайшее время будет реализован процессор сопоставления на ПЛИС, а в дальнейшем – создан макет параллельной потоковой вычислительной системы.

ЛИТЕРАТУРА

- [1] Стемпковский А.Л., Левченко Н.Н., Окунев А.С., Цветков В.В. Параллельная потоковая вычислительная система – дальнейшее развитие архитектуры и структурной организации вычислительной системы с автоматическим распределением ресурсов // Информационные технологии. 2008. №10. С. 2-7.
- [2] Левченко Н.Н., Окунев А.С. Специализация архитектуры многоядерной параллельной потоковой вычислительной системы для решения задачи быстрого преобразования Фурье // IV Всероссийская научно-техническая конференция “Проблемы разработки перспективных микро- и наноэлектронных систем - 2010”. Сб. трудов / под общ. ред. академика РАН А.Л. Стемпковского. 2010. С. 458-461.
- [3] Kohonen, T. Content-addressable memories. Berlin; New York: Springer-Verlag. 1980.