УДК 004.312.44

Сопроцессор комплексных вычислений

П.С. Зубковский, Е.В. Ивасюк, С.И. Аряшев

Учреждение Российской академии наук

Научно-исследовательский институт системных исследований РАН,

zubkovsky@cs.niisi.ras.ru

Аннотация **– представлено описание сопроцессора комплексных вычислений и контроллера прямого доступа к памяти (DMA), реализованных в составе универсального суперскалярного RISC-микропроцессора.**

Ключевые слова **– сопроцессор, комплексные вычисления, DMA, память, расположенная на кристалле.**

I. Введение

Целью создания сопроцессора комплексных вычислений и контроллера прямого доступа к памяти было повышение производительности универсального RISC-микропроцессора при выполнении задач обработки больших потоков данных, таких как быстрое преобразование Фурье, перемножение матриц и другие. Одним из требований к полученной системе было достижение производительности на уровне 10-ти вещественных операций за такт с сохранением универсальности, гибкости и удобства программирования.

Разработанный сопроцессор комплексных вычислений представляет собой вычислительное устройство, ориентированное на выполнение арифметических операций над 32-разрядными комплексными и вещественными числами [1]. Структурная схема сопроцессора представлена на рис. 1. Основными характерными чертами сопроцессора являются:

- два параллельных конвейера, выполняющие две инструкции процессора за такт: одну арифметическую инструкцию и одну инструкцию обмена данными;
- возможность 128-разрядного обмена данными с накристальной памятью процессора;
- регистровый файл, содержащий 64 64-разрядных регистра, имеющий 4 порта для записи и 8 портов для чтения данных;
- арифметическое ядро, позволяющее выполнять до 12-ти вещественных операций за такт;
- схема передачи операндов на арифметическое ядро, позволяющая реализовать в сопроцессоре большой набор инструкций комплексной и вещественной арифметики;
- схема обратных связей по данным, позволяющая уменьшить время выполнения инструкций, избежав, в тоже время, конфликтов по записи данных в регистровый файл;
- схемы вычисления обратной величины и обратного квадратного корня.

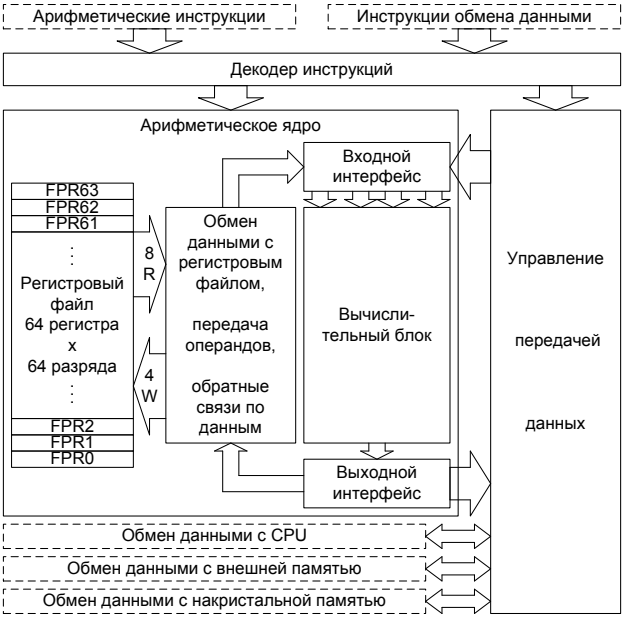

Рис. 1. Структурная схема сопроцессора

Контроллер прямого доступа к памяти интегрирован с сопроцессором комплексных вычислений через набор управляющих регистров, отображенных на управляющие регистры сопроцессора. Контроллер предоставляет большие возможности для организации массива данных, являющихся входными данными для сопроцессора комплексных вычислений во внутренней памяти, т.е. памяти, расположенной на кристалле. Это достигается за счет использования алгоритма трехмерной выборки данных из внешней памяти и независимого трехмерного размещения данных во внутренней памяти.

Основным источником входных данных для сопроцессора является внутренняя память, физически являющаяся кэш-памятью второго уровня.

II. Описание сопроцессора комплексных вычислений

Сопроцессор подключен к стандартному потоку инструкций микропроцессора и способен выполнять две инструкции процессора одновременно в каждом такте. На первый поток инструкций поступают арифметические инструкции, а на второй – инструкции обмена данными с внешней памятью, памятью, расположенной на кристалле, с регистровыми файлами ядра микропроцессора и блока вещественной арифметики. Обмен данными с внешней памятью, с ядром микропроцессора и блоком вещественной арифметики производится 64-разрядными словами. В обмене данными с внешней памятью участвуют буфер трансляции адресов и кэш-памяти 1-го и 2-го уровней.

Обмен данными с внутренней памятью, происходит без использования буфера трансляции адресов и кэш-памяти 1-го уровня. Физический адрес для доступа во внутреннюю память, формируется непосредственно в ядре микропроцессора при выполнении инструкции загрузки/сохранения. Обмен может производиться 128-разрядными словами. Для этого между внутренней памятью и сопроцессором предусмотрены две 128-разрядные шины данных для загрузки и выгрузки. Два 64-разрядных слова, поступившие из внутренней памяти, записываются в соседние регистры регистрового файла сопроцессора. Выгрузка 128-разрядного слова производится также из двух соседних регистров.

Характеристики регистрового файла оказывают большое влияние на возможности сопроцессора в целом. Недостаток портов для записи и чтения может вызвать конфликты между инструкциями и блокировки конвейера, а недостаток регистров заставит пользователя применять более короткие вычислительные циклы, что тоже может отрицательно сказаться на производительности системы.

Используемый регистровый файл имеет 4 порта для записи и 8 портов для чтения данных. Этого достаточно, чтобы избежать блокировок конвейера при любой комбинации инструкций процессора на обоих входах сопроцессора. Так, любая арифметическая инструкция требует чтения не более четырех 64-разрядных операндов и может записать в регистровый файл не более двух 64-разрядных результата. Одновременно выполняемая инструкция загрузки/сохранения требует чтения одного или двух 64-разрядных слов или может записать в регистровый файл не более двух 64-разрядных слов. Таким образом, этих ресурсов достаточно для непрерывной работы сопроцессора. Кроме того, возможна запись только половины 64-разрядного регистра, что позволяет использовать 32-разряные инструкции процессора. Пользователю доступны 64 регистра по 64 разряда каждый, адресуемые либо по отдельности, либо парами четный/нечетный.

Арифметическое ядро сопроцессора включает в себя набор аппаратных средств, позволяющий выполнять до 16-ти операций над 32-разрядными числами с плавающей запятой за один такт машинного времени. Арифметическое ядро построено из 4-х блоков, выполняющих операции умножения и суммирования 32-разрядных чисел с плавающей запятой – блоков MAF (Multiply-Add Fused – смешанное умножение с суммированием) [2]. Блоки объединены попарно в два параллельных конвейера (см. рис.2).

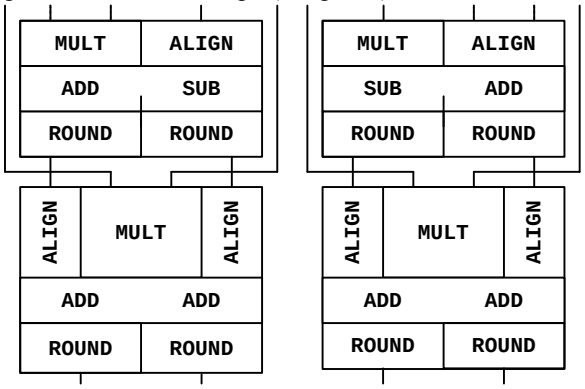

Рис. 2. Структурная схема арифметического ядра сопроцессора

В базовом исполнении блок MAF предназначен для выполнения операции вида D = C + A * B [3], причем все три операнда начинают обрабатываться одновременно, что дает существенный выигрыш в производительности. Кроме того, число этапов округления сведено к одному, причем оно выполняется одновременно с финальным суммированием. Это также дает прирост производительности и повышает точность результата. Процесс обработки входных операндов условно можно разделить на три этапа:

1) умножение мантисс чисел A и B, параллельно выравнивание мантиссы числа C;

2) предварительное суммирование результата AB и выровненной мантиссы C, параллельно предсказание суммы, подсчет старших нулей, нормализация промежуточного результата;

3) окончательное суммирование, округление, коррекция экспоненты результата.

В представленной реализации используются два вида более сложных схем MAF. Блоки MAF, находящиеся на первых стадиях вычислительных конвейеров, имеют три входных операнда (OP1, OP2, OP3), как и обычный блок MAF. Их отличие состоит в том, что на выходе формируются два округленных результата – результат умножения с суммированием OUT2=OP3+OP1*OP2 и результат умножения с вычитанием OUT3=OP3-OP1*OP2. Блоки MAF, работающие на второй стадии вычислительного конвейера,

имеют четыре входных операнда (OP1, OP2, OP3, OP4). Два операнда перемножаются, к их произведению прибавляются два других операнда по отдельности. Эти блоки имеют два выхода, на одном из которых формируется результат OUT1=OP3+OP1*OP2, а на другом результат OUT2=OP4+OP1*OP2.

Комбинируя входные 32-разрядные операнды на входах арифметического блока, а также управляя знаками входных операндов можно реализовать большой набор арифметических инструкций как комплексной, так и вещественной и векторной арифметики. Сюда относятся инструкции сложения, вычитания, умножения, умножения с накоплением, производимые над комплексными числами, сдвоенными вещественными числами, вещественными 2-векторами или матрицами 2x2. Отдельно следует отметить инструкции вычисления квадрата модуля комплексного числа и вычисления свертки с накоплением.

Инструкция умножения с накоплением и вычитанием комплексных чисел наиболее полно использует возможности арифметического ядра сопроцессора. Эта инструкция выполняет операции вида:

*RES1.RE = RES1.RE + OP1.RE * OP2.RE - OP1.IM * OP2.IM*
*RES1.IM = RES1.IM + OP1.RE * OP2.IM + OP1.IM * OP2.RE*
*RES2.RE = RES2.RE - OP1.RE * OP2.RE + OP1.IM * OP2.IM*
*RES2.IM = RES2.IM - OP1.RE * OP2.IM - OP1.IM * OP2.RE,*

где RE и IM – действительная и мнимая части комплексного числа, соответственно.

Из приведенной формулы видно, что эта инструкция выполняет 4 операции умножения и 6 операций суммирования или вычитания над числами с плавающей запятой одинарной точности.

Для выполнения ряда арифметических инструкций достаточно использовать только блоки MAF, работающие на первых стадиях вычислительного конвейера. Длительность таких инструкций составляет 4 такта. Полная длительность арифметического конвейера составляет 7 тактов. Различие длительностей инструкций, выполняемых в одном потоке, может привести к конфликту по записи. Чтобы избежать подобных конфликтов, длительность коротких инструкций была сделана равной длительности более длинных путем добавления 3-х «холостых» тактов.

Схема обратных связей по данным позволяет передавать результат предыдущей инструкции на вход следующей, минуя стадию записи в регистровый файл. Это позволяет не только уменьшить промежутки между зависимыми инструкциями, но и компенсировать увеличение длительности коротких инструкций. Т.е. инструкция, имеющая зависимость от длинной 7-митактовой инструкции, может быть выдана на выполнение с промежутком в 6 тактов, а инструкция, имеющая зависимость от 4-хтактовой инструкции, может быть выдана через 4 такта.

В конвейер сопроцессора были включены два арифметических блока для вычисления обратной величины и обратного квадратного корня от 32-разрядного вещественного числа. В этих блоках реализован алгоритм функциональных итераций [4] (метод Ньютона) с начальной кусочно-линейной аппроксимацией первого порядка. Встраивание этих блоков не потребовало изменять свойства арифметического конвейера сопроцессора.

III. Описание контроллера прямого доступа в память

Несмотря на то, что данные могут быть загружены в регистровый файл сопроцессора и из внешней памяти, и из регистрового файла микропроцессора или блока вещественной арифметики, наибольшая производительность достигается при осуществлении обмена данными с внутренней памятью. Заполнение внутренней памяти и выгрузка данных из нее производится по каналу прямого доступа к памяти из внешней памяти процессора. Дополнительное повышение эффективности программирования можно получить, если данные организованны нужным образом уже на стадии загрузки во внутреннюю память. Например, в ряде алгоритмов работы с матрицами возникает необходимость транспонировать отдельные участки этих матриц произвольного размера перед загрузкой в сопроцессор.

Решить эти задачи призван контроллер прямого доступа в память (DMA) с возможностью трехмерной выборки данных во внешней памяти и независимого трехмерного размещения данных во внутренней памяти. Контроллер управляется набором регистров, доступных для программиста как контрольные регистры сопроцессора комплексных вычислений. В эти регистры записываются начальные значения областей во внешней и внутренней памяти, а также параметры циклов, вычисления адреса для текущей передачи.

Вычисление адреса во внешней памяти происходит по следующему правилу:

Цикл по z = {0, MEM_Nz – 1}
Цикл по y = {0, MEM_Ny – 1}
Цикл по x = {0, MEM_Nx – 1}
$MEM\_Addr_{35\ldots0} = (0_{35\ldots32} \,||\, MEM\_BASE\_ADDR_{31\ldots0}) + z * (0_{35\ldots32} \,||\, Z\_MEM\_STEP_{31\ldots0}) + y * (0_{35\ldots32} \,||\, Y\_MEM\_STEP_{31\ldots0}) + x * (0_{35\ldots32} \,||\, X\_MEM\_STEP_{31\ldots0})$
Конец цикла по x
Конец цикла по y
Конец цикла по z

Вычисление адреса во внутренней памяти происходит по следующему правилу:

Цикл по z = {0, L2_Nz – 1}
Цикл по y = {0, L2_Ny – 1}
Цикл по x = {0, L2_Nx – 1}

$L2\_Addr_{17...0} = L2\_BASE\_ADDR_{17...0} + z * Z\_L2\_STEP_{17...0} + y * Y\_L2\_STEP_{17...0} + x * X\_L2\_STEP_{17...0}$

Конец цикла по x

Конец цикла по y

Конец цикла по z

Подсистема памяти процессора устроена таким образом, что минимальным объемом передаваемых данным является 8 64-разрядных чисел. Но для транспонирования участков матриц необходимо адресовать отдельные 64-разрядные слова с шагом любого размера. Для этого в контроллере DMA реализован механизм выравнивания передач до нужного размера и отмены лишних передач.

После инициализации контроллера DMA и начала передачи, его работа происходит полностью независимо и не занимает ресурсов микропроцессора. После окончания передачи в контрольном регистре выставляется флаг готовности.

IV. Описание внутренней памяти

При выполнении задач обработки больших объемов данных, на первое место по влиянию на производительность выходят возможности подсистемы памяти. В представленной системе роль быстрой внутренней памяти выполняет память, расположенная на кристалле, физически являющаяся кэш-память 2-го уровня. В центральном ядре микропроцессора предусмотрена возможность отключить кэш-память 2-го уровня и использовать эту память как внутреннюю для обмена данными с сопроцессором. Физический адрес для доступа во внутреннюю память при выполнении инструкций загрузки/сохранения вычисляется в центральном ядре процессора и передается непосредственно на внутреннюю память. Буфер трансляции адресов и кэш-память 1-го уровня не участвуют в этом обмене данными. Внутренняя память предоставляет возможность одновременной загрузки и выгрузки 128-разрядных слов в каждом такте. В наборе команд сопроцессора предусмотрены инструкции загрузки и выгрузки 128-ми разрядных слов из регистрового файла сопроцессора. При этом оказываются задействованы два соседних регистра в регистровом файле сопроцессора. В теле инструкции загрузки/сохранения 12-разрядных слов содержится четный номер регистра, одновременно обращение происходит и к следующему за ним регистру.

Внутренняя память имеет объем 256 кбайт и разделена на две половины, которые могут работать одновременно независимо друг от друга с контроллером DMA и инструкциями загрузки/сохранения сопроцессора.

Во время выполнения вычислительного цикла, обмен данными производится между регистровым файлом сопроцессора и одной половиной внутренней памяти. В другой половине в это время может происходить выгрузка результатов предыдущего вычислительного цикла и загрузка данных для следующего.

V. Заключение

Описанные схемы сопроцессора комплексных вычислений и контроллера прямого доступа к накристальной памяти позволяют существенно повысить производительность универсального микропроцессора на задачах обработки данных по сравнению с блоком вещественной арифметики. На представленном наборе команд пиковая производительность сопроцессора составляет 10 вещественных операций одинарной точности за такт для инструкции умножения с накоплением и вычитанием комплексных чисел (элементарная операция БПФ) (3 ГФлопс при частоте 300 МГц) и 8 операций для умножения матрицы на вектор с накоплением (2,4 ГФлопс при частоте 300 МГц). Данные представлены для реализации по проектным нормам 0,18 мкм. Производительность канала DMA ограничена возможностями подсистемы памяти процессора и составляет примерно 1,8 ГБ/с при частоте внешней памяти 166 МГц. При этом скорость передачи данных между внутренней памятью и регистровым файлом сопроцессора достигает 4,8 ГБ/с при частоте процессора 300 МГц. Таким образом, максимальной эффективности можно добиться при увеличении процента повторно используемых данных во внутренней памяти.

Реальная производительность измерялась для нескольких реализаций быстрого преобразования Фурье разных размерностей и составляет от 55% от пиковой для БПФ размерности 256 до 77% от пиковой для БПФ размерности 1024.

Сопроцессор комплексных вычислений и контроллер прямого доступа к памяти реализованы в виде программных моделей на языке Verilog и предназначены для использования в составе программной модели микропроцессора.

Литература

[1] Parhami Behrooz. Computer arithmetic: algorithms and hardware designs USA, New York, Oxford University Press, 2000. 490 p.

[2] Tomas L., Javier B. Floating-point multiply-add fused with reduced latency // IEEE Transactions on Computers. 2004. Vol. 53. No. 8. P. 988-1003.

[3] Hennessy J.L., Patterson D.A. Computer Architecture – a Quantitative Approach. USA, San Francisco, Morgan Kaufmann Publishers, 2003. 1072 p.

[4] Бахвалов Н.С. Численные методы. М.: Наука, 1973.