

УДК 004.272.33

РАЗРАБОТКА ПРОБЛЕМНО-ОРИЕНТИРОВАННЫХ ПРОЦЕССОРОВ

Р.Р. Ковязин, Н.П. Постников

Рассмотрена проблема выбора процессора при проектировании современных информационно-управляющих систем (ИУС), поддерживающих программирование на прикладном уровне. Показано, что решение, предлагаемое традиционной методикой проектирования ИУС, не всегда дает удовлетворительный результат. Показана необходимость разработки нового проблемно-ориентированного процессора. Исследован ряд вопросов, возникающих в ходе проектирования архитектуры нового процессора: микроархитектура, система команд, инструментальные средства и др. Приведено описание оригинальной архитектуры проблемно-ориентированного процессора NL3, разработанного в рамках проекта ООО «ЛМТ» для обработки цифровых данных в реальном времени.

Ключевые слова: VLIW, цифровая обработка дискретных сигналов, микроархитектура, система команд.

Введение

Современные ИУС способны выполнять широкий круг задач. Столь широким распространением ИУС обязаны тому, что они являются системами с доминирующей программной компонентой (Software-

Dominated). В таких системах программная компонента настраивает (конфигурирует) аппаратную компоненту на выполнение нужных функций. Сама по себе аппаратная компонента этих функций не выполняет.

Выбор устройства реализации программного управления является ключевым моментом в процессе разработки ИУС [1]. В идеале разработчики должны оценивать эффективность исполнения программной компоненты на разных процессорах. Производители элементной базы предоставляют широкий выбор процессоров: от 4-битных и до 64-битных, с различным соотношением вычислительных и коммуникационных возможностей, с различными микроархитектурами и системами команд Instruction Set Architecture (ISA). На практике выбор тяготеет к тем процессорам, с которыми коллектив разработчиков имеет опыт работы, т.е. основным критерием становится время разработки. Дополнительную сложность создают сторонние программные компоненты (библиотеки драйверов, операционные системы и др.), чьи свойства не до конца известны в силу большого объема и закрытости информации.

Иногда требования к ИУС таковы, что выбор подходящей модели среди существующих процессоров оказывается затруднителен. Приведем наиболее распространенные причины, по которым возникает такая ситуация.

- Заказчику требуется ИУС с прикладным программированием. При этом заказчик – это прикладной специалист (физик, энергетик, технолог и др.), а не программист. Ему нужен инструмент, позволяющий эффективно формировать прикладную задачу и защищающий систему от его неумелых действий.
- Среди процессоров, удовлетворяющих ограничениям по энергопотреблению, габаритам, интегрированным периферийным блокам и др., не нашлось ни одного, обеспечивающего реализацию функциональности ИУС.

Все это приводит к необходимости разработки собственного специализированного процессора [2, 3]. Если ISA разрабатываемого процессора широко распространена, то для него существует богатый выбор развитых инструментальных средств. Если же разрабатываемый процессор обладает оригинальной системой команд, то приходится разрабатывать весь комплекс инструментальных средств, как минимум, компилятор [4]. Для сокращения времени разработки ИУС компилятор может разрабатываться параллельно с самим процессором, так что ISA и микроархитектура этого процессора для компилятора не будут зафиксированными. В проектировании архитектуры процессора смогут принять участие как разработчики процессора, так и разработчики компилятора. Так появляется возможность выбора того, что будет реализовано инструментальными средствами, а что в самой ИУС.

Данная статья описывает архитектуру проблемно-ориентированного процессора NL3, разработанного авторами в рамках проекта ООО «ЛМТ» по реализации алгоритмов цифровой обработки сигналов в реальном времени.

Выбор архитектуры процессора

Зачастую обработку сигналов представляют в виде схем, состоящих из фильтров, усилителей, сумматоров, аттенуаторов, преобразователей и т.п. Топология такой схемы показывает суть выполняемой обработки цифровых сигналов. Следовательно, говоря об алгоритме обработки дискретных сигналов, будем подразумевать схему этого алгоритма. Изменение топологии схемы – это изменение подключения ее элементов, т.е. изменение алгоритма. Вместе с тем у отдельных элементов схемы есть численные характеристики: коэффициенты усиления, коэффициенты фильтров и др. Изменение этих параметров не меняет топологии схемы, поэтому алгоритм обработки данных не меняется, но меняется результат работы схемы.

Традиционно задачи цифровой обработки дискретных сигналов в ИУС решаются на базе процессоров сигнальной обработки (DSP) или программируемых логических интегральных схем (ПЛИС). DSP предоставляют расширенные вычислительные возможности по сравнению с процессорами общего назначения, однако, чтобы воспользоваться ими, нужно писать программы, учитывая особенности компилятора. ПЛИС позволяют воспользоваться параллелизмом в обработке сигналов, но технология их программирования существенно отличается от программирования последовательных процессоров. В любом случае программирование DSP и ПЛИС требует высокой квалификации и специальных знаний от разработчика.

Если ИУС является программируемой системой, а ее заказчик – прикладной специалист, то решения на базе DSP и ПЛИС не подходят, поскольку у заказчика нет необходимой квалификации для эффективного решения его прикладной задачи. В данном случае требуется возможность конфигурирования работы системы заказчиков в следующих проявлениях:

- изменение топологии схемы алгоритма на фазе разработки;
- изменение параметров блоков схемы алгоритма на фазе исполнения.

Было принято решение о необходимости разработки специализированного процессора [5]. В качестве базового устройства реализации программного управления была выбрана ПЛИС, поскольку она может обеспечить высокую производительность за счет параллелизма исполнения схемы обработки сигналов.

Архитектура процессора NL3

По классификации Флинна [6] процессор NL3 относится к классу MISD-процессоров. NL3 содержит в себе набор специализированных арифметико-логических устройств Data Processing Unit (DPU), соединенных шиной данных (рис. 1). DPU способны выполнять различные вычисления: это могут быть сумматоры, фильтры, умножители, память, преобразователи интерфейсов, мультиплексоры и др. Каждое DPU выполняет функции определенного элемента схемы обработки цифровых сигналов. Поскольку в схеме эти элементы связаны, то нужно передавать данные между DPU. Для этого в микроархитектуре процессора NL3 предусмотрены шины данных. Одно DPU выставляет свои данные на шину данных, а другие DPU (их может быть несколько) загружают эти данные в себя. Время работы DPU не зависит от значений входных данных, загруженных с шины. Несмотря на такой способ обмена, все DPU функционируют независимо друг от друга. Передача данных между DPU осуществляется с помощью совмещения тактов чтения и записи. Состоянием шины данных отдельно никто не управляет, она является разделяемым ресурсом для DPU.

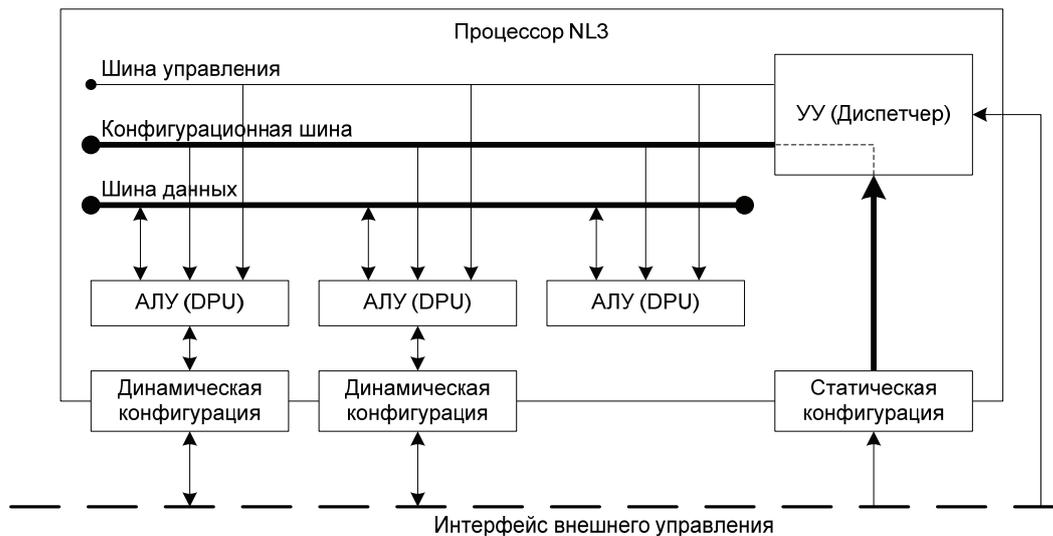


Рис. 1. Архитектура процессора NL3

Для выполнения любой операции DPU должно получить команду. Для подачи команд DPU предусмотрены конфигурационная шина и шина управления. Вся функциональность DPU реализуется через выполнение трех типов команд:

- загрузка данных с шины данных;
- запуск вычислений;
- выгрузка результатов вычислений на шину данных.

По конфигурационной шине передаются индивидуальные команды DPU. Кодирование команд на конфигурационной шине уникально для каждого DPU, при этом к конкретному DPU подключены только те линии конфигурационной шины, по которым передаются команды (рис. 2). По одним и тем же линиям могут передаваться команды разным DPU. Программа для процессора NL3 называется статической конфигурацией. Это последовательность значений, выставляемая на конфигурационную шину. В этих значениях закодирован порядок подачи команд загрузки/выгрузки и запуска вычислений DPU. Поскольку через загрузку и выгрузку осуществляется передача данных между DPU, то это формирует топологию алгоритма цифровой обработки сигналов. Операции запуска вычислений формируют функциональность этого алгоритма.

Диспетчер процессора NL3 – это блок, управляющий состоянием конфигурационной шины на каждом такте в соответствии со статической конфигурацией. Состояние конфигурационной шины на конкретном такте является инструкцией процессора NL3, а список таких инструкций – программой. Программа исполняется диспетчером всегда линейно, без пропусков инструкций, от начала и до конца, за одно и то же время (количество тактов синхронизации диспетчера). В отличие от конфигурационной шины, по шине управления передаются широкоэвентральные команды. Диспетчер по шине управления передает команды, сигнализирующие начало и конец исполнения списка инструкций. Кодирование этих команд – одинаковое для всех DPU. По командам, полученным по шине управления, DPU могут инициализировать свои внутренние регистры, а некоторые из них запускают вычисления. Все линии шины управления одинаково интерпретируются и подключены ко всем DPU.

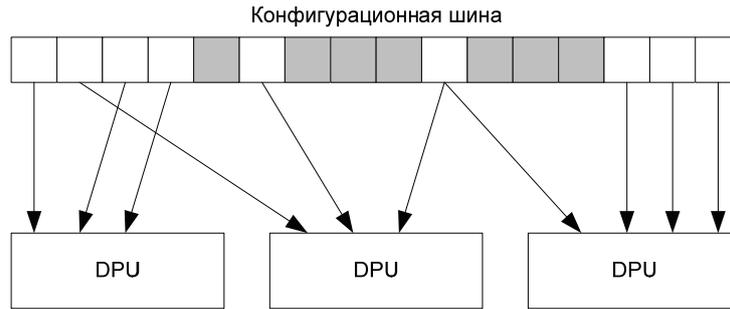


Рис. 2. Кодирование команд на конфигурационной шине

Постоянная работа схемы обработки цифровых сигналов реализована в процессоре NL3 через периодическое исполнение программы диспетчером: закончив формирование команд по списку инструкций, диспетчер начинает его заново. Поскольку программа всегда исполняется в одно и то же время, то задержка активации DPU относительно входных данных постоянна. Поскольку алгоритм цифровой обработки сигналов работает в реальном времени, то он должен успевать выполняться для каждого нового сигнала. Таким образом, для каждого входного сигнала алгоритм повторяется. При известной частоте дискретизации F_{SR} и частоте работы диспетчера F_{DISP} , можно легко вычислить максимальный размер программы LEN – количество инструкций, которое может исполнить диспетчер между появлением двух дискретных сигналов:

$$LEN = F_{SR} / F_{DISP} .$$

Процессор NL3 является лишь вычислительным ядром. Для запуска и остановки, для загрузки статической конфигурации нужен сервисный процессор – внешний вычислитель, обеспечивающий работу процессора NL3. Эти функции может выполнять любой процессор общего назначения. Изменение топологии алгоритма цифровой обработки – это изменение связей между DPU. Для этого требуется перекомпиляция программы, в то время как у некоторых DPU есть параметры, изменение которых не влияет на их функциональность: масштабные множители, значения табличных выборок, коэффициенты фильтров и др. Их можно менять без остановки процессора, и для этого предусмотрен механизм динамического конфигурирования. Динамическая конфигурация – это память, доступная DPU и сервисному процессору. Сервисный процессор может читать и писать в эту память. Поскольку сервисный процессор и DPU могут работать на разных частотах, для корректной работы DPU предусмотрены механизмы синхронизации. Если для обработки цифровых сигналов не требуется производить настройку алгоритма, то механизм динамического конфигурирования не нужен. Возможна реализация, когда статическая конфигурация хранится во внутренней памяти диспетчера, а параметры всех DPU задаются по умолчанию корректными значениями. В таком случае процессор NL3 будет работать без интерфейса внешнего управления и динамического конфигурирования.

Поскольку процессор NL3 обладает оригинальной архитектурой, для него требуется разработка специализированных инструментальных средств, в том числе компилятора. Основная задача компилятора заключается в формировании статической конфигурации. Критериями размещения команд в рамках цикла инструкций является то, что у соединенных DPU должны совпадать такты команд загрузки и выгрузки. Тогда данные будут передаваться между DPU. Кроме того, если у DPU пересекаются команды на конфигурационной шине, то нужно эти команды расположить так, чтобы исключить некорректную активацию DPU. Также необходимо решить проблему повторного использования отдельных DPU в течение одного цикла. Разработка компилятора с такими функциями нетривиальна. К тому же крайне желательна графическая форма разработки алгоритмов, поддерживаемая мощными САПР.

Заключение

Архитектура процессора NL3 сравнима со свойствами процессоров с суперскалярной и VLIW (Very Long Instruction Word) [7, 8] архитектурами, которые приведены в таблице. Часть свойств NL3 выходит за рамки суперскалярной и VLIW архитектур и является оригинальной.

- У DPU разных типов отличается время работы. Вместе с тем время работы однотипных DPU тоже может отличаться. В суперскалярных и VLIW архитектурах активация исполнительного блока – это инструкция. NL3 поддерживает более низкоуровневое программирование. Для активации DPU требуется несколько команд: загрузка, выгрузка и начало вычислений. Можно варьировать количество тактов между этими командами и, благодаря этому, время работы даже однотипных DPU может отличаться.
- Длинные инструкции передаются по конфигурационной шине. К каждому DPU подключается только часть линий шины. Одни и те же линии могут быть подключены к нескольким DPU, что соответст-

вует наложению команд для разных DPU в рамках одной инструкции. Если в VLIW архитектуре каждый фрагмент длинной инструкции предназначен для отдельного исполнительного блока, то у NL3 эти фрагменты могут пересекаться.

- DPU поддерживают конвейеризацию. Вычисления начинаются по специальной команде, так что в DPU можно загружать входные данные до выгрузки результатов предыдущих вычислений. Загрузка входных данных требует определенного времени, но она не портит значения, посчитанные ранее.

		NL3	Суперскалярная архитектура	VLIW архитектура
Исполнительные блоки (DPU)				
Количество		Много	Много	Много
Время работы	Однотипных	Разное	Одинаковое	Одинаковое
	Разнотипных	Разное	Разное	Одинаковое
Регистровые банки		Нет	Нет	Есть
Программирование				
Общая инструкция		Да	Нет	Да
Планирование		Статическое	Динамическое	Статическое
Плотность кода		Низкая	Высокая	Низкая
Перекрытие инструкций		Да	Нет	Нет

Таблица. Сравнение свойств процессоров с суперскалярной и VLIW архитектурами

Процессор NL3 разработан на базе ПЛИС XC3S400A. Он обладает следующими характеристиками:

- разрядность: шины данных – 32 бита, конфигурационной шины – 72 бита, шины управления – 3 бита;
- частота: дискретизации F_{SR} – 1 МГц, диспетчера F_{DISP} – 200 МГц.

Типовые задачи цифровой обработки дискретных сигналов для процессора NL3: фильтрация, ПИД-регулирование, синхродетектирование. Для этого были разработаны следующие DPU:

- цифровой обработки: вычислитель поворота координат, коррелятор, КИХ/БИХ-фильтр;
- общего назначения: сумматор, табличная выборка, умножитель, память, мультиплексор;
- специализированные: связи с внешними устройствами, осциллограф, отладчик.

Реализация процессора NL3, апробация его на тестовых и целевых задачах с учетом затрат на его разработку показали высокую эффективность предложенной архитектуры. На текущий момент продолжают исследования вопросов компиляции и оптимизации работы DPU.

Литература

1. Ковязин Р.Р. Влияние аппаратной организации на технологию программирования во встроенных системах / Р.Р. Ковязин // Научно-технический вестник СПбГУ ИТМО. – 2003. – Вып. 19. – С. 172–177.
2. Ковязин Р.Р. Использование аппаратных интерпретаторов для управления неоднородными компонентами микропроцессорной системы / Р.Р. Ковязин, А.Г. Чистяков // Научно-технический вестник СПбГУ ИТМО. – 2002. – Вып. 6. – С. 71–75.
3. Ковязин Р.Р. Создание локальных регуляторов на базе виртуальной машины с динамическим набором инструкции / Р.Р. Ковязин, Н.П. Постников // Научно-технический вестник СПбГУ ИТМО. – 2006. – Вып. 32. – С. 55–62.
4. Ковязин Р.Р. Проектирование проблемно-ориентированных языков для встроенных вычислительных систем / Р.Р. Ковязин // Политехнический симпозиум «Молодые ученые – промышленности Северо-Западного региона». Тезисы докладов. – СПб: СПбГПУ, 2002. – 66 с.
5. Smith J. Virtual Machines: Versatile Platforms for Systems and Processes / Jim Smith, Ravi Nair. – Morgan Kaufmann, 2005. – P. 656.
6. Flynn M. Some Computer Organizations and Their Effectiveness // IEEE Trans. Comput. – 1972. – V. C-21. – P. 948–960.
7. Fisher J. Very Long Instruction Word Architectures and the ELI-512 / Joseph A. Fisher // Proceedings of the 10th annual international symposium on Computer architecture. International Symposium on Computer Architecture. – New York: ACM. – 1983. – P. 140–150.
8. Fisher J. Embedded Computing: A VLIW Approach to Architecture, Compilers and Tools // Joseph A. Fisher, Paolo Faraboschi, Cliff Young. – Morgan Kaufmann, 2004. – P. 712.

Ковязин Рустам Раисович

Постников Николай Павлович

– ООО «ЛИТ», магистр математики, rus-k@yandex.ru

– ООО «ЛИТ», главный инженер, кандидат технических наук, nick.postnikov@gmail.com