Применение композитной логики для проектирования реконфигурируемых цифровых устройств управления

Баркалов А.А., Зеленёва И.Я., Гриценко А.А.

кафедра ЭВМ ДонНТУ, irina@cs.dgtu.donetsk.ua
Зеленогурский университет (Польша), A.Barkalov@iie.uz.zgora.pl

Abstract

Barkalov A.A., Zelenyova I.J., Grytsenko A.A. Composite logic approach for reconfigurable digital control units design. *In the paper composite module architecture is described – adaptive hardware system using dynamically reconfiguration. Showed composite module architecture approach for digital control units design for which control algorithm formalization deferred to deploying and maintenance phases.*

Введение

Реконфигурируемые системы являются современной аппаратной платформой, предоставляющей возможность динамического изменения структуры без остановки работы устройства [1]. Реконфигурируемые системы могут использоваться в различных направлениях, в частности, для разработки адаптивных вычислительных систем [2].

На данный момент актуальным является проектирование адаптивных систем, в том числе аппаратных, которые могут после развертывания, в процессе эксплуатации, реализовывать широкий круг алгоритмов для взаимодействия с окружением. Такие системы должны предоставлять возможности гибкой обработки запросов и расширения базы поддерживаемых алгоритмов, что достигается с помощью использования паттернов проектирования.

Понятие паттернов проектирования включает описание взаимодействия некоторых объектов для решения общей задачи проектирования [4]. Паттерны могут применяться в различных областях, в частности, при проектировании цифровых систем управления. Для перехода от объектно-ориентированных систем к цифровым системам можно использовать унифицированный язык моделирования [2], с помощью которого описаны паттерны проектирования, и управляемую моделями архитектуру, позволяющую описывать программно-аппаратные системы на высоком уровне, независимом от реализации [3]. Таким образом, можно применять некоторые паттерны проектирования для описания аппаратных систем с сохранением их семантики.

В данной работе предлагается архитектура композитного модуля – адаптивной аппаратной системы, прозрачно изменяющей свою внутреннюю структуру для обработки запросов внешних систем [5]. В качестве примера применения предлагаемой архитектуры рассматривается построение цифрового устройства управления [6, 7] с композитной логикой. Эта система предоставляет возможность развертывания любого алгоритма управления с учетом налагаемых физических ограничений на производительность и время отклика.

Архитектура композитного модуля

Композитный модуль – это аппаратная система, которая объединяет жесткую [6,7] (статическую составляющую) и реконфигурируемую логику [1] (динамическую составляющую), обеспечивая возможность адаптивного взаимодействия с внешней средой. Объединение жесткой и реконфигурируемой логик формирует композитную логику. Композитный модуль описывается с использованием ряда паттернов [4] (рис. 1):

- фасад (facade) – обеспечивает взаимодействие с внешними подсистемами, поддержку соответствующих интерфейсов и получение запросов; инкапсулирует компоненты, реализующие внутреннюю логику модуля;
- прокси (proxy) – интерфейсный модуль приложения обработки запросов, обеспечивающий правильность выбора необходимого состояния, предварительную обработку и ретрансляцию запросов;
- состояние (state) – часть приложения обработки запросов, обеспечивающая обработку запросов и формирование результатов;
- строитель (builder) – инфраструктурный модуль, обеспечивающий конфигурацию различных частей приложения обработки запросов по требованию прокси.

Композитный модуль обладает рядом свойств:

- гибкость – обеспечивается возможностью реализации множества алгоритмов проблемной области ограниченной поддерживаемыми контрактами с внешней средой (определяемого физическими интерфейсами);
- прозрачность – обеспечивается инкапсуляцией использования реконфигурируемой логики и соответствующих этому процессов;
- адаптивность – обеспечивается возможностью изменения контрактов с внешней средой, следовательно, изменения проблемной области.

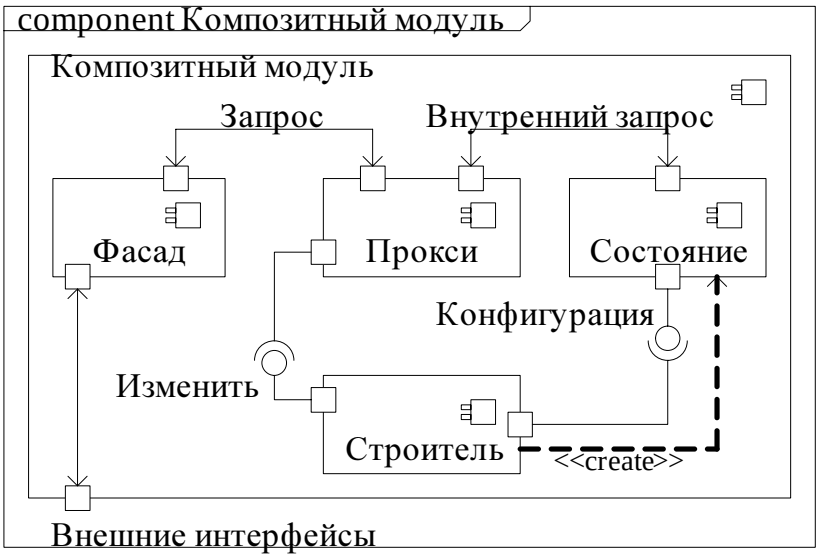

Рисунок 1 – Архитектура композитного модуля с использованием паттернов проектирования

Композитный модуль имеет многослойную архитектуру (рис. 2), которая реализует общую модель, используемую для разделения сложных систем [8].

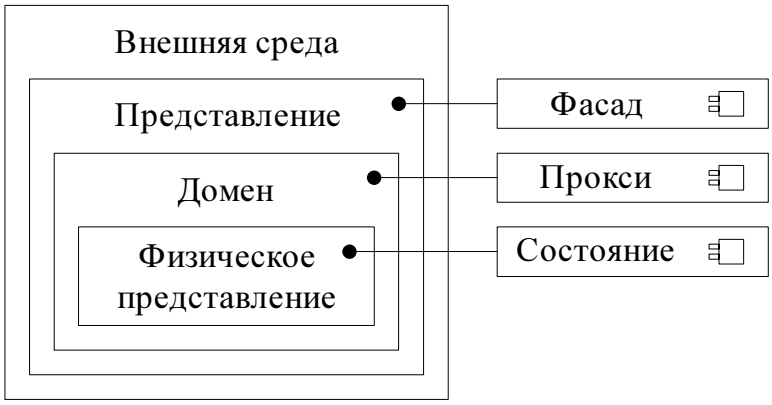

Рисунок 2 – Расслоение архитектуры композитного модуля

Физическое представление алгоритма, реализуемого композитным модулем, определяется текущим «состоянием» реконфигурируемой логики, являющимся элементом множества возможных состояний. «Прокси» реализует уровень предметной области алгоритма, обеспечивая правильность смены состояний (для физической смены состояний используется «строитель»), реализацию общих алгоритмов, поддержку механизмов оптимизации и т.д. Представление, реализуемое «фасадом», обеспечивает интерфейс взаимодействия с внешней средой. «Фасад» не обладает собственной логикой, но обеспечивает сопряжение и буферирование.

Для каждой конкретной архитектуры, базирующейся на архитектуре композитного модуля, реализуются собственные компоненты, отвечающие

шаблонной схеме (рис. 1) и выполняющие роли, определенные расслоением архитектуры (рис. 2).

Архитектура цифрового устройства управления, базирующаяся на архитектуре композитного модуля

Логика цифрового устройства управления (УУ) определяется конечным автоматом [6,7]. В каждый момент времени конечный автомат находится в одном из состояний, следовательно, на некотором промежутке времени активной является только некоторая часть автомата. Применение архитектуры композитного модуля (рис. 1) для проектирования цифрового управляющего устройства (рис. 3) базируется на том, что конечный автомат управления может быть подвержен декомпозиции с целью разделенного конфигурирования его секций, каждая из которых определяет состояние композитного модуля в некоторый момент времени.

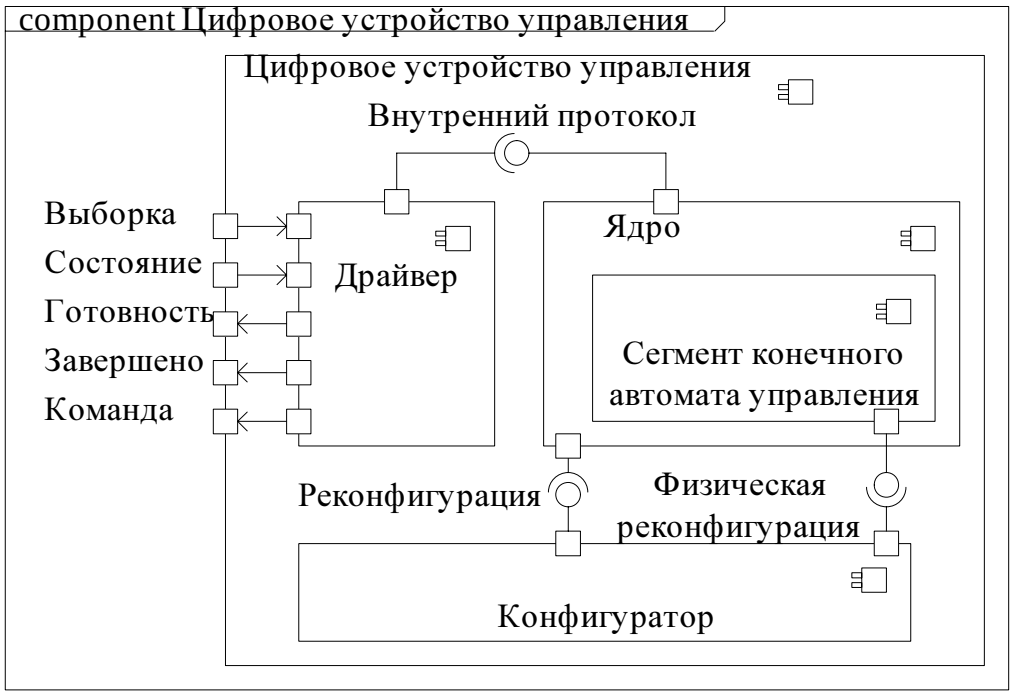

Рисунок 3 – Архитектура цифрового устройства управления, базирующаяся на архитектуре композитного модуля

Применение композитных модулей в данном аспекте обеспечивает следующие возможности:

- с физической стороны – уменьшение количества ресурсов, необходимых для развертывания управляющего автомата, при этом метрики экономии определяются свойствами исходного алгоритма и выбранного алгоритма декомпозиции;
- с логической стороны – обеспечение возможности изменения управляющего автомата, когда такое изменение ограничено физическими характеристиками устройства, предоставляющего ресурсы

реконфигурируемой логики, и предметной областью, определяемой внешней средой.

Ограничение физическими характеристиками устройства выполняется, если максимальное количество ресурсов, необходимое для развертывания каждой секции измененного управляющего автомата, не превышает максимальное количество ресурсов, необходимое для развертывания каждой секции исходного УА. Ограничение предметной областью выполняется, если измененный управляющий автомат поддерживает те же контракты, что и исходный УА.

Драйвер обеспечивает взаимодействие с внешними системами по асинхронному протоколу (также может применяться асинхронно-синхронный протокол [6, с. 245-248]), буферирование кода операции (команды) и флагов (состояния). Ядро реализует логику конечного автомата управления, сегментированного в результате декомпозиции алгоритма. Конфигуратор обеспечивает физическую реконфигурацию сегментов управляющего автомата в соответствии с логическими запросами, поступающими из ядра.

Тривиальный алгоритм работы цифрового устройства управления с композитной логикой

Ядро, базирующееся на реконфигурируемой логике, может находиться в одном из трех состояний [9]:

- операбельно – сегмент управляющего автомата сконфигурирован и актуален. Актуальность сегмента управляющего автомата определяется возможностью обработки следующего запроса выборки без реконфигурации сегмента;
- анализируется – сегмент управляющего автомата сконфигурирован, но неактуален. Ядро анализирует состояние системы, для определения следующего сегмента и его конфигурирования;
- реконфигурируется – в результате анализа выявлен следующий сегмент и происходит его конфигурация.

Если ядро использует один сегмент управляющего автомата, то на время анализа и реконфигурирования оно будет простаивать, вызывая снижение производительности (ядро простаивает и между получениями запросов на выборку команды, но при этом не происходит потери производительности) [9] (рис. 4).

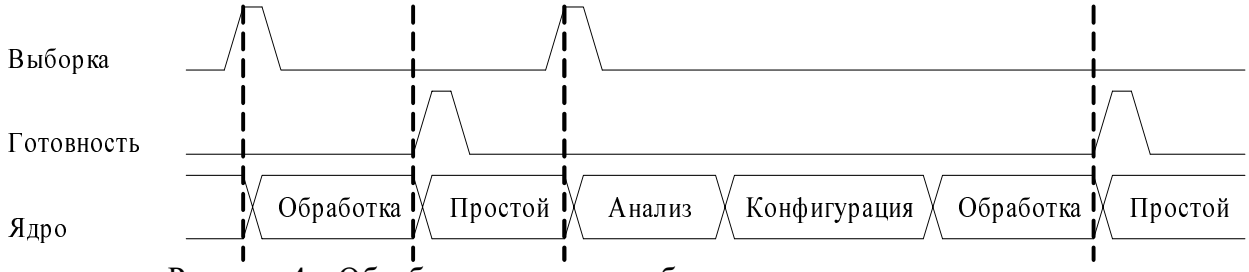

Рисунок 4 – Обработка запросов выборки односекционным ядром

Основным недостатком тривиального алгоритма (рис. 5) является изменчивость времени обработки запроса на выборку команды, что ведет к сложности сопряжения с другими системами, в частности, к сложности синхронизации. При этом также нарушается свойство прозрачности композитного модуля, т.к. процессы, которые обеспечивают реконфигурирование, оказывают явное влияние на внешние характеристики системы. Преимуществом тривиального алгоритма является его простота и надежность. Тривиальный алгоритм не требует наличия дополнительных подсистем (менеджеров секций, анализаторов) в ядре, уменьшая количество необходимых ресурсов. При применении тривиального алгоритма на способ декомпозиции конечного автомата управления не налагаются какие-либо ограничения и требования.

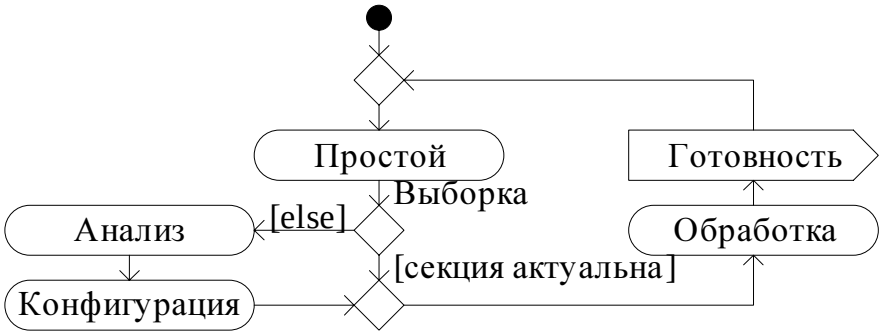

Рисунок 5 - Тривиальный алгоритм работы цифрового устройства управления с композитной логикой

Для оптимизации работы устройства можно применять ряд алгоритмов, в частности, описанных в [9]: использование параллельных сегментов; наложение ограничений на алгоритмы декомпозиции, для обеспечения упрощенного управления сегментами; стековая выборка последовательных команд и т.д. Однако эти алгоритмы приводят к усложнению архитектуры ядра, а также требуют применения специализированных алгоритмов декомпозиции.

Выводы

В статье предложена архитектура композитного модуля – аппаратного устройства, объединяющего жесткую и реконфигурируемую

логику. Описание предлагаемой архитектуры дано с использованием паттернов проектирования, что представляет собой попытку применения семантики паттернов, применяемых в программных системах, для аппаратных систем с использованием многослойной модели.

Показано применение архитектуры композитного модуля для проектирования цифрового устройства управления. При этом рассматриваются компоненты такой системы и тривиальный алгоритм ее работы, который является наиболее простым и надежным решением, но обладает рядом недостатков. Направление дальнейших исследований заключается в рассмотрении вопросов, связанных с архитектурой отдельных компонент предлагаемой системы и разработкой оптимизированных алгоритмов работы системы.

Список источников

1. Палагин А.В. Реконфигурируемые вычислительные системы: Основы и приложения. / А.В.Палагин, В.Н.Опанасенко. – Київ: Просвіта, 2006. – 280 с.: ил.
2. Буч Г., Якобсон А., Рамбо Дж. UML. Классика CS. 2-е изд.: Пер. с англ. Под общей редакцией проф. С. Орлова – СПб: Питер, 2006. – 736 с.: ил.
3. Баркалов А.А., Зеленева И.Я., Гриценко А.А. Адаптация MDA для моделирования управляющих автоматов в стандартах UML // Радіоелектроніка. Інформатика. Управління. – Запоріжжя; ЗНТУ, 1(15)’2006. – С. 33-37
4. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования. – СПб: Питер, 2006. – 366 с.: ил.
5. Neema S., Barty T., Scott J. Adaptive Computing and Runtime Reconfiguration // ISIS, Vanderbilt University; USA, 1999. – 8 pp.
6. Майоров С.А., Новиков Г.И. Принципы организации цифровых машин. – Львів: Машиностроение, 1974. – 432 с.
7. Баркалов О.О., Ковальов С.О., Мальчева Р.В. Проектування операційних пристроїв. – Донецьк: РВА ДонНТУ, 2005. – 312 с.
8. Фаулер М. Архитектура корпоративных программных приложений.: Пер. с англ. – М.: Изд. дом «Вильямс», 2006. – 544 с.
9. Grytsenko A., Malcheva R., Barkalov A. Run-time reconfigurable system for distributed algorithm execution // Proceedings of the International Conference on Computer Science and Information Technologies, CSIT’2006. – Lviv: September 28th-30th, 2006 – P. 164-166.