

На правах рукописи

Поздняков Вадим Александрович

**РАЗРАБОТКА МЕТОДОВ УНИВЕРСАЛЬНОЙ ОБРАБОТКИ
ЦИФРОВЫХ ИНФОРМАЦИОННЫХ ПОТОКОВ СТАНДАРТА DVB**

Специальность 05.12.13 — Системы, сети и устройства телекоммуникаций

АВТОРЕФЕРАТ

диссертации на соискание ученой степени

кандидата технических наук

Москва — 2010

Работа выполнена на кафедре электронных вычислительных машин
Института точной механики и вычислительной техники им С.А. Лебедева.

Научный руководитель: кандидат технических наук, старший научный
сотрудник

Кречетов Евгений Георгиевич

Официальные оппоненты: доктор технических наук, доцент

Сарьян Вильям Карпович

ФГУП НИИР

кандидат технических наук, доцент

Басок Борис Моисеевич

МИРЭА

Ведущая организация: **ЗАО «МЦСТ»**, г. Москва

Защита состоится 19 октября 2010 г. в 15⁰⁰ на заседании
диссертационного совета Д 212.156.04 при Московском физико-техническом
институте (ГУ) по адресу: 141700, г. Долгопрудный Московской обл.,
Институтский пер., д. 9, Новый корпус, ауд. 204.

С диссертацией можно ознакомиться в библиотеке Московского
физико-технического института (ГУ).

Автореферат разослан «__» сентября 2010 г.

Ученый секретарь

диссертационного совета Д 212.156.04,

кандидат технических наук, доцент

Куклев Л. П.

Общая характеристика работы

Актуальность темы. Телекоммуникационные сети на основе семейства европейских стандартов DVB, появившиеся в середине 90-х годов прошлого века, получили широкое распространение во всем мире. В частности, в 2004 году для внедрения на территории России в рамках программы по развитию телерадиовещания была выбрана именно эта технология.

Сети DVB постоянно развиваются, в них улучшаются уже имеющиеся услуги и появляются новые, поэтому соответствующие стандарты периодически дорабатываются. Несколько лет назад началась разработка более совершенных систем второго поколения. Спектр оборудования и программного обеспечения, используемого для создания и обслуживания сетей DVB, чрезвычайно широк.

Мультиплексирование логических потоков данных во всех DVB-совместимых системах реализуется с помощью сложноструктурированного цифрового потока (называемого транспортным потоком). Его обработку поддерживают все устройства и программные продукты, предназначенные для взаимодействия с сетями DVB. Алгоритмы обработки, используемые при этом, достаточно сложны и разнообразны. Их функциональность может зависеть от многих факторов: типа и назначения сетевого устройства, поддерживаемых стандартов, перечня предоставляемых услуг.

Постоянное обновление стандартов и стремительное расширение возможностей оборудования требует повышения универсальности методов обработки потоков DVB, используемых в наиболее функциональных устройствах. Под универсальностью в данном случае понимается возможность изменения и настройки алгоритмов обработки в широких пределах для решения разнообразных задач, а также возможность

выполнения одних и тех же алгоритмов на разных аппаратно-программных платформах.

Применение более универсальных методов обработки транспортных потоков DVB позволяет снизить себестоимость, уменьшить время разработки, расширить круг решаемых задач для ряда продуктов (например, профессиональных анализаторов и формирователей транспортных потоков), поэтому проблема повышения универсальности этих методов является актуальной в практическом плане.

В настоящее время некоторые из применяемых в рассматриваемой области методов позволяют решить задачу универсализации алгоритмов лишь частично, с помощью шаблонов и интерпретируемых языков для описания структур данных. Методы, применяемые в смежных областях, плохо подходят для практического использования по различным причинам: неудобный синтаксис описания алгоритмов обработки, сложность реализации посредника между алгоритмами и системой, в которой они должны выполняться, ограничения при переносе алгоритмов на другие платформы и повторном использовании их частей.

Таким образом, из-за отсутствия адекватных методов реализации универсальной обработки транспортных потоков DVB, данную проблему следует признать недостаточно разработанной и, следовательно, актуальной с точки зрения науки.

Цель и задачи исследования. В работе проводится исследование методов создания и выполнения алгоритмов обработки цифровых потоков стандарта DVB с целью разработки методов, позволяющих повысить универсальность обработки потоков данного типа.

Задачами данной работы являются:

- классификация задач обработки транспортных потоков DVB;
- определение критериев универсальности алгоритмов обработки;

- исследование существующих методов создания и выполнения алгоритмов обработки, выявление их достоинств и недостатков;
- разработка методов повышения универсальности обработки;
- разработка методов создания универсальных алгоритмов обработки транспортных потоков DVB.

Методология исследования. Из используемых в методологии технических наук методов в данной работе наиболее широко применяется системный подход: реализация алгоритмов обработки транспортных потоков рассматривается как система, состоящая из взаимосвязанных компонентов, которая, в свою очередь, является подсистемой общей системы обработки. Определяются элементы, составляющие такую систему, связи и зависимости между ними, функции системы и ее частей, проистекающие в системе процессы, внешние связи и т. п.

Кроме системного подхода в работе применяются также абстрактно-логический (создание теоретической основы универсальной системы обработки) и проектный (порядок практической реализации всех необходимых программных продуктов) методы.

Научная новизна. В диссертации получены следующие новые научные и практические результаты:

- Сформулированы критерии универсальности алгоритмов обработки.
- Предложен новый метод создания алгоритмов обработки транспортных потоков DVB на основе модифицированного принципа крупномодульной потоковой обработки.
- Разработана универсальная система обработки потоков данных, которая по ряду характеристик превосходит существующие системы потокового программирования.
- Предложен метод выбора типов потоков данных между отдельными модулями обработки.

- Разработаны методы создания алгоритмов, удовлетворяющих всем критериям универсальности и позволяющих решать основные задачи обработки транспортных потоков DVB.

Практическая ценность результатов работы. Задачи, решаемые в данной работе, имеют практическую направленность, и полученные результаты в виде алгоритмов обработки цифровых потоков DVB использованы при разработке продукции ООО «Телекоммуникации, Технологии, Информация» (что подтверждено соответствующим актом):

- В аппаратно-программном комплексе приема цифровых потоков спутниковых систем связи VSAT: ячейка приема сигнала от базовой станции с функцией обработки цифровых потоков DVB.
- В интегрированной программной среде анализа структуры цифровых потоков: компоненты анализа и обработки цифровых потоков DVB.

В работе приведены практические рекомендации, которые могут быть полезны специалистам в рассматриваемой области и смежных с ней областях. Они могут быть также использованы при обработке и исследовании свойств цифровых потоков в других типах сетей передачи данных.

Апробация результатов исследования. Основные положения и результаты работы докладывались на 49-й научной конференции МФТИ (Москва, 2006), на 14-й международной открытой научной конференции «Современные проблемы информатизации» (Воронеж, 2009), на международной конференции молодых ученых «Актуальные задачи современной науки» (Красноярск, 2009), на научном семинаре кафедры радиотехники МФТИ (Москва, 2010).

Публикации. Материалы исследований представлены в 6 печатных работах: 3 статьи в периодических научных изданиях, одно из которых

входит в перечень ВАК, и 3 публикации в виде тезисов докладов. Общий объем публикаций по теме диссертации составляет 1,5 п. л.

Структура и объем работы. Диссертационная работа состоит из введения, четырех глав, заключения и списка литературы, включающего 73 наименования. Работа изложена на 143 страницах, она содержит 20 рисунков и 13 таблиц.

Положения, выносимые на защиту:

1. Метод создания алгоритмов обработки на основе модифицированного принципа крупномодульной потоковой обработки.
2. Архитектура, алгоритм работы и особенности реализации универсальной системы обработки потоков данных.
3. Метод выбора типов потоков данных между модулями.
4. Методы создания универсальных алгоритмов обработки транспортных потоков DVB.

Основное содержание работы

Во **введении** обоснована актуальность выбранной темы, сформулированы цели и задачи исследования, показаны научная новизна, теоретическая и практическая значимость результатов работы.

В **первой главе** рассматриваются проблемы, возникающие при создании алгоритмов обработки транспортных потоков DVB. Приводится классификация задач обработки, формулируются критерии универсальности алгоритмов, на основе известных автору источников рассматриваются особенности современных методов создания и выполнения алгоритмов обработки потоков данных, вводятся основные понятия.

Семейство открытых стандартов DVB, описывающих системы и методы распространения различных цифровых услуг (телевидение, радио, передача данных и др.), в последние годы развивается стремительными темпами: постоянно увеличивается качество и количество поддерживаемых услуг, улучшается взаимодействие с другими сетями. В 2005 году начали появляться стандарты DVB второго поколения.

В целях облегчения обмена данными между DVB-совместимыми системами разных типов (спутниковых, наземных, мобильных, кабельных), все они должны использовать единую среду мультиплексирования логических потоков данных — транспортный поток. Он представляет собой поток данных с байтовой структурой, который формируется передающей системой из нескольких элементарных потоков изображения и звука (телевизионные программы) или данных, а также служебной информации, для последующего кодирования, модуляции и передачи по физической среде.

В системах DVB первого поколения используется транспортный поток MPEG-2, состоящий из пакетов одинаковой длины (188 байтов), в системах второго поколения может также использоваться так называемый общий поток, который состоит из пакетов переменной длины.

Служебная информация в транспортном потоке DVB передается с помощью ряда стандартных информационных таблиц. В них содержится информация о программных потоках, которая служит для настройки демультимплексора на необходимую программу (пять типов таблиц), и дополнительная информация о самих программах (девять типов таблиц).

Если в транспортном потоке присутствуют услуги, отличные от стандартных, он может также содержать дополнительные таблицы, формат которых описан в соответствующих документах. Набор и структура таблиц для второго поколения стандартов DVB пока не определены.

В некоторых таблицах предусмотрено размещение дескрипторов — элементов данных с упрощенной структурой. Дескрипторы содержат дополнительную информацию о предоставляемых услугах. На сегодняшний день уже насчитывается порядка сотни различных типов дескрипторов.

В общем случае обработка транспортного потока является достаточно сложной задачей, так как в нем используются взаимосвязанные таблицы, дескрипторы и числовые идентификаторы для корректного выделения элементарных потоков и определения их свойств. По мере расширения функциональности технологии DVB, в стандарты добавляются описания новых информационных структур и принципов работы с ними.

Автором составлен перечень наиболее распространенных задач обработки транспортных потоков DVB, которые разделены на три группы: стандартные, дополнительные и исследовательские. К первой группе относятся ключевые задачи обработки, решаемые во всех DVB-совместимых системах. Дополнительные задачи возникают в системах, рассчитанных на предоставление нестандартных услуг (например, передачи данных). Исследовательские задачи, как правило, связаны с анализом качества передачи и поиском неполадок.

Делается вывод, что обработка транспортного потока DVB подразумевает целый ряд взаимосвязанных операций, часть из которых

достаточно хорошо определена (стандартные и большинство дополнительных задач), другая же их часть определена лишь в общих чертах и требует гибкой реализации, или не определена вовсе.

Потребность в уменьшении издержек на разработку и модернизацию оборудования и программного обеспечения для обработки транспортных потоков DVB приводит к универсализации используемых для этого алгоритмов. Автором сформулированы четыре критерия универсальности алгоритмов обработки:

- возможность использования одних и тех же алгоритмов в различных системах обработки;
- возможность повторного использования частей алгоритмов;
- возможность гибкого изменения алгоритмов для решения других задач;
- возможность расширения функциональности алгоритмов без существенной коррекции уже имеющихся частей.

Рассматриваются наиболее распространенные в настоящее время методы создания алгоритмов обработки потоков данных в телекоммуникационных системах (использование процедурных и объектно-ориентированных языков программирования, в том числе с компиляцией в промежуточный байт-код для виртуальной машины, шаблонов, специализированных и формальных языков).

Анализируются недостатки каждого метода, делается вывод, что ни один из них не подходит для создания алгоритмов, соответствующих критериям универсальности в полной мере. Таким образом, на сегодняшний день не существует общедоступных средств для создания достаточно универсальных алгоритмов обработки транспортных потоков DVB.

С учетом специфики задач обработки транспортных потоков DVB и недостатков существующих средств разработки формулируются основные требования к новым средствам, которые позволят создавать и выполнять

универсальные алгоритмы. В частности, для выполнения алгоритмов предлагается использовать виртуальную машину с урезанным набором команд, а сам процесс обработки должен представлять собой обмен данными между отдельными модулями, функциональность которых следует описывать с помощью специализированного языка высокого уровня с последующей компиляцией в байт-код.

Средства выполнения алгоритмов (виртуальную машину, системы управления процессом обработки и передачей данных между модулями) автор предлагает называть универсальной системой обработки потоков данных (сокращенно УСОПД), понимая при этом под универсальностью возможность выполнять алгоритмы обработки транспортных потоков DVB, соответствующие всем сформулированным ранее критериям универсальности.

Вторая глава посвящена подробному рассмотрению концепции УСОПД. Реализованная отдельно или как часть более сложной системы обработки потоков данных, УСОПД по своей сути является «черным ящиком» с входным, выходным и управляющим интерфейсами. Она позволяет выполнять алгоритмы обработки, загружаемые в нее извне, контролировать процесс их выполнения и производить обмен информацией с внешним миром. УСОПД с загруженными в нее алгоритмами обработки предлагается называть универсальным обработчиком.

Сам алгоритм обработки представляется в виде ориентированного графа, вершинами которого являются элементарные обработчики (модули), а ребрами — потоки данных между вершинами. Такой подход к созданию алгоритмов известен как крупномодульное потоковое программирование (large-grain dataflow programming).

Вводится понятие слоя — это группа модулей, которые могут выполняться одновременно или в произвольном порядке. Слои служат для задания последовательности запуска модулей, а также для логического

разделения модулей по функциональному назначению. Обратная связь, при необходимости, реализуется с задержкой на один такт обработчика. На рис. 1 изображен пример логического устройства универсального обработчика.

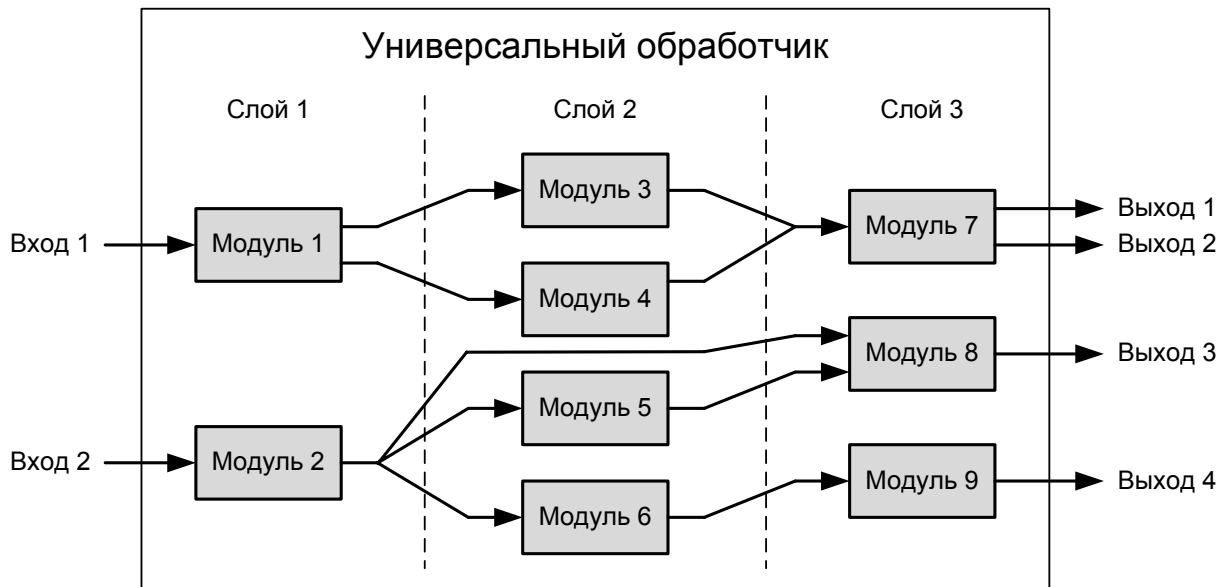


Рис. 1. Пример универсального обработчика

Единицей обмена информацией между модулями служит элемент данных. Он представляет собой массив байтов с заданным размером, в котором последовательно записан некоторый набор значений. Тип (и, соответственно, размер в байтах) каждого значения определяется модулем-источником, который порождает элементы данных и помещает их в выходной поток.

Поток — это последовательность элементов данных, он выполняет функцию буфера при передаче данных между модулями. Поток реализуется в виде очереди, в конец которой добавляются элементы данных, порождаемые модулями-источниками. Один или несколько модулей-приемников получают элементы данных из начала очереди.

Совокупность модулей и связывающих их потоков элементов данных называется сборкой. Фактически сборка представляет собой реализацию

общего алгоритма обработки — УСОПД с загруженной сборкой становится универсальным обработчиком.

Модуль, как элементарный обработчик, должен выполнять следующие операции: чтение элементов данных из каждого входного потока, их обработка и формирование элементов данных для каждого выходного потока. Перед началом обработки модуль должен быть установлен в начальное состояние (которое может зависеть от значений его параметров).

Предлагается разделение байт-кода модуля на три части: код инициализации, коды обработки входов и итоговый код. Коды инициализации всех модулей сборки выполняются перед началом обработки, они служат для установки их в начальное состояние. Код обработки входа выполняется для каждого нового элемента данных на этом входе.

Когда все коды обработки входов для всех новых элементов данных выполнены, выполняется итоговый код. Как правило, он используется для совместной обработки данных из двух и более входов и обобщения результатов обработки. Формирование элементов данных для выходных потоков может производиться как в кодах обработки входных потоков, так и в итоговом коде.

В общем случае УСОПД состоит из трех основных компонентов: системы управления процессом обработки, виртуальной машины и системы управления данными (см. рис. 2). Обмен данными и управляющими командами с внешней системой, частью которой является УСОПД, осуществляется с помощью управляющего, входного и выходного интерфейсов.

Система управления процессом обработки получает через управляющий интерфейс файл со сборкой, преобразует его в удобный для дальнейшей работы вид и помещает в специально отведенную область оперативной памяти. В области временных данных выделяет память для

переменных и параметров модулей, там же хранятся элементы данных всех потоков.

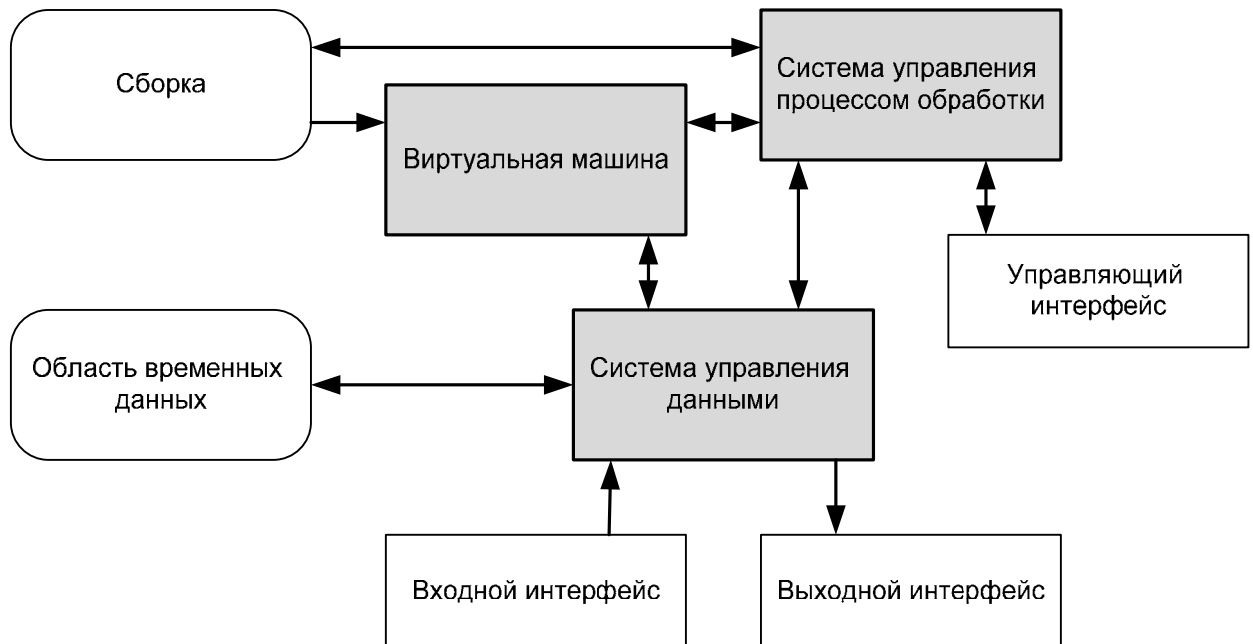


Рис. 2. Архитектура УСОПД

Когда все подготовительные операции завершены, внешняя система передает через управляющий интерфейс команду инициализации обработчика, затем через входной интерфейс на входы обработчика помещаются элементы данных, которые система управления данными копирует в область временных данных, и дается команда обработки: выполняются все модули, у которых хотя бы на одном из входов присутствуют элементы данных, соблюдая очередность слоев.

Все операции с переменными, параметрами и элементами данных виртуальная машина осуществляет посредством системы управления данными. После выполнения всех байт-кодов обработки входов модуля для всех элементов данных на этих входах, виртуальной машине передается команда на выполнение итогового байт-кода — на этом процесс выполнения модуля заканчивается. Во время выполнения обработки производится

обработка возможных ошибок (нехватка памяти, зависание модуля, ошибки при выполнении байт-кода и др.).

Алгоритм работы УСОПД, подробно описанный в тексте диссертации, достаточно прост и легко реализуем; кроме того, реализация самой УСОПД не представляет значительной сложности: необходимо создать и связать друг с другом три интерфейса и три функциональных блока, два из которых (системы управления процессом обработки и данными) выполняют несколько простых операций, а третий (виртуальная машина) представляет собой интерпретатор небольшого набора элементарных команд.

В **третьей главе** подробно рассматривается практический пример программной реализации УСОПД и сопутствующего программного обеспечения. Он предназначен для доказательства реализуемости универсальной системы, демонстрации подхода к ее созданию и получения инструментария для последующей разработки и тестирования универсальных алгоритмов.

Так как использование УСОПД должно уменьшать издержки при создании и модификации алгоритмов обработки для одной или нескольких платформ, сначала необходимо разработать единую систему команд байт-кода, общие форматы файлов описаний модуля и сборки, а также общий язык высокого уровня для описания модулей.

В этом случае потребность в создании компилятора и сборщика возникнет лишь единожды, кроме того, все реализации УСОПД будут поддерживать одну и ту же систему команд байт-кода и одинаковый формат файла сборки, следовательно, уменьшатся издержки на реализацию виртуальных машин и систем управления процессом обработки для различных внешних систем. Реализации УСОПД будут отличаться друг от друга в основном системами управления данными и интерфейсами с внешними системами.

Система команд байт-кода модулей должна обеспечивать реализацию всех необходимых функций обработки, однако количество и сложность команд необходимо минимизировать, чтобы упростить реализацию виртуальных машин для каждой системы.

В рассматриваемом примере выбрана виртуальная машина регистрового типа, в которой используются восемь 32-разрядных регистров общего назначения и регистр флагов. Система команд построена на основе ассемблера семейства микропроцессоров Intel x86. Исходный набор команд избыточен для задач, решаемых УСОПД, поэтому лишние команды были опущены. Кроме того, из-за особенностей передачи данных между модулями и необходимости обработки ошибок, введено несколько новых команд.

Каждая команда представляется в виде однобайтового кода команды, за которым следуют один или несколько байтов с операндами. В качестве операндов выступают регистры, переменные или константы. Общее число команд в рассматриваемой реализации — 36.

Полностью отсутствуют команды работы с указателями. Это связано с тем, что в данной реализации УСОПД весь набор переменных (а также структура потоков на входе и на выходе модуля) и, соответственно, их адреса и размеры должны быть известны заранее, во время выполнения кода не разрешается динамическое выделение памяти.

Значительно уменьшить сложность виртуальной машины позволяет отказ от работы с числами со знаком, так как многие современные телекоммуникационные стандарты, в том числе и DVB, в подавляющем большинстве случаев оперируют с беззнаковыми целыми числами. Необходимость трактовать какую-либо последовательность битов как число со знаком возникает крайне редко, для обработки подобных ситуаций не обязательно иметь инструментарий для работы с числами в дополнительном коде — с помощью общеизвестных операций можно определить знак такого числа и получить его модуль. Неудобства при создании кода в этих случаях

минимальны, а выигрыш очень существенен (уменьшение количества команд примерно вдвое).

Файл, в котором хранятся данные о сборке, должен содержать общую информацию о сборке, а также описания модулей, потоков и их взаимосвязей. Описание модуля содержит общую информацию о нем и семь областей: переменные, параметры, входные потоки, выходные потоки, код инициализации, итоговый код и коды обработки входных потоков.

Описание потока содержит информацию о полях элементов данных этого потока, а также о модулях-источниках и модулях-приемниках. Потокам, модулям (в рамках сборки), переменным и параметрам (в рамках модуля) соответствуют уникальные числовые идентификаторы. Форматы описаний модулей и сборок разработаны с учетом возможности последующей модификации (при изменении функциональности УСОПД).

Описание модуля УСОПД является результатом компиляции кода модуля, написанного на высокоуровневом языке программирования. Этот язык должен иметь простой и понятный синтаксис, предоставлять необходимый минимум инструкций для описания основных операций с данными и удобный инструментарий для чтения исходных данных из входных потоков и записи результатов обработки в выходные потоки, в том числе инструкции для последовательного чтения/записи заданного числа битов в одномерных байтовых массивах. В качестве примера рассмотрен язык, за основу которого взят синтаксис языка Си.

В главе описан класс, реализующий УСОПД, который используется для создания объектов-обработчиков. Конструктору этого класса в качестве параметра передается сборка (массив байтов). В теле конструктора реализована обработка этой сборки и заполнение соответствующих структур данных.

Методы класса по их назначению разделены на три группы: методы, предоставляющие информацию о сборке (потоки, элементы данных в потоке,

слои, модули в слое, входы, выходы, переменные и параметры каждого модуля); методы инициализации (задание начальных значений параметров, инициализация модулей, сброс обработчика); методы управления процессом обработки в нескольких режимах.

При разработке рассмотренной реализации УСОПД основной целью являлось создание на ее основе алгоритмов обработки транспортных потоков DVB, удовлетворяющих критериям универсальности. Однако в приложениях, где может быть использована УСОПД, максимизация универсальности алгоритмов не всегда является наиболее приоритетной задачей. Поэтому в некоторых случаях возможна реализация УСОПД с некоторыми изменениями, которые позволяют получить требуемые характеристики.

В диссертации рассмотрены модификации универсальной системы, которые позволяют повысить скорость обработки (оптимизация системы команд, дублирование кодов обработки в виде функций внешней системы), еще более упростить реализацию УСОПД (отказ от использования некоторых возможностей) и расширить функциональность обработчиков (расширение системы команд и перечня типов данных, вызов внешних функций, оперативная обратная связь между модулями).

На рис. 3 показан процесс создания обработчика на основе УСОПД. Соответствующий программный инструментарий для создания, тестирования и выполнения универсальных алгоритмов обработки должен включать в себя компилятор языка описания модулей, сборщик и реализацию УСОПД для некоторой внешней системы. Кроме того, на этапе разработки компилятора и сборщика необходимо средство контроля правильности записи модулей и сборок в файлы. В диссертации приведены описания функциональности и интерфейса программных компонентов, которые использовались для разработки всех модулей иборок.

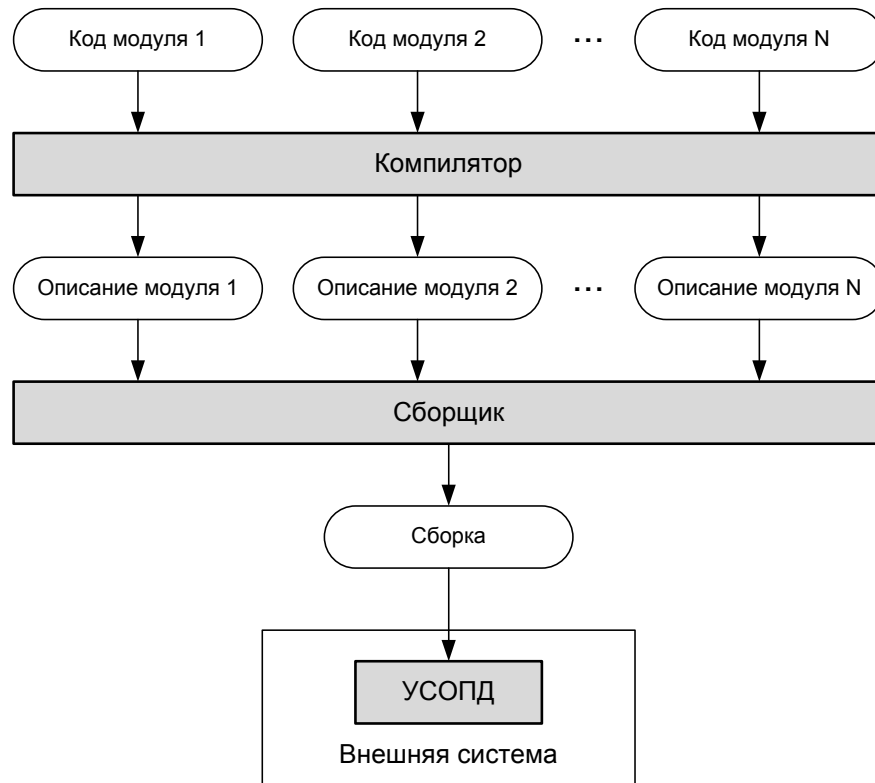


Рис. 3. Процесс создания обработчика

В конце главы рассмотрены основные преимущества, которые может дать использование УСОПД для реализации алгоритмов обработки транспортных потоков DVB на примере анализатора транспортных потоков. Функциональность такого анализатора может быть в любой момент расширена или изменена пользователем без привлечения производителя (нет необходимости ожидать обновления программного обеспечения), при этом могут редактироваться алгоритмы обработки; можно также добавить функции, на поддержку которых анализатор изначально не был рассчитан.

При возникновении проблемы совместимости устройств разных производителей или при поиске неполадок в работе устройства использование УСОПД позволяет не только выявлять подробности возникновения проблем, но и в некоторых случаях проверять возможные пути их решения непосредственно с помощью анализатора потоков.

В четвертой главе рассматривается методика создания универсальных алгоритмов обработки транспортных потоков DVB, на практических примерах иллюстрируются подходы к организации обработки различных структур данных, показывается, что предложенные алгоритмы соответствуют критериям универсальности.

Перед началом любого проекта, связанного с применением УСОПД, необходимо определить перечень всех типов потоков данных, которые теоретически могут быть использованы при взаимодействии между модулями. Затем, если возможно, этот перечень необходимо сократить, заменяя несколько потоков, которые имеют пересекающиеся множества полей, одним потоком.

При этом не следует допускать излишнего усложнения структуры полей потоков, а также объединять потоки, которые используются на различных стадиях обработки. Другими словами, необходимо найти компромисс между количеством модулей и их сложностью.

В диссертации предложен и продемонстрирован на примере метод выбора типов межмодульных потоков УСОПД, которые могут использоваться при обработке транспортных потоков DVB. Сначала необходимо обобщить информацию о структурах данных и производимых с ними действиях. Для этого строятся таблицы, в столбцах которых указываются все структуры данных, а в строках — все элементарные операции и преобразования, которые совершаются с ними в процессе обработки.

Затем с помощью этих таблиц составляется избыточный перечень типов потоков данных — каждый тип однозначно определяется структурой данных и действием, и может быть помещен в ячейку таблицы. Далее, на основе анализа ветвей и стадий алгоритмов обработки, некоторые типы потоков из избыточного перечня объединяются в группы.

Наконец, для каждой такой группы формируется новый тип потока, который заменяет все типы группы, объединяя множества их полей. В рассмотренном примере исходный перечень, содержащий более 40 типов потоков, удалось сократить до 10 (два из которых имеют подтипы).

Далее в диссертации подробно разобраны все стадии обработки транспортного потока DVB, приведены решения типичных задач с использованием рассмотренных типов потоков, описаны функциональность и интерфейс соответствующих модулей. Даны рекомендации по построению структур обработчиков для решения сложных задач обработки.

Первой описывается стадия обработки транспортных пакетов, приводится пример использования языка описания модулей. Рассматриваются четыре наиболее распространенных операции: фильтрация/ветвление по значению полей заголовка, склейка пакета следующего уровня из полезных нагрузок транспортных пакетов, обработка поля адаптации и сбор статистики.

Обработка разделов информационных таблиц должна производиться в два этапа: сначала анализируются поля общего заголовка, затем — поля, характерные для каждого типа таблиц. Рассмотрены особенности обработки двух важнейших типов таблиц, которые присутствуют практически в каждом транспортном потоке: PAT (информация о соответствии номеров программ и логических потоков транспортных пакетов с таблицами PMT) и PMT (перечень логических потоков, входящих в состав программы).

Для каждой стадии обработки приводится схема обработчика УСОПД, описываются структуры используемых потоков и функциональность всех модулей. Пример такой схемы для стадии обработки информационных таблиц приведен на рис. 4.

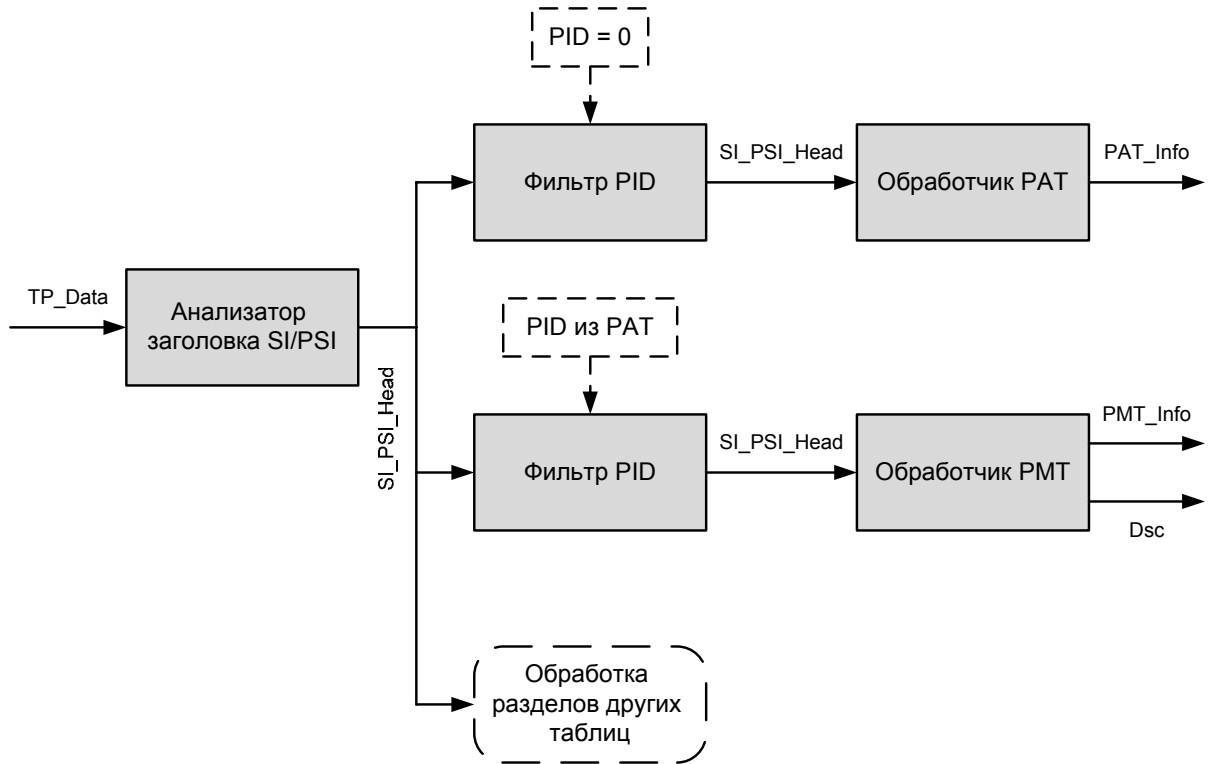


Рис. 4. Пример реализации стадии обработки разделов таблиц

Отдельно рассматривается обработка дескрипторов и разделов MPE (Multi-Protocol Encapsulation), которые используются для передачи пакетов данных различных протоколов более высокого уровня, они отличаются от разделов информационных таблиц заголовком (содержащим физический адрес приемника) и порядком обработки.

Для демонстрации подхода к созданию сложного обработчика, позволяющего в полной мере использовать преимущества УСОПД, выбрана реализация одной из наиболее востребованных функций профессиональных анализаторов транспортных потоков DVB — поиск различных ошибок и несоответствий стандартам. Рассмотрен обработчик, реализующий испытания первого приоритета (наиболее важные, оценка правильность приема транспортного потока как такового).

Для его создания использовались описанные ранее модули и межмодульные потоки с необходимыми изменениями и дополнениями, а также новые модули и потоки, когда функциональности имеющихся было

недостаточно. Показывается, что полученный обработчик (а также обработчики, рассмотренные ранее) удовлетворяет первым трем критериям универсальности.

Чтобы показать соответствие обработчика последнему критерию универсальности, его функциональность была расширена — добавлена поддержка наиболее нетривиального испытания третьего приоритета, целью которого является поиск скрытых логических потоков. Этому удалось достичь без каких-либо изменений существующей части обработчика, что полностью согласуется с последним критерием универсальности. Подобная гибкость обеспечивается удачным выбором типов потоков и правильностью разделения функциональности между отдельными модулями.

Рамки работы не позволяют описать все ситуации, с которыми можно столкнуться в процессе реализации обработчиков, тем не менее, общность принципов представления, структурирования и использования данных, заложенных в основу всех стандартов DVB, позволяет утверждать, что рассмотренные подходы к созданию алгоритмов будут применимы и во многих других случаях.

В заключении сформулированы основные научные выводы и практические результаты работы, которые состоят в следующем:

1. Сформулированы критерии универсальности алгоритмов обработки транспортных потоков DVB и требования к средствам для создания и выполнения таких алгоритмов.

2. Предложена концепция универсальной системы обработки потоков данных с байтовой структурой, на основе которой можно реализовать алгоритмы, в полной мере удовлетворяющие критериям универсальности. Подробно рассмотрены все особенности УСОПД: архитектура, алгоритм работы, логическое устройство обработчиков и их составных частей. Использование УСОПД в системах обработки

транспортных потоков DVB позволяет существенно расширить их функциональность.

3. Рассмотрен пример практической реализации УСОПД: выбрана система команд байт-кода, приведены структуры файлов модуля и сборки, рассмотрены язык описания модулей на основе синтаксиса Си и структура класса, реализующего УСОПД, описан состав программного обеспечения, необходимого для создания и выполнения алгоритмов на основе УСОПД; рассмотрены варианты модификаций УСОПД, удовлетворяющие различным требованиям.

4. Описана методика создания универсальных алгоритмов обработки транспортных потоков DVB: предложен метод выбора типов межмодульных потоков в обработчиках на основе УСОПД, рассмотрены структуры и предназначение наиболее часто встречающихся типов данных, даны рекомендации по построению архитектуры соответствующих обработчиков, приведены примеры обработчиков с подробными описаниями; показано, что приведенные алгоритмы удовлетворяют сформулированным ранее критериям универсальности.

Список публикаций по теме диссертации

1. Поздняков В. А. Метод выбора типов связей между модулями при решении задач обработки потоков данных // Естественные и технические науки. – М.: Компания Спутник+, 2009. – № 1. – С. 250-251.
2. Поздняков В. А. Основные концепции универсальной системы обработки потоков данных // Труды 49-й научной конференции МФТИ. – Долгопрудный, 2006. – С. 92-93.
3. Поздняков В. А. Выбор системы команд в рамках универсальной системы обработки потоков данных // Сборник научных трудов ИТМиВТ. I квартал 2007 года. – М., 2007. – С. 66-69.
4. Поздняков В. А. Особенности языка программирования, предназначенного для описания модулей в рамках универсальной системы обработки потоков данных // Наука и образование: электронное научное издание. – М., 2008. – № 12. – URL: <http://technomag.edu.ru/doc/111599.html> (дата обращения: 22.03.2010).
5. Поздняков В. А. Модификации универсальной системы обработки потоков данных // Современные проблемы информатизации в проектировании и информационных системах: Сб. трудов. Вып. 14. – Воронеж: Научная книга, 2009. – С. 530-532.
6. Поздняков В. А. Структура универсальных алгоритмов обработки транспортных потоков DVB // Материалы Международной научной конференции молодых ученых «Актуальные задачи современной науки» (апрель 2009 г.). – Красноярск: Научно-информационный издательский центр, 2009. – С. 22-23.