

МОДЕЛЬ ОПЕРАЦИОННОГО УСТРОЙСТВА РЕККУРЕНТНОГО ОБРАБОТЧИКА СИГНАЛОВ

Морозов Н.В., Андреев Р. В.

Выбор модели для нетрадиционной архитектуры

При разработке микропроцессора всегда встает задача анализа и исследования различных сторон его функционирования. Современный подход к проектированию новых электронных изделий заключается в создании программной модели разрабатываемого устройства для этих целей. С ее помощью можно не только проверить истинность принципов, лежащих в основе разрабатываемой архитектуры, но и даже создавать программное обеспечение еще до того, как готовое устройство увидит свет. Рассмотрим несколько подходов к созданию подобных программных моделей для архитектуры устройств рекуррентной обработки сигналов (РОС).

Программная логическая модель архитектуры микропроцессора позволяет на стадии проектирования оценить все аспекты применяемых архитектурных решений: производительность, узкие места, цену. Обычно такая модель детально отражает все процессы при функционировании устройства. Например, цикл обработки команды может содержать выборку команды, разбор операндов, формирование адресов исходных данных, выполнение операции, формирование адресов результирующих данных, передачу и запись результирующих данных. Моделирование каждой из этих операций связано с функционированием множества составляющих компонент: регистры, АЛУ, всевозможные шины, мультиплексоры, адресные устройства и т.д. Сами компоненты могут иметь достаточно сложные модели. Построение такой полной и детальной модели достаточно трудоемко, но дает исчерпывающий инструмент исследования различных аспектов функционирования устройства и позволяет уверенно подойти к этапам электрического моделирования и собственно производству устройства

Таким образом, для логического моделирования разрабатывается подробная и полная программная модель функционирования будущего устройства. Достоинства: полнота функционирования и максимально доступный анализ решений. Недостатки: трудоемкость, высокая стоимость и медленное моделирование, что может затруднить длительные испытания или обработку больших по объему данных.

Другой целью моделирования архитектуры микропроцессоров является проверка функциональных решений (набор устройств - АЛУ, регистры, периферия, память: шины; набор операций; общая производительность), а также подготовка программ на стадии проектирования устройства. Такая симулирующая модель менее детальна, более проста и эффективна. Однако, с ее помощью нельзя анализировать детальное функционирование, а только снимать общие параметры и подготавливать алгоритмы к исполнению на микропроцессоре. Основным требованием к ней является точность и скорость симулирования для разработки больших и сложных приложений с ее помощью.

Специфика РОС [1] предполагает построение архитектуры устройства из нескольких уровней [2]. На нижнем уровне архитектуры находится устройство операционного уровня (операционное устройство - ОУ), непосредственно реализующее рекуррентную обработку сигналов. Основная задача этого уровня дать возможность программировать на языке высокого уровня и управлять их выполнением.

На данном этапе разработки архитектуры ОУ РОС наиболее важным является исследование всех аспектов функционирования ОУ, следовательно для всестороннего исследования нижнего уровня РОС требуется использование логической модели. Статья

посвящена построению такой модели нижнего уровня - непосредственно модели ОУ. Данная модель должна:

- отражать физическую структуру ОУ, т.е. содержать модели составляющих ее устройств - РАПРЕДЕЛИТЕЛЯ, ЭКСПЛИКАТОРА, СЕКЦИЙ и т.д.;
- оперировать реальными операндами, содержащими данные с тегами, поддерживать их обработку и прохождение через устройство;
- создавать выходной набор данных и формировать капсулу для передачи на верхний уровень;
- функционировать потактово. В каждом такте модель ОУ, управляет порядком работы устройств в соответствии с их состоянием.

Такая модель соответствует классу полных моделей. При этом она должна быть функционально легко расширяема и обладать возможностью просто заменять устройства. Для реализации модели ОУ взята полная логическая модель. Рассмотрим подробнее структуру выбранной модели.

Структура программной модели ОУ РОС

На начальном этапе разработки ОУ РОС помощь в разработке структуры, путем моделирования устройства, выполнял программный комплекс ОПЕРА. С его помощью были исследованы процессы, протекающие при организации потока самоопределяемых данных, исследованы и оптимизированы реальные алгоритмы, получены целочисленные рекуррентные последовательности с заданными характеристиками, освоены методы кодирования алгоритмов в виде условий, управляющих генерацией рекуррентных последовательностей. Ознакомиться более подробно с целями создания программы, а также с результатами, достигнутыми с её помощью можно в [3].

Однако, для проведения дальнейших этапов проектирования ОУ программа ОПЕРА перестала удовлетворять растущим требованиям из-за низкого быстродействия, ориентации на символическое представление данных (операндов) и трудностей с дальнейшей модернизацией.

В связи с этим разработана структура новой программы моделирующей ОУ. Данная программная модель является частью, как минимум, двухуровневой модели РОС – операционного и управляющего уровней. В настоящее время в качестве межуровневого интерфейса выступает капсула, сформированная на управляющем (верхнем) процедурном уровне.

Основным недостатком предыдущей модели является тесная взаимосвязь вычислительного ядра и средств визуализации. Оперативное изменение программы и корректировка функций составляющих устройств давались с большим трудом. Поэтому новая версия программной модели построена по принципу разделения на две независимые части.

Проектирование структуры модели ОУ и моделей всех составляющих устройств проводилось в терминах классов с учетом приведенных выше двух основных требований:

- изменения состава моделируемого ОУ параметрически;
- легкости сопровождения модели, т.е. простота замены одного устройства другим и добавления принципиально новых устройств в структуру модели ОУ.

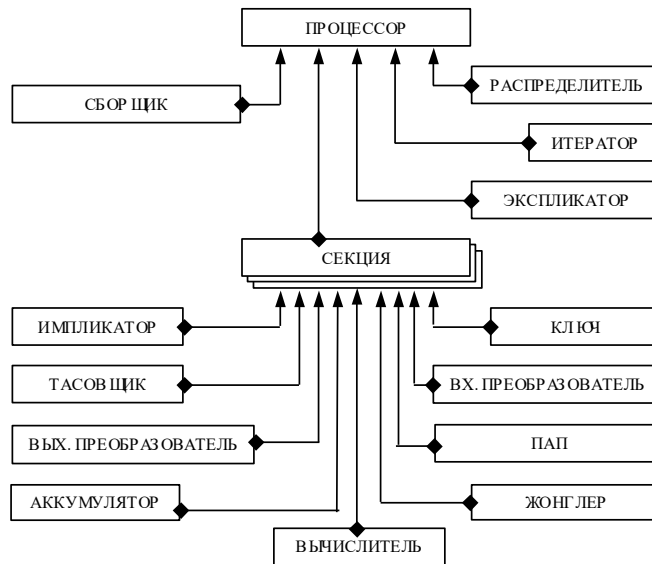


Рис. 1. Диаграмма классов вычислительного ядра.

Структура вычислительного ядра (рис. 1) программной модели разработана и описывается в терминах классов и состоит из набора устройств, каждое из которых способно выполнять отведенную ему функцию. В нее входят устройства: Распределитель, Итератор, Экспликатор и Секции. Каждая Секция в свою очередь включает в себя устройства: Ключ, Входной Преобразователь, Память Адресной Проверки Совпадения, Жонглер, Вычислитель, Аккумулятор, Выходной Преобразователь, Тасовщик и Импликатор. Описание классов устройств и их функций приведено в таблице 1. Подробно описание устройств, терминов и используемых обозначений приведено в работе [4].

Таблица 1. Описание классов устройств

Устройство	Вход	Выход	Функция
1. Распределитель	Исходная цифровая капсула	S-операнды	Принимает цифровую капсулу, декодирует ее и создает связанный список операндов. На каждом такте позволяет выбрать очередной операнд.
2. Итератор	S-операнды	C-операнды	Обрабатывает заменяемые операнды многократного применения.
3. Экспликатор	S, C или Em-операнды	I-операнды	Эксплицирует операнды согласно режиму экспликации. Реплицирует выбранный операнд согласно установленному режиму. Помещает реплицированные операнды в соответствующие секции по I-шине.
4. Секция	E и I-операнды	P, E, Em-операнды	Принимает на вход I-операнды от Экспликатора и E-операнды с шины межсекционной связи, обрабатывает их и распределяет результаты.
5. Ключ	E и I-операнды	M-операнды	Выбирает из E и I-операндов один согласно режиму селекции и помещает в память.
6. Память	M-операнды	F и M-операнды	Записывает поступивший на вход операнд в ячейку согласно установленному режиму. В

			случае столкновения операндов, помещает их на выходные F и M-шины.
7. Жонглер	F и M-операнды	L и R-операнды	В зависимости от режима формирует L и R-операнды и их функциональные поля.
8. Вычислитель	L и R-операнды	R-операнды	Выполняет арифметические, логические, битовые, DSP и другие операции над входными операндами. Производит преобразование теговых полей.
9. Аккумулятор	R-операнды	A-операнды	Хранит результат предыдущей операции.
10. Тасовщик	R-операнды	R, E, и Em-операнды	Помещает результирующий операнд на требуемую шину.
11. Импликатор	R-операнды с выхода Тасовщика	R-операнды	Пропускает на выход операнды с требуемым значением кода совпадения.
12. Сборщик	R-операнды	Выходная цифровая капсула	Принимает на вход результирующие операнды от Секций и помещает их в один из двух выходных наборов данных.

Работой устройств управляет Диспетчер (класс ПРОЦЕССОР). На каждом такте Диспетчер в строго определенном порядке запускает все устройства модели, обеспечивает обмен операндами между устройствами, разрешает конфликтные ситуации. Связь устройств в моделируемом ОУ и последовательность их работы определяется структурной моделью ОУ, отобразенной на рис. 2.

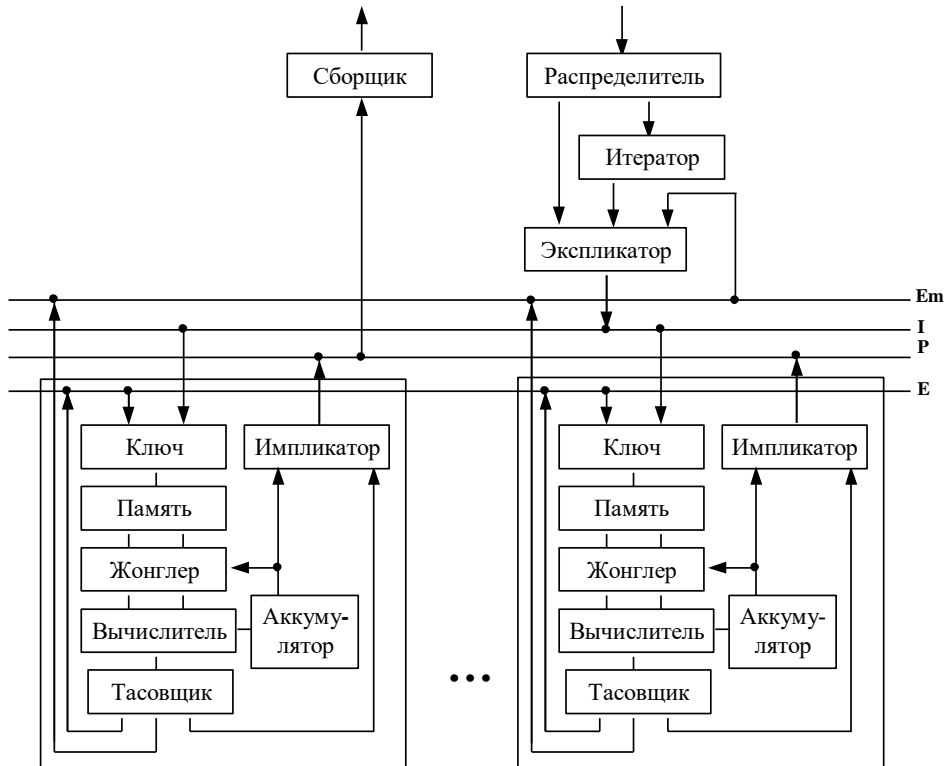


Рис. 2. Структурная модель ОУ.

Взаимодействие между устройствами происходит посредством передачи отдельных операндов или целых очередей операндов (рис. 3). Поэтому основной информационной

единицей здесь является Операнд. Каждый операнд состоит из Типа операнда (описывается в базовом классе), Значения (опционально) и набора (или нескольких наборов) Функциональных полей.

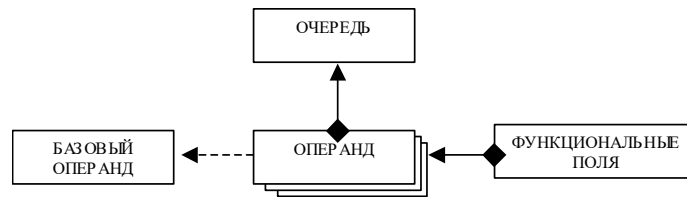


Рис. 3. Диаграмма классов Операнд и Очередь.

Средства визуализации для данной модели создаются отдельно от моделей устройств, что позволит запускать программы на заданное количество шагов без отображения происходящих процессов на экране монитора и существенно ускорит процесс вычисления и отладки капсул.

Достоинством данной программной модели является объектно-модульная структура, позволяющая быстро вносить изменения в работу любого устройства, а также добавлять, удалять или заменять устройства. Благодаря такой структуре появилась возможность реализации различных вариантов устройств, что является необходимым фактором при проектировании и исследовании разрабатываемого ОУ. В частности, возможно исследовать работу различного рода Функциональных Преобразователей, различных моделей Вычислителя и т.д. В качестве основного функционального преобразователя используются модели универсального преобразователя, преобразователя настраиваемого в процессе выполнения задачи и преобразователя с фиксированной настройкой.

Основным назначением ОУ является работа в системах, ориентированных на цифровую обработку сигналов. В связи с этим в программную модель заложены возможности моделирования задач, относящихся к классу ЦОС. Для этого в модели Вычислителя реализован набор команд, свойственных ЦОС-процессорам.

В класс каждого устройства интегрированы функции, позволяющие в любой момент времени отобразить на экране монитора состояние данного устройства. Наличие данных функций позволяет осуществить процесс протоколирования, что существенно облегчает процесс отладки программ. После отладки протоколирование может быть отключено, что в свою очередь, приведет к увеличению скорости выполнения программ.

Для тестирования работы модели ОУ разработан набор тестовых алгоритмов, покрывающий класс задач ЦОС, включающий:

- 1) GSM: AUTOCORRELATION, алгоритм автокорреляции в соответствии со стандартом GSM 6.10, стр. 3.1.4;
- 2) COMPLEX FFT BUTTERFLY, алгоритм "бабочка" из комплексного преобразования Фурье;
- 3) GSM: OFFSET COMPENSATION and PREEMPHASIS, алгоритм компенсации и прекомпенсации в соответствии со стандартом GSM 6.10, стр. 3.1.1 и 3.1.2;
- 4) FACTORIAL, алгоритм вычисления факториала;
- 5) ADAPTIVE FILTER, алгоритм адаптивной фильтрации;
- 6) MULTADD, различные алгоритмы умножения с накоплением.

Выводы

Выбрана и обоснована модель нижнего уровня - непосредственно модель ОУ.

Реализован прототип модели. Данная модель отражает физическую структуру ОУ, содержит модели составляющих устройств, модели реальных операндов и отражает их прохождение через устройства.

Разработанная модель ОУ соответствует классу полных моделей. При этом она может быть легко расширена дополнительными устройствами и обладает возможностью просто заменять функциональные свойства составляющих устройств.

Литература

1. Филин А.В. Динамический подход к выбору архитектуры вычислительных устройств обработки сигналов. // Системы и средства информатики. Вып. 11. М.: Наука, 2001 г.

2. Филин А.В. Особенности обработки сигналов на процессоре с рекуррентно-динамической парадигмой вычислений. // Системы и средства информатики. Вып.11. М.: Наука, 2001 г.

3. Морозов Н.В., Степченков Ю.А. Средства моделирования и анализа параллельных процессов в рекуррентном операционном устройстве. // Системы и средства информатики. Вып.12. М.: Наука, 2002 г.

4. Степченков Ю.А., Петрухин В.С., Филин А.В. Рекуррентное операционное устройство для процессоров обработки сигналов. // Системы и средства информатики. Вып.11. М.: Наука, 2001 г.