УДК 004.052.42

Принципы построения средств отладки рекуррентного вычислителя

А.Ю. Шнейдер[1], В.С. Петрухин[2], Ю.А. Степченков[3]

Федеральное государственное бюджетное учреждение науки Институт проблем информатики РАН,

alexshnd@rambler.ru[1], cokrat2@rambler.ru[2], ystepchenkov@ipiran.ru[3]

***Аннотация* — Обозначены проблемы построения отладочных средств гибридного двухуровневого рекуррентного вычислителя. Предложена логическая структура САПР, базирующаяся на языке сценариев. Определена логическая структура встроенных средств отладки рекуррентного операционного уровня вычислителя, позволяющая пользователю оперативно и гибко настраивать отражение процесса во множество наблюдаемых событий в зависимости от характера решаемой задачи.**

***Ключевые слова* — потоковая архитектура, отладка, рекуррентность**

I. Введение

В Институте проблем информатики Российской академии наук (ИПИ РАН) ведутся работы по созданию нетрадиционной рекуррентной архитектуры, предназначенной для реализации параллельных вычислений ограниченной размерности в области цифровой обработки сигналов. Для экспериментальной апробации предлагаемой архитектуры разрабатывается рекуррентный вычислитель в виде гибридного двухуровневого варианта с ведущим фон-неймановским процессором на управляющем (верхнем) уровне (УУ) и рядом потоковых процессоров на нижнем уровне – рекуррентном операционном устройстве (РОУ) [1].

На управляющий уровень возложены минимально необходимые функции управления операционным уровнем и связь с внешним окружением, и он реализуется на традиционных фон-неймановских (ФН) принципах. Многоядерный операционный уровень реализуется в виде РОУ, предназначенного для эффективного исполнения параллельных алгоритмов в речевой области.

В работе [2] отмечается, что в области разработки нетрадиционных параллельных архитектур наиболее трудоемким этапом был и остается этап отладки программ. Поэтому для разработки концепции построения средств аппаратного обеспечения РОУ важно выявить особенности данной архитектуры.

Архитектура РОУ радикально отличается по основным моментам не только от классической ФН-архитектуры, но и от других нетрадиционных параллельных архитектур, которые характеризуются наличием двух потоков: потока инструкций и потока данных. В РОУ оба потока сливаются в один общий поток *самодостаточных* данных, в котором, помимо собственно обрабатываемых данных, содержится управляющая и служебная информация (теги), необходимая для их обработки (в существующих архитектурах кодируемая в форме инструкций).

Архитектура РОУ относится к классу потоковых архитектур, причем в отличие от них тегируемые данные являются рекуррентно разворачивающимися. В памяти РОУ нет исполняемой программы в традиционном смысле. Теги содержат некоторую начальную сжатую информацию, обеспечивающую обработку связанных с ними данных (тегированных данных) для выполнения требуемой процедуры. Эти начальные значения тегов подвергаются рекуррентной саморазвертке устройством преобразования тегов, что обеспечивает выполнение как текущей процедуры, так и последующих. Каждый следующий шаг процедуры рекуррентно самоопределяется, в том числе с учетом результата предыдущего шага. Такое представление программы называется *капсулой*.

Анализ характерных особенностей архитектуры РОУ позволяет сформулировать требования, предъявляемые к средствам тестирования и отладки.

II. Особенности организации процесса отладки в РОУ

Процесс отладки программ можно представить в виде следующих действий:

а) обнаружение ошибки;

б) формулировка гипотезы о дефекте;

в) анализ контекста;

г) установление места возникновения ошибки;

д) проведение обратной трассировки;

е) установление причины;

ж) разработка корректирующих изменений;

з) исправление ошибки;

и) проверка.

Отсюда видно, что процесс отладки имеет значительную аналитическую интеллектуальную составляющую и поэтому является одной из самых сложных и трудоемких частей разработки новых сложных систем, плохо поддающихся прямой автоматизации.

При этом самыми сложными действиями являются:

анализ контекста, установление места возникновения ошибки, определение ее причины и разработка изменений для исправления. Поэтому все устремления направлены на повышение эффективности процесса формирования отладочной ситуации (контрольной точки и соответствующего «замороженного» состояния контекста выполнения) и на помощь в проведении обратной трассировки. Для ускорения разработки корректирующих изменений желательно также иметь инструменты для быстрой экспериментальной корректировки контекста и проверки гипотезы «на лету».

В традиционной архитектуре понятие контрольной точки привязано к адресу команды и данных. В архитектуре РОУ (потока данных) отсутствует понятие адреса команды. Более того, инициатором вычислительного процесса являются данные, которые несут информацию о выполняемых операциях. Таким образом, в процессе отладки параллельных программ в РОУ происходит размывание понятия контрольной точки и возрастание роли аналитической составляющей в процессе анализа отладочной ситуации. Поэтому предлагается процесс отладки РОУ строить на основе графа вычислительного процесса, в котором вершины будут выступать в качестве обозначений контрольных точек. Учитывая параллельность выполнения ветвей графа, необходимо ввести понятие общей контрольной точки для нескольких процессов и ее внутреннего содержания (событийности).

Как правило, процесс отладки носит иерархический характер. При этом на каждом уровне иерархии уточняется наполнение модели программы. Переход на более низкий уровень означает уточнение деталей модели, что значительно упрощает процесс написания и отладки параллельных программ. Вполне логично возникает вопрос о поэтапной детализации уровней абстракции модели и уточнения деталей процесса моделирования. Это в значительной степени зависит от стратегии организации отладки, в которой первостепенной является параллельная составляющая алгоритма. Поэтому верхняя модель программы РОУ должна содержать параллельную составляющую алгоритма. С переходом на нижние уровни абстракции уточняется (детализируется) вычислительная составляющая алгоритма.

На первом этапе решаются следующие задачи:

– оценка параллельности программы;

– оценка корректности загрузки и синхронизации данных;

– обнаружение нежелательных побочных эффектов (непредусмотренные изменения состояния памяти и регистров);

– выявление тупиковых ситуаций.

На последующих этапах осуществляется уточнение и детализация вычислительной составляющей программы – правильности вычислений, вызовов процедур, обмена данными и т.д.

Процесс отладки аппаратных и программных средств РОУ состоит из 3 этапов:

1. Отладка на поведенческом уровне.

2. Отладка на регистровом уровне.

3. Отладка непосредственно в ПЛИС.

На первом уровне отладки используется поведенческая модель, реализованная с использованием языка программирования C++.

Следующий уровень представляет собой систему капсульного программирования и отладки (СКАТ) [3].

Отладка в реальной аппаратной среде (ПЛИС) является наиболее трудоемким уровнем. Велика вероятность наложения программных и аппаратных ошибок. Тем более, что речь идет о разработке новой нетрадиционной архитектуры. Поэтому дальнейший материал будет посвящен именно этому этапу.

Для рассмотрения характера отладочных событий необходимо разработать стратегию отладки аппаратных и программных средств РОУ. Процесс отладки РОУ характеризуется следующими особенностями:

а) процесс отладки РОУ может осуществляться одновременно с процессом отладки управляющего уровня (при комплексной отладке рекуррентного вычислителя);

б) основной «движущей силой» выполнения алгоритма является поток данных;

в) процесс отладки имеет событийный характер;

г) понятие контрольной точки носит комплексный характер и глубоко связано с состоянием вычислительного контекста.

Можно выделить следующие этапы отладки РОУ:

– загрузка входных данных в шаблоны капсул и проверка их значений в буферной памяти (включая преобразование структур данных из «классического» формата в формат капсулы — при комплексной отладке);

– запуск процесса исполнения капсулы в РОУ и наблюдение за внутренними событиями, связанными с формированием выходных данных;

– отслеживание процесса загрузки выходных данных в шаблоны капсул, расположенных в буферной памяти (БП);

– отслеживание событий, связанных с синхронизацией процессов отладки РОУ и управляющего уровня;

– загрузка результатов вычислений управляющего уровня в шаблоны капсул, расположенных в БП;

– извлечение данных для использования управляющим уровнем — при комплексной отладке (включая преобразование из формата капсул в «классическую» форму).

Процесс отладки РОУ сопровождается появлением внутренних и внешних событий. Чтобы эти события состоялись, необходимо описать условия их возникновения (задать условия). Любое отладочное событие определяется логическим условием, построенным на совокупности сигналов и логических функций. Сигналы формируются непосредственно в аппаратуре, а логические функции могут быть заданы программно или реализованы с помощью аппаратных средств.

Любое отладочное событие должно содержать следующую информацию: место и время его возникновения; тип события и его порядковый номер; иден-

тификатор события.

По типу события можно определить связанный с ним контекст исполнения (состояние памяти и регистров), представляющий интерес с точки зрения диагностики, анализа и обратной трассировки (подмножество общего контекста исполнения РОУ).

Можно выделить четыре класса событий, наиболее полезных при анализе отладочных ситуаций:

а) *развитие процесса* (начало процедуры/действия, конец процедуры/действия, прохождение определенного этапа/реперной точки);

б) *доступ к объекту/ресурсу* (доступ для чтения/использования, доступ для записи/изменения);

в) *принятие решения* (выбор определенного варианта действия, выбор ресурса/объекта);

г) *изменение состояния* (резервирование объекта/ресурса, освобождение объекта/ресурса, создание объекта, уничтожение объекта, придание объекту определенного свойства/атрибута, изменение значения).

Для определения порядка во времени событий предлагается реализовать в системе аппаратный отладочный счетчик циклов.

При определении отладочных событий желательно разрешить ссылки на другие события и на данное событие. Это позволит обнаруживать неправильные последовательности событий и значительно упростит отладку.

Для этого среди функций, которые могут использоваться для формирования условий наступления отслеживаемых событий, должны присутствовать элементы темпоральной логики, позволяющие проверить, например, такие ситуации: событие А произошло после события В; событие А произошло перед событием В; после события А должно произойти событие В и т.п.

В свою очередь, отладочное событие должно воздействовать на отлаживаемую систему (РОУ). Действие события на РОУ может быть в виде: прерывания; останова; фиксации/протоколирования некоторого состояния системы; фиксации/протоколирования последовательности событий и т.д.

Таким образом, процесс отладки аппаратных и программных средств РОУ носит иерархический и весьма трудоемкий характер. Для автоматизации процесса отладки РОУ предлагается разработать САПР.

В первую очередь необходимо описать в виде сценария совокупность отладочных событий и вызываемых действий, разработав (или используя существующий) для этого язык описания сценариев. Это даст возможность не только зафиксировать (запомнить) порядок проведения отладки, но и представить этот процесс в удобной и обозримой форме.

На рис. 1 приведена укрупненная структура элементов САПР отладки РОУ. Пользователь САПР описывает процесс отладки на языке описания сценариев. Далее система компилирует исходный текст в совокупность сценариев отладки. Для наполнения сценариев используется библиотека событий и действий. В свою очередь, события представлены библиотекой сигналов и функций. Сценарии должны строиться из Правил, каждое из которых фиксирует Событие и состоит из условия (Логического описания события) и Действия. События должны запоминаться в САПР.

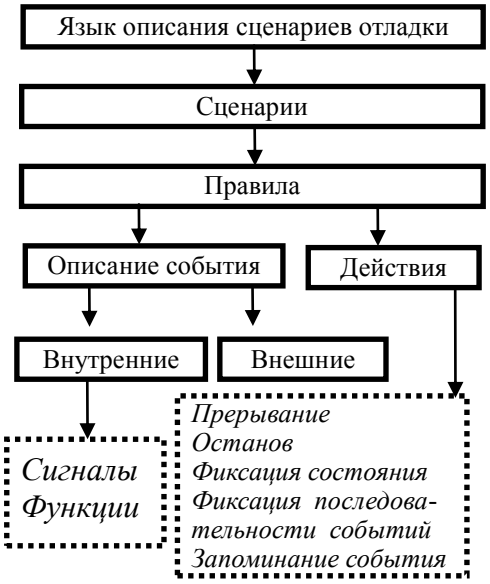

Рис. 1. Элементы описания сценария отладки РОУ.

III. Встроенные средства отладки РОУ

Эффективную отладку аппаратных и программных средств сигнальных процессоров невозможно осуществить без средств, встроенных непосредственно в кристалл. Поэтому далее мы будем говорить о внутрисхемной отладке с использованием инструментальных средств, непосредственно встроенных в РОУ.

Существует набор базовых действий, которые необходимо выполнить для осуществления контроля за выполнением отлаживаемой задачи. Эти действия можно классифицировать следующим образом:

– формирование отладочных событий;

– запуск и прерывание (и/или останов) выполнения задачи;

– получение информации о состоянии РОУ;

– продолжение/изменение выполнения задачи.

Пользователь формирует (программирует) отладочные события на базе библиотеки внутренних сигналов РОУ и набора функций (логических, арифметических и функций сравнения).

Все события можно разделить на логические (связанные с архитектурой вычислительного процесса) и физические (связанные с архитектурой и функционированием аппаратуры РОУ).

В свою очередь, любое отладочное событие на аппаратном уровне можно рассматривать как совокупность сигналов, объединенных логическим выражением (условием). Такое условие может быть реализовано

аппаратно (встроенная реализация) или программно.

Условно перечень сигналов, необходимых для формирования отладочных событий в РОУ, можно разделить на 2 группы.

Группа 1: запись данных в БП со стороны УУ; запись данных со стороны РОУ; чтение данных из БП; размножение операнда в распределителе.

Группа 2: запись операнда в память совпадений без образования пары; запись операнда в память совпадений с образованием пары; перенаправление операндов; активизация рекуррентного преобразователя тегов; формирование перехода; исполнение кода операции; запись результата операции в аккумулятор; запись результата операции в регистр X; пересылка операнда в соседнюю секцию.

Сигналы в группе 2 должны быть привязаны к номеру исполняемой секции.

Для формирования (программирования) события необходим набор объединяющих их логических и арифметических функций.

Процесс загрузки конфигурации ПЛИС (загрузка конфигурационного файла) сопровождается также загрузкой конфигурации аппаратных событий в РОУ. Далее пользователь должен определить на текущий момент совокупность отладочных событий и порядок их появления (в случае необходимости). Таким образом, любое событие должно быть активировано в системе для использования в процессе отладки. Активированное событие становится наблюдаемым и может использоваться при отладке.

Далее выполняется действие, связанное с запуском задачи (капсулы). Для этого пользователь загружает соответствующее значение указателя порта 2 БП и формирует сигнал пуска РОУ. Предполагается, что эти действия будут выполняться не напрямую, а через вызовы функций программного интерфейса (API) РОУ.

В процессе исполнения капсулы возникает отладочное событие, которое может вызвать останов РОУ или фиксацию/протоколирование некоторого состояния системы. Останов РОУ сопровождается генерацией сигнала отладочного прерывания, который поступает на УУ, так как РОУ не имеет возможности обрабатывать (анализировать) возникающие события. В свою очередь, УУ вызывает программу обработки данного прерывания (передает управление Отладчику).

Если фиксация/протоколирование состояния системы реализованы на аппаратном уровне, то сигналы останова РОУ и отладочного прерывания не формируются.

Последующие действия связаны с процессом анализа пользователем состояния РОУ. Для этого отладчик считывает состояние регистров и устройств памяти РОУ и отображает их состояние на устройстве визуализации и/или записывает в файл трассировки.

На основании анализа полученной информации пользователь имеет возможность изменять состояние регистров и ячеек памяти. Введенные изменения должны быть занесены в соответствующие регистры и ячейки памяти РОУ и БП.

Обеспечение целостности и непротиворечивости внутреннего состояния РОУ полностью ложится на пользователя. Однако в более интеллектуальном отладчике возможно реализовать проверку этих условий, предотвращая формирование пользователем некорректного состояния РОУ. При этом у пользователя остается возможность принудительного введения некорректного состояния памяти и регистров для проверки некоторой отладочной гипотезы. На рис. 2 приведена логическая схема организации работы встроенных средств отладки РОУ.

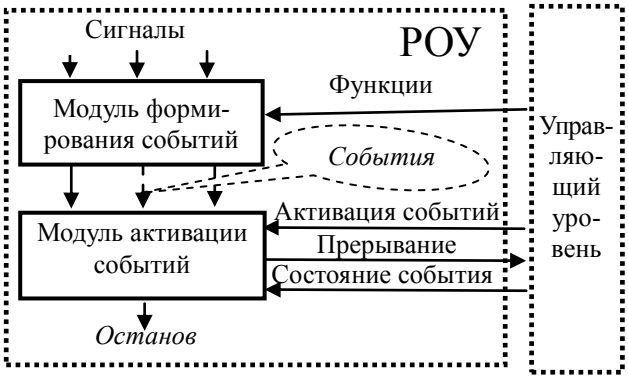

Рис. 2. Схема работы встроенных средств отладки РОУ.

IV. Выводы

1) Показано, что процесс отладки аппаратных и программных средств РОУ носит иерархический и весьма трудоемкий характер, для автоматизации которого предлагается разработать специализированную САПР отладки РОУ с использованием языка сценариев.

2) Показано, что процесс отладки аппаратных и программных средств РОУ носит событийный характер. В условиях отсутствия явного описания процедуры (последовательности инструкций) процесс обработки потока данных порождает множество событий, анализ которых позволяет оценить правильность функционирования РОУ.

3) Предложенные встроенные средства отладки РОУ позволяют пользователю оперативно и гибко настраивать отражение процесса во множество наблюдаемых событий в зависимости от характера решаемой задачи.

Литература

[1] Степченков Ю.А., Петрухин В.С. Особенности гибридного варианта реализации на ПЛИС рекуррентного обработчика сигналов // Системы и средства информатики. Доп. вып. М.: ИПИ РАН, 2008. С. 118–129.

[2] Липаев В.В. Качество программного обеспечения. М.: Финансы и статистика, 1983. 264 с.

[3] Зеленов Р.А., Степченков Ю.А., Волчек В.Н., Хилько Д.В., Шнейдер А.Ю., Прокофьев А.А. Система капсульного программирования и отладки // Системы и средства информатики. М.: ТОРУС ПРЕСС, 2010. Вып. 20. № 1. С. 24-30.