УДК 004.052.42

Использование технологии Remote Procedure Call для тестирования и отладки процессоров "Эльбрус".

А.Е. Ометов[2], И.Е. Билялетдинов[1], Л.С.Тимин[1,2], А.А. Виноградов[2]

[1]Институт электронных управляющих машин им. И.С. Брука

[2]АО «МЦСТ»

Во время отладки процессоров необходим доступ к их диагностическим средствам, который можно осуществить через отладочный интерфейс. В процессорах «Эльбрус» в качестве отладочного интерфейса используется JTAG. Он позволяет получить доступ к системе управления частотой процессора, температурным датчикам, физическим уровням памяти и межпроцессорных связей, отладочным механизмам ядер и других частей процессора. В момент прихода новых процессоров нет гарантии, что всё работает так, как задумывалось, и из-за этого можно потерять прямой программный доступ к каким-нибудь частям процессора. В таких случаях может помочь использование заранее заложенных в процессор отладочных средств, например, можно менять начальные настройки процессора, в обход штатных средств, с помощью JTAG [1]. Даже если все работает, отладочный доступ позволяет проводить исследование границ работоспособности процессора, получать информацию и тестировать процессор с минимальным количеством оборудования (часто необходимо лишь подать питание) в экстремальных условиях. Однако, работоспособность диагностических средств и программного обеспечения, для управления ими, необходимо проверять на этапе разработки процессора.

Диагностическое программное обеспечение, работающее через JTAG-интерфейс, взаимодействует с процессором путем подачи тестовых воздействий в виде битовых векторов и наблюдения отклика устройства [2]. Однако, отлаживать данный процесс на уже готовом изделии проблематично, т.к. возможно наблюдать только конечный результат воздействий, что не всегда дает точную информацию об ошибке в векторе. В рамках данной работы, для осуществления верификации диагностических средств процессора и тестов была разработана модель JTAG-контроллера, позволяющая взаимодействовать с Verilog моделью целевого устройства через JTAG-интерфейс.

Для любой модели JTAG-контроллера предъявляются следующие требования:

1. Слежение за состоянием TAP [3] контроллера. Модель должна самостоятельно находить правильные пути перехода по таблице состояний для выполнения поставленной задачи.

2. Обмен данными по интерфейсу JTAG. Модель должна быть ответственна за преобразование битовых векторов в сигналы и формирование отклика.

Для получения максимальной выгоды с функциональной точки зрения, контроллер разделен на две части: одна на языке C, другая на System Verilog. Используя такой подход, можно пользоваться обширным количеством различных библиотек языка C и C++, при этом не теряя возможностей языка Verilog.

Часть на языке System Verilog ответственна за формирование синхросигнала Test Clock (TCK) [3], за подачу значений сигналов ко входам и за снятие значений с выходов JTAG-интерфейса. Эта часть контроллера представляет из себя не синтезируемый модуль, который подключается к соответствующему интерфейсу модели процессора.

Часть на языке C ответственна за формирование значений управляющих сигналов для JTAG-интерфейса, за слежение за состояниями контроллера, за формирование вектора отклика и за обеспечение программного интерфейса JTAG-контроллера.

Между собой части контроллера общаются с помощью System Verilog DPI [3]. Это интерфейс, который используется для связи System Verolog'a и другого языка программирования.

Моделирование работы процессора, или даже его небольшой части, является сложной и объемной вычислительной задачей, поэтому, в большинстве случаев, оно происходит на мощных серверных машинах, доступ к которым имеется по сети. Вследствие этого, было решено реализовать возможность удаленного тестирования с использованием метода удаленного вызова процедур (RPC). Обычно, реализация RPC технологии включает в себя два компонента: сетевой протокол для обмена в режиме клиент-сервер и язык сериализации объектов.

В качестве сетевого протокола была выбрана библиотека ZeroMQ, а языком сериализации был выбран YAML.

В итоге, после реализации RPC, модель контроллера разделилась на серверную и клиентскую части с взаимодействием типа Request-Reply. Клиентская часть представляет из себя библиотеку функций, которые позволяют производить манипуляции с JTAG-цепочками внутри процессора с помощью битовых векторов, абстрагируясь от Verilog моделирования. Библиотека ответственна за формирование и посылку запроса серверу, а также за получение ответа.

В серверную часть входит модуль на языке System Verilog, который производит симуляцию тестовых векторов, и часть на C и C++, ответственная за обработку запросов клиента и формирование выходных данных.

В результате, реализованная архитектура позволяет:

1. Отлаживать тестовые вектора и диагностическое оборудование на Verilog модели целевого устройства, абстрагируясь от самого процесса моделирования.

2. Детально проследить за воздействием, которое оказывает тестовый вектор, используя возможности САПРа, в котором происходит моделирование.

3. Производить удаленную отладку на сервере по сети.

4. Не прерывать процесс моделирования для изменения тестовых воздействий, т. к. это можно сделать «на лету» в отладочной программе.

Когда средства диагностики процессора отлажены на модели появилась необходимость проверять сам процессор. Обычно машины с процессорами находятся где-то в серверной, где человеку находиться не удобно, поэтому мы применили тот же подход, что и на модели – разделили программу на серверную и клиентскую части. Серверная часть управляет JTAG-контроллером и реализует часто используемые высокоуровневые функции. Клиентская часть – библиотека, предоставляющая простой доступ к этим функциям. Клиентская часть была написана на нескольких языках программирования, включая скриптовые TCL и Python, позволяющие изменять программу тестирования без перекомпиляции и без использования дополнительных программ. Для того, чтобы не перекомпилировать серверную часть для каждого процессора серверная часть анализирует состав JTAG цепочки и автоматически определяет тип процессора, к которому подключен JTAG-контроллер. Для передачи данных между серверной и клиентской частью используются те же библиотеки YAML и ZMQ.

Для управления процессором были реализованы функции записи и чтения регистров процессора через отладочную инфраструктуру: часть буфера инструкций процессоров Эльбрус доступна для записи через JTAG-интерфейс. Чтение значений доступно через специализированный 64-битный регистр, видимый как внутри процессора, так и в JTAG цепочке. На основе этого механизма можно запускать тесты любой сложности, например, тренировать оперативную память и проверять ее работоспособность. Также, для запуска более сложных и комплексных тестов на рабочей частоте процессора, авторами была разработана следующая методология: тест с помощью отладочных средств записывается напрямую в кэш процессора, а затем ядро получает команду исполнения этого теста уже на рабочей частоте. Однако у этого метода есть очевидное ограничение на размер теста – не более размера кэша процессора.

Авторы считают, что в данной работе новым является объединение всех отладочных инструментов в одной программе-сервере, с возможностью абстрагироваться от того, как на самом деле происходит доступ к инфраструктуре процессоров. Были созданы удобные в использовании библиотеки для разных языков программирования. Вместе с моделью JTAG-контроллера появилась возможность запускать одну и ту же программу тестирования для реальных процессоров и их моделей без модифицирования кода. Дополнительным преимуществом является кроссплатформенность всех программ и библиотек, что позволяет запускать их на различных машинах.

Литература

1. *Laung-Tern Wang, Charles E. Stroud, Nur A. Touba*. System-on-Chip Test Architectures: nanometer design for testability. – Burlington: Morgan Kaufmann Publishers, 2008. – 856 p.

2. *Laung-Terng Wang, Cheng-Wen Wu, Xiaoqing Wen*. VLSI Test Principles and Architectures: Design for Testability. – San Francisco: Morgan Kaufmann Publishers, 2006. – 777 p.

3. IEEE Std 1149.1-2001: IEEE Standard Test Access Port and Boundary-Scan Architecture. – New York: Institute of Electrical and Electronics Engineers, 2001. – 208 p.