

# Инструменты для периферийного сканирования: тестирование плат и отладка функциональных узлов

Алексей ИВАНОВ  
alexey@jtag.com

**Сравнительно недавно компания JTAG Technologies анонсировала новое направление отладки аппаратных средств под названием JTAG Live. Для этого в дополнение к основному сайту [www.jtag-technologies.ru](http://www.jtag-technologies.ru) был запущен отдельный сайт [www.jtaglive.ru](http://www.jtaglive.ru). В чем кроется смысл такого разделения? Очевидно, что если это просто были бы примитивные средства тестирования с низким уровнем автоматизации, то разделение на два направления было бы не совсем понятно. Причина кроется в абсолютно разных задачах, которые призваны решать те и другие средства, а следовательно, и в разной целевой аудитории.**

Существуют профессиональные автоматические системы тестирования и программирования плат при помощи JTAG, позволяющие при наличии файлов схемотехники протестировать большинство цифровых цепей на плате на наличие дефектов и локализовать эти дефекты. То есть, имея только что собранную плату или модуль, отказывающийся работать, можно нажатием кнопки запустить серию заранее созданных тестовых последовательностей и получить сообщение о физических причинах отказа, таким образом, действительно проведя «сканирование». Общаясь с инженерами и объясняя принцип работы профессиональных средств тестирования, автор часто сталкивался с вопросом: «А можно ли с помощью этих средств задать такие-то сигналы на таких-то ножках и посмотреть, что будет на таких-то выводах?» Скорее всего, такой вопрос задает разработчик аппаратных средств, занимающийся отладкой схемы. Ответ — да, можно, но это всего лишь 1% от всех возможностей системы. К чему тогда вся эта математика, анализирующая схему, генерирующая тестовые векторы и обрабатывающая результаты тестирования для локализации всех возможных дефектов? А между тем технология периферийного (граничного) сканирования по стандарту IEEE 1149.1 дает отличный шанс «заглянуть», например, под корпус BGA, что инженеру-разработчику, занимающемуся отладкой, тоже бы не помешало. Как раз для этого и появились программные средства JTAG Live — простые и не обремененные интеллектом.

Но вот обратная ситуация: автор получает письмо от работника отдела тестирования

при производстве, который пишет, что он загрузил бесплатную программу Buzz с сайта JTAG Live и пытается адаптировать ее под серийное тестирование выпускаемых изделий. Но Buzz — это в своей сущности мультиметр, только виртуальный, использующий регистры периферийного сканирования, микросхем с поддержкой IEEE 1149.1. Если сравнивать профессиональные системы JTAG-тестирования плат со средствами, позволяющими «дергать» определенные сигналы на плате, то аналогией может служить сравнение мультиметра (или хотя бы многоканального осциллографа) и установки с летающими щупами: они выполняют абсолютно разные функции, хотя используют один и тот же принцип. Можно ли себе представить методику тестирования узлов собранных серийных изделий, базирующуюся на использовании мультиметра? Думаю, что с трудом.

В связи с вышесказанным автор и решил написать эту статью, которая на примере профессиональной системы тестирования JTAG ProVision и программ JTAG Live подробно объясняет принцип их работы и возможности, для того чтобы у читателя сложилось ясное представление о назначении тех и других инструментов. В теории очень сложно все это объяснить, поэтому все показано на практических примерах. Здесь следует упомянуть, что ни те, ни другие средства все же не требуют знания стандарта IEEE 1149.1 или JTAG-интерфейса. Простые или сложные — программы создания тестов используют графический интерфейс, все остальное делают компьютер и контроллеры периферийного сканирования.

## Профессиональные системы тестирования и программирования на основе JTAG ProVision

За основу описания профессиональных средств возьмем систему проектирования тестов и приложений для программирования JTAG ProVision. Эта же система позволяет выполнять и созданные приложения, а также диагностировать обнаруженные дефекты. Аппаратное обеспечение автор решил опустить, чтобы не перегружать статью.

Предположим, что у нас есть плата для тестирования, содержащая несколько компонентов с поддержкой стандарта IEEE 1149.1 (в дальнейшем будем называть их компонентами с поддержкой JTAG, хотя это и не всегда верно [3]). Есть также другие компоненты, которые не имеют JTAG-логики, но их все же можно протестировать, используя ячейки периферийного сканирования основных микросхем платы. Чтобы начать генерацию тестов для проверки цепей этой платы, необходимо создать проект, который будет содержать всю необходимую информацию о межкомпонентных связях и самих компонентах. Таким образом, первоначальный этап — это сбор всей информации и добавление ее в проект. Список исходных данных для создания проекта представлен в таблице. Причем для генерации минимального набора тестовых векторов обязательными являются нетлист и BSDL-файлы на компоненты с поддержкой JTAG. При добавлении в проект других моделей (для памяти, функциональной логики и т. д.) можно будет создать дополнительные наборы тестов и приложений. BSDL-файлы в большинстве случаев

Таблица. Исходные данные для создания проекта

Тип исходных данных	Вид	Обязательны для создания минимального набора тестов	Примечания
Схематика	Нетлист (список соединений)	Да	Генерируется из САПР, где разрабатывалась схематика
Модели компонентов, поддерживающих JTAG (IEEE 1149.1)	BSDL-файлы	Да	Описывают архитектуру JTAG-регистров ИМС, используемых для тестирования
Модели компонентов без JTAG	Библиотечные файлы моделей для памяти, логики и т. д.	Нет	Описывают функциональность других компонентов для генерации тестов для них

находятся в открытом доступе и загружаются с сайтов производителей ИМС. Модели остальных функциональных устройств берутся из библиотеки либо создаются пользователем JTAG ProVision.

В итоге, когда исходные данные загружены в проект, мы можем наблюдать и при необходимости редактировать карту компонентов платы с соответствующими им моделями, как показано на рис. 1.

Убедившись, что все модели определены правильно (или хотя бы определены BSDL-модели компонентов с поддержкой JTAG), можно приступить к генерации различных приложений. Меню выбора типа создаваемого приложения показано на рис. 2. Количество различных приложений в рамках проекта ничем не ограничено.

Обычно самым первым тестом является тест межкомпонентных связей, обозначенный на рис. 2 как Interconnect Test. Однако его название, которое автор привел на русском языке, несколько обманчиво. Правильнее было бы назвать его «Тест окружающих цепей всех компонентов, поддерживающих JTAG». Если взять в качестве эксперимента плату всего лишь с одним таким компонентом, то при генерации и выполнении теста можно будет увидеть приличный набор тестовых векторов с различными комбинациями тестовых данных для большого количества цепей. Очевидно, что в такой тест могут входить не только собственно проверка связей между компонентами, поддерживающими периферийное сканирование

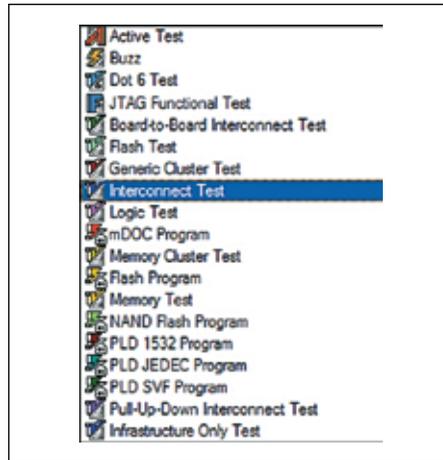


Рис. 2. Виды тестовых приложений, создаваемых в JTAG ProVision

JTAG, но и тестирование цепей, окружающих каждый из таких компонентов, на короткие замыкания, «залипания» на питание или «землю». Причем все эти дополнительные тесты могут регулироваться при генерации, изменяя, таким образом, количество тестовых векторов, которое может достигать при расширенном наборе опций 80–100 штук, а при минимальных настройках ограничиваться двумя-четырьмя. Каждый из таких векторов затем будет по очереди вдвигаться по JTAG-интерфейсу в тестируемую плату из контроллера. Следует также отметить, что чем меньше моделей для окружающих компонентов без поддержки JTAG

определено, тем меньше цепей будет тестироваться, так как модель также сообщает об активности или пассивности компонента и о том, как держать этот компонент в пассивном состоянии во время теста.

После генерации теста вся информация о тестируемых цепях, участвующих компонентах и ограничениях является открытой. Можно открыть список цепей и для каждой посмотреть атрибуты, возможные конфликты или установить принудительно для любой цепи фиксированный логический уровень во время всего теста. Еще более подробную информацию о тестировании (или о том, что происходит на плате во время тестирования) можно получить, запустив созданный тест на реальной плате. При выборе соответствующего режима отображения результатов после тестирования можно увидеть отчетную «таблицу истинности» (Truth Table Reporter). Только это не таблица истинности входов и выходов логического элемента, а полученных с тестируемой платы векторов по отношению к ожиданиям системы проектирования. Фрагмент такой таблицы показан на рис. 3 (на самом деле она довольно большая). По этой таблице видны не только полученные результаты тестирования, но и все воздействия на цепи в каждом из векторов вплоть до уровней «вывод компонента» и «ячейка периферийного сканирования». Следует отметить, что на производстве при отлаженном тестировании, конечно, такую таблицу не выводят на экран компьютера оператора: ему(ей) достаточно сообщения «прошел/не прошел» и диагностических сообщений в случае найденных дефектов (рис. 4).

Но вернемся к таблице с результатами тестирования на рис. 3. Судя по данным таблицы, тестовые векторы (или паттерны) построены довольно «хитрым» образом. Система сама их сгенерировала на основе анализа списка соединений схемы платы. Причем она выстроила комбинации логических уровней, выставляемых с ячеек периферийного ска-

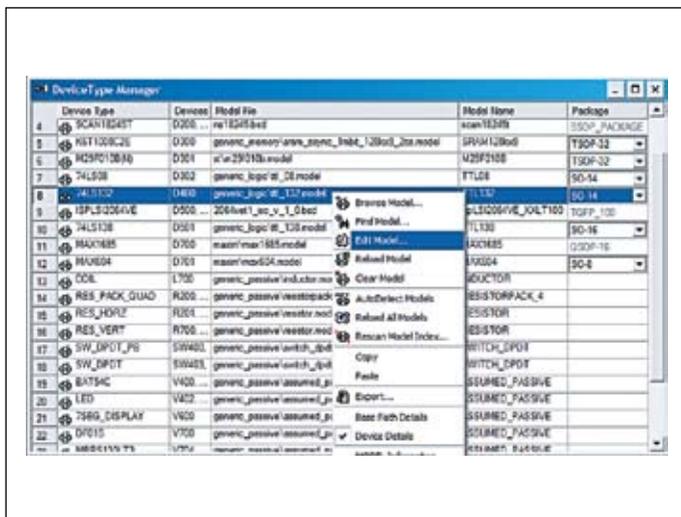


Рис. 1. Менеджер моделей компонентов в JTAG ProVision

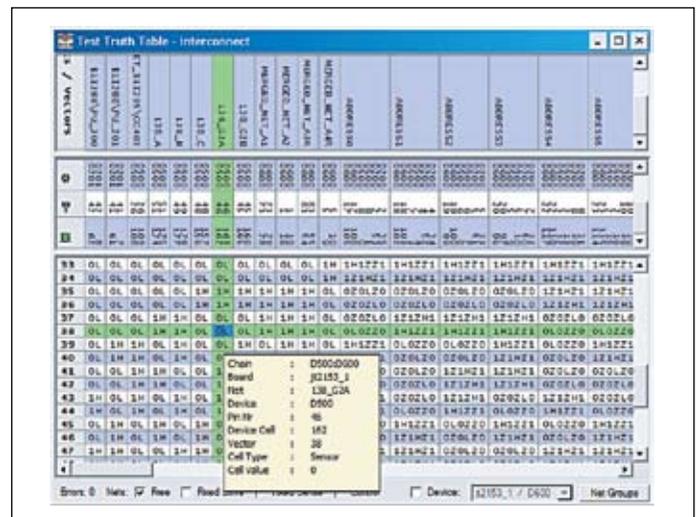


Рис. 3. Отчетная таблица по результатам тестирования

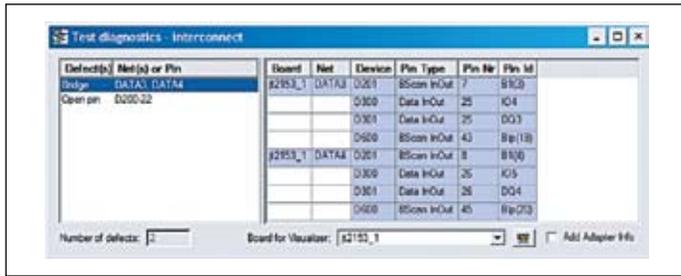


Рис. 4. Диагностика дефектов

нирования компонентов с поддержкой JTAG-тестирования, таким образом, чтобы эти комбинации локализовали все возможные дефекты, которые могут появляться в этих цепях. Здесь же можно наблюдать и разные типы тестируемых цепей: например, цепи, имеющие контакт с несколькими компонентами с поддержкой JTAG или только с одним. Если выбрать опцию расширенного теста во время генерации приложения, то при тестировании будут использоваться все до единого драйверы и сенсоры, имеющиеся на каждой из цепей, чтобы проверить работу каждого из них.

Так, для цепи ADDRESS1, которая идет на ячейки периферийного сканирования ИМС D200, D500 и D600, в тесте участвуют все три микросхемы, каждая из которых по очереди «общается» с двумя другими. Делается это для более точной диагностики дефектов. А вот цепь 138\_G2A имеет только один компонент с JTAG, и для полноценного ее тестирования требуется создать еще тест логического компонента, на который она идет. Однако на рис. 3 можно заметить, что в тесте межкомпонентных связей она все равно присутствует: на ней устанавливаются различные значения, призванные детектировать замыкания с другими цепями, питанием и «землей». Это становится возможным благодаря тому, что вывод единственного JTAG-компонента этой цепи D500 имеет как драйвер, так и сенсор. Цепь 138\_G2A появится в тесте только при условии, что логический компонент, который имеет отношение к этой цепи, объявлен пассивным или имеет модель, сообщающую, что данный вывод безопасен.

Вот поэтому один из новых пользователей JTAG ProVision искренне удивился, спросив автора о том, почему столько много цепей в результате выполнения приложения Interconnect Test. Он полагал, что тест межсоединений — это только прямые связи между компонентами, поддерживающими периферийное сканирование JTAG.

Здесь следует отметить и то, что, скажем, на той же цепи ADDRESS1 присутствуют не только компоненты с поддержкой JTAG, но и другие, например микросхемы флэш-ПЗУ и оперативной памяти, которые без отсутствия контроля могут влиять на тест межсоединений. Чтобы цепь ADDRESS1 была включена в тест межкомпонентных связей, требуется исключить их влияние, что делается автоматически

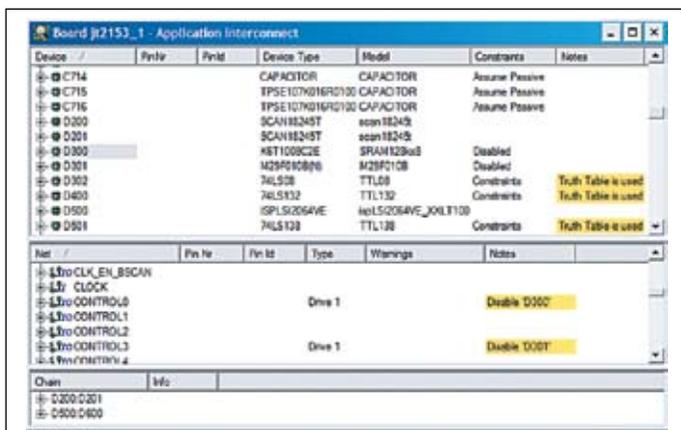


Рис. 5. Информация о цепях и компонентах во время генерации теста

при генерации теста. Дело в том, что модель компонента, не поддерживающего JTAG-тест (например, памяти), содержит не только ее функциональное описание для создания тестов, но и информацию о сигналах ENABLE/DISABLE для перевода выводов этого компонента в третье состояние. При генерации теста система распознает эти атрибуты в моделях и автоматически «держит» шины памяти в состоянии высокого импеданса. Это можно видеть на рис. 5, где микросхемы D300 и D301 отключены путем удержания во время теста контрольных сигналов в заданном состоянии. Поэтому тест цепи ADDRESS1 можно считать надежным и достоверным. Таким образом, цепи действительно тестируются: для каждой из них автоматически производятся все действия, которые можно выполнить при помощи периферийного сканирования.

Про один только тест межсоединений (или, как мы договорились, — тест окружающих цепей JTAG-компонентов) можно написать целую статью, так как много опций в описании выше было опущено. Но это только тест межсоединений. Далее следует тест кластеров памяти, функциональной логики, флэш-памяти и тесты, созданные вручную для особых узлов схемы. А еще дальше — JTAG Functional Test (JFT) для тестирования тех участков, где требуется алгоритмическое описание проверки функциональности, для чего в JFT используется встроенный язык программирования. Каждое из этих приложений дополняет тест межсоединений, позволяя детектировать больше дефектов на плате. В итоге все эти приложения могут запускаться в последовательности нажатием одной кнопки.

Для примера приведем еще тест памяти. Микросхемы памяти не должны поддерживать периферийное сканирование. Как уже говорилось ранее, для тестирования участков схемы, где отсутствуют компоненты с поддержкой JTAG, используются функциональные модели компонентов. Для логики это может быть таблица истинности микросхемы, для памяти — описание циклов чтения и записи. По модели генерируется набор определенных воздействий, которые опять в виде векторов будут выставляться с ячеек периферийного сканирования окружающих JTAG-микросхем на цепи, связанные с тестируемым кластером. Для ОЗУ чаще всего хватает библиотечных моделей, имеющихся в библиотеке ProVision (рис. 6). Здесь представлен только фрагмент библиотеки. В случае с памятью количество таких векторов больше, чем у теста межсоединений, и может варьироваться от нескольких сотен для SRAM до нескольких тысяч для DDR. После тестирования мы можем просмотреть таблицу с результатами, подобную той, что показана на рис. 3. Однако там будут присутствовать только цепи, связанные с тестируемой микросхемой (микросхемами) памяти. Следует отметить, что при наличии полученных ошибок в тесте диагностика в текстовом формате с указанием типа дефекта тоже работает, так как в память записываются

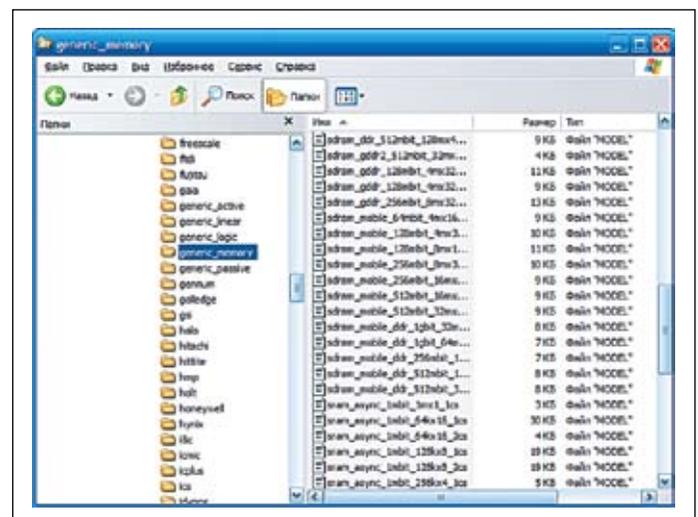


Рис. 6. Отрывок из библиотеки моделей в JTAG ProVision

«специальные» коды, при этом перебираются адресные цепи в требуемом для лучшей диагностики порядке.

Это краткое описание сделано для того, чтобы на примере нескольких тестов из профессиональных систем проектирования и тестирования перейти к описанию низкоуровневых средств из семейства JTAG Live, где автоматическое проектирование отсутствует. Автор сознательно опустил описание программирования ПЗУ и ПЛИС при помощи ProVision, решив, что тестирования будет достаточно.

### Ручные средства отладки цифровых цепей

Как уже было упомянуто, программы семейства JTAG Live не относятся к автоматическим, более того, их сложно даже отнести к средствам тестирования. Соответственно, здесь нет анализа схематики, генерации тестовых векторов, а без всего этого невозможно и автоматическая диагностика дефектов. Но эти средства бывают полезны во многих случаях. Например, когда нам требуется проверить состояние заранее выбранных цепей или выводов, технология периферийного сканирования позволяет сделать это без использования внешних измерительных приборов. Вот отрывок из одного отзыва, приведенного на сайте JTAG Live: «...Отладка аппаратного ПО производится при помощи профессиональных средств эмуляции, поэтому верификация самой платы и ее межсоединений чаще всего делалась при помощи тех же средств. Старые средства, такие как осциллографы, логические анализаторы и мультиметры, не годились из-за отсутствия физического доступа для щупов или зажимов. Но, к сожалению, использование эмуляции тоже имеет свои недостатки, так как инженеры по аппаратным средствам не всегда знакомы со средствами эмуляции и детальной структурой микропроцессоров. Более того, использование эмулятора требует, чтобы аппаратное ПО было отлажено до верификации самой платы, вследствие чего получается замкнутый круг и потеря времени...»

Для управления отдельными выводами или группами выводов конкретно взятых микросхем, хоть и с использованием JTAG, не требуются все возможности профессиональных средств проектирования тестов и сложная аналитическая математика. Как следствие этого — программ JTAG Live не нужны списки соединений платы из САПР и модели всех компонентов. Главное — это BSDL-модели, описывающие архитектуру периферийного сканирования соответствующих микросхем. Кроме того, конечно же, требуется определить каналы сканирования, причем сделать это вручную, так как информация о цепях у нас отсутствует. Таким образом, основная идея этого рода средств — ра-

бота с отдельными выводами или группами выводов, при этом «рабочими» пинами будут только выводы микросхем с поддержкой периферийного сканирования: моделей остальных устройств на плате у системы также нет.

Давайте перейдем от теории к практике на примере бесплатной программы Buzz, которую можно загрузить с сайта. Затем коснемся и других модулей семейства JTAG Live. На рис. 7 показано стартовое окно любой из программ JTAG Live. Если в JTAG ProVision работа начинается с конвертации нелистов, откуда берутся названия компонентов и цепей и связи между узлами, то в JTAG Live такую информацию брать неоткуда, и пользователь сам вводит нужное количество каналов и порядок JTAG-микросхем в этих каналах путем последовательного открытия BSDL-файлов. Следует отметить, что в JTAG Live ограничено максимальное количество JTAG-каналов, которыми можно одновременно управлять: максимум их может быть два. Но использовать сразу два канала можно только с двухпортовым контроллером JT3705/USB. Если вы решили воспользоваться отладочными кабелями Altera или Xilinx (которые являются однопортовыми), то для того, чтобы «прозванивать» межсоединения на плате с несколькими JTAG-каналами, единственный выход — это объединение физических разделенных каналов в один последовательный при помощи кабельной оснастки вне платы. В JTAG ProVision такой проблемы не существует: нет ограничений по количеству каналов сканирования при создании проекта и тестов, контроллеры, работающие с JTAG ProVision, имеют большее количество портов и являются более мощными.

Теперь взглянем на сам проект в JTAG Live Buzz, который показан на рис. 8. Собственно, в этом окне делается все: задаются пины, которые нужно проверить, устанавливаются дополнительные сигналы и отображаются

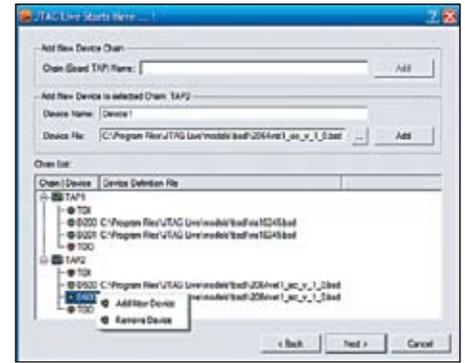


Рис. 7. Стартовое окно JTAG Live

результаты измерений. Как уже говорилось, здесь нет названий цепей и информации о межсоединениях, как в ProVision: предполагается, что пользователь знает, между какими выводами должна быть взаимосвязь (прямая или косвенная) и какие отклики он ожидает на определенных выводах в ответ на выставленные воздействия. То есть пользователь сам определяет взаимосвязи и сообщает их системе. Здесь можно проводить несколько типов измерений. Пины перетаскиваются указателем мыши из левой области в измерительные окна.

В поле Watch можно наблюдать за состоянием выбранного пина любой JTAG-микросхемы. При этом используется JTAG-инструкция SAMPLE. Наблюдать можно в динамике, однако следует отметить, что частота сэмпирования сильно ограничена и зависит от частоты сигнала ТСК, поэтому при быстро меняющемся сигнале на внешней цепи некоторые состояния могут теряться.

В поле Buzz имитируется обычный мультиметр для «прозвонки» цепей. Выбираем два пина JTAG-микросхем (предполагая, что между ними должен быть физический контакт), нажимаем кнопку измерения и получаем зеленую или красную индикацию

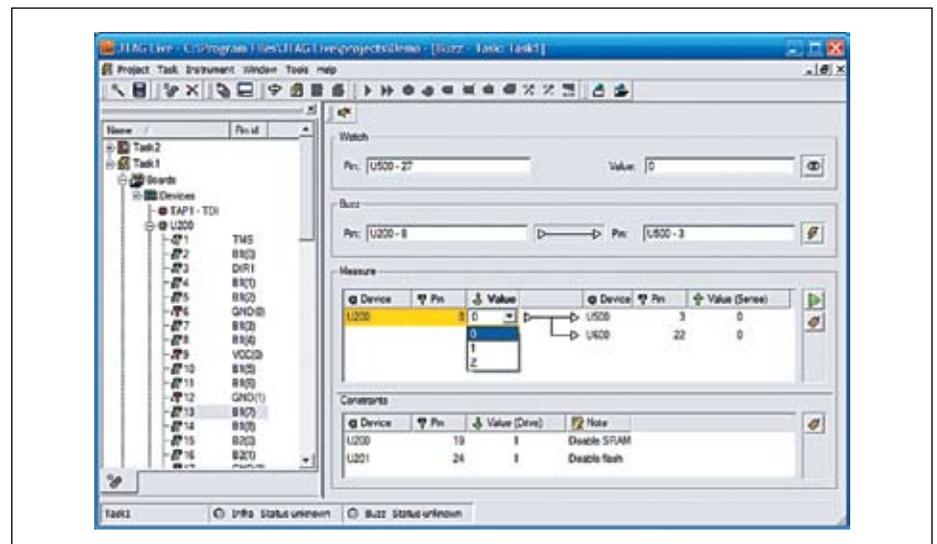


Рис. 8. Измерительная область в JTAG Live Buzz

о наличии или отсутствии контакта, а также звуковой сигнал «пищалку».

Поле Measure предназначено для снятия значений на определенных пинах при подаче воздействия с других. Ведь микросхемы с поддержкой JTAG необязательно могут быть соединены цепями напрямую: возможно, между ними присутствуют какие-то функциональные компоненты, не поддерживающие JTAG. Поэтому полученное значение на приемном конце не должно обязательно совпадать с выставленным на драйвере другого пина. Но пользователь должен сам знать для себя ожидаемые результаты: система не будет проверять правильность полученных логических уровней, она просто сообщит, получили ли мы 0 или 1. Проверить можно до 10 связей. Для полноценного тестирования логических кластеров этого недостаточно, но для определенных операций верификации вполне может пригодиться. Для сравнения: ProVision одновременно тестирует тысячи цепей, при этом для каждой цепи перебирает множество комбинаций, используя все участвующие микросхемы, а затем анализирует таблицу результатов для получения диагностических сообщений.

Вполне возможно, что у читателя возник справедливый вопрос: а как же быть с теми компонентами, которые мешают? Ведь, скажем, ProVision контролирует их при помощи моделей и держит сигналы отключения во время тестов межкомпонентных связей, а как быть здесь? Для этого в программе Buzz предусмотрено поле Constrains. Но, как вы уже поняли, вы сами должны догадаться, какой вывод какой микросхемы следует держать в нужном состоянии. На рис. 8 в этом поле заданы сигналы отключения микросхем памяти, однако комментарии типа "Disable SRAM" написаны автором вручную (автоматически они не появились).

Есть еще два программных продукта из семейства JTAG Live, которые называются Clip и Script. Они обладают несколько более широкими функциями, чем программа Buzz. Однако функциональность и того, и другого заложена в JTAG ProVision, но уже как дополнительные опции к основным возможностям.

Программа Clip полностью повторяет утилиту ActiveTest из профессионального пакета ProVision (о ней писалось в [1], поэтому мы не будем углубляться в ее описание). Clip, как и ActiveTest, позволяет вручную создавать тестовые векторы в интерактивном редакторе: обе программы имеют практически одинаковый интерфейс. ActiveTest оказался очень удобным для отладки, и его решили выделить в отдельный пакет Clip. Только, как мы уже знаем, JTAG Live не работает со схематикой платы, поэтому составление векторов хоть и производится в таком же графическом редакторе, но работа ведется не с цепями, а с выводами компонентов, поддерживающими периферийное сканирование. Это не всегда удобно, так как, не имея моделей промежуточных функциональных компонентов, система «не видит» всех сканируемых цепей, поэтому приходится вооружаться ручкой или карандашом и записывать, какие цепи фактически связаны с какими пинами JTAG-компонентов. JTAG ProVision же «видит», что цепь сканируется, даже если она уже прошла через буфер или другую логику, благодаря схематике и моделям. ActiveTest, в отличие от Clip, работает и с целыми цепями (когда компьютер решает, какие драйверы и сенсоры использовать) и с отдельными выводами компонентов. Название Clip происходит от слова «клипса» — клипса логического анализатора, только вместо нее здесь используются ячейки сканирования окружающих JTAG-компонентов.

Название третьего продукта JTAG Live — Script — говорит само за себя. Аналог Script в JTAG ProVision — это JFT (JTAG Functional Test). Только в ProVision JFT является дополнительным инструментом для тестирования сложных кластеров, а здесь — основным. Основа этого типа программ — язык программирования, позволяющий писать алгоритмы, сценарии и работать с переменными. Одновременно с этим требуется самому писать воздействия и считывать результаты в созданные переменные. Диагностика на основе полученных результатов также пишется программистом (если она вообще нужна). В связи с этим JFT является превосходным средством тестирования сложных

кластеров, внешних по отношению к компонентам с JTAG, но представляется довольно сложным инструментом для написания программ тестирования простых межсоединений. Собственно, JFT и разрабатывался как метод «квазифункционального» тестирования кластеров (компонентов или групп компонентов без поддержки JTAG).

Встроенный язык программирования хорош для работы с переменными, например, при тестировании данных с АЦП, управлении ЦАП, а также для работы с компонентами, требующими сложной инициализации, например картридерами и т. д. Так как Script полностью повторяет JFT, то его также не следует рассматривать как средство тестирования и диагностики плат, а скорее как специфический инструмент для специфических целей. Хотя можно отметить, что есть созданные тесты в JFT (Script) даже для тестирования Flash, но это гораздо более трудоемкий процесс, чем автоматическая генерация тестов на основе моделей. Кроме того, даже если вручную описать в виде алгоритма, что нужно сделать, чтобы считать ID-микросхемы флэш, то после запуска тестов вы не увидите ни таблицы с результирующими битами и цепями, ни диагностику, только если вы сами не напишете программу для обработки результатов. То есть Script предоставляет полную свободу творчества, но эта свобода хороша только для отдельных видов кластеров.

В связи с вышеперечисленным автор хотел бы оспорить еще одно предубеждение, что Script (или его аналог в JTAG ProVision — JFT) является самым высшим уровнем (наиболее автоматизированным) программ семейства JTAG Live. Это не так: программы нужно писать вручную. Просто Script выполняет задачи какой-то конкретно определенной области и в отдельном виде для тестирования не очень годится. Программа Script скорее подходит опять же разработчикам, которые, во-первых, хорошо знакомы с программированием, а во-вторых, им требуется отладка периферийных кластеров, окружающих компоненты, поддерживающие JTAG, для чего можно написать унифицированные алгоритмы.

Вообще JTAG Functional Test и Script — это тема отдельной статьи. Поэтому автор был бы признателен тем читателям, которые сообщат, требуется ли осветить данную тему в отдельной статье или нет. ■

## Литература

1. Иванов А. В. Новые возможности тестирования плат при помощи периферийного сканирования // Компоненты и технологии. 2010. № 5.
2. Иванов А. В. Два направления JTAG Technologies // Производство электроники. 2010. № 3.
3. Иванов А. В. Ключевые моменты тестопригодной разработки // Электронные компоненты. 2010. № 4.