

Программы JTAG Live. Бесплатный цифровой тестер Buzz

Технологией периферийного сканирования успешно пользуются уже более 20 лет, как во всем мире, так и в России. Возможность тестирования через интерфейс JTAG собранных печатных плат, определения и локализации дефектов, возникающих при монтаже, уже встроена в цифровые микросхемы, использующиеся в электронных изделиях.

Гиви Чхутиашвили

givi@jtag.com

З апросы от клиентов были зачастую таковы, что им не требовалась серьезная система в полном объеме, готовая работать с большим количеством JTAG-компонентов, с моделями микросхем и с автоматической генерацией тестовых векторов. Вместо этого они хотели пользоваться простой программой, позволяющей сделать очень простой тест, который помог бы проследить прохождение цифрового сигнала по элементам печатной платы. Причем сделать это как можно быстрее, не затрачивая времени на отладку самого теста. Таким образом, гарантированно готовый к выполнению тест после своего запуска мог бы дать максимально точную информацию о соответствии исследуемого изделия принципиальной схеме, программному проекту, об отсутствии дефектов монтажа при сборке, о «правильности» изготовления печатной платы.

Принимая это во внимание, компания JTAG Technologies в 2010 году открыла новое направление, которое получило название JTAG Live, сформировавшее новую линейку программно-аппаратных средств. Девизом стала фраза “Making JTAG accessible!”: теперь технология периферийного сканирования стала по-настоящему доступной.

Отличия JTAG Live от ProVision

Что отличает новую систему JTAG Live от ставшей уже стандартом системы периферийного сканирования JTAG ProVision? Кроме запоминающегося логотипа ярко-оранжевого цвета существует несколько принципиальных различий между этими программами в подходе к тестированию. Упомянем только об основных из них.

Во-первых, JTAG Live не требует списка соединений изделия — нетлиста. Это отличие является самым главным преимуществом. Очень часто разработчик начинает отлаживать макет изделия, принципиальная схема которого еще не утверждена после прошедших изменений, или доработки возникают прямо во время процесса отладки. Соответственно, при таком подходе не формируется нетлист, который можно экспортировать из САД-системы. Для работы JTAG Live это не проблема — достаточно знать, как организованы цепочки микросхем с поддержкой периферийного сканирования.

Во-вторых, программа JTAG Live поддерживает работу с разнообразными контроллерами, с помощью которых производится подключение к исследуемому изделию. Это контроллеры JTAG Technologies: серии DataBlaster, промышленный RMI-контроллер, двухпортовый контроллер JT 3705 USB и специально разработанный однопортовый контроллер JTAG Live USB, являющийся полностью бюджетным вариантом. Можно также использовать имеющиеся у вас кабели Altera и Xilinx, причем для последнего поддерживаются как USB-версия, так и LPT.

В-третьих, одна из программ пакета, а именно, JTAG Live Buzz, абсолютно бесплатна. Ее можно скачать прямо сейчас с сайта www.jtaglive.ru и пользоваться без ограничений. В целом система JTAG Live модульная (равно как и старшая система ProVision), с гибким механизмом лицензирования. Доступность программы — ключевой фактор в продвижении продукта, поэтому стоимость лицензии на один модуль на порядок ниже, чем у полновесной системы ProVision.

В-четвертых, при генерации тестовых приложений процессы в старшей системе полностью автоматизированы, а JTAG Live требует ручного управления. До определенной степени, конечно, ведь тестовые векторы, которые отправляют в цепочку, все равно подготавливаются без участия человека. Автоматизация, надо отметить, иногда является барьером для разработчика, который хочет управлять электрическими цепями по своему усмотрению.

В-пятых, необходимо упомянуть про невозможность использования моделей компонентов. У старшей системы файлы моделей содержат описание основных функций, а также методы переключения режимов работы. В ProVision, например, для отключения компонента достаточно зайти в его контекстное меню и выбрать пункт Disabled. JTAG Live, не имея нетлиста, «знает» только о наличии в схеме JTAG-микросхем, всех остальных элементов для программы просто не существует. Поэтому управлять компонентами, не имеющими JTAG-интерфейса, можно только путем непосредственного управления ячейками периферийного сканирования. При этом оператору необходимо иметь принципиальную схему изделия и техническое описание элементов.

И, наконец, шестое отличие — отсутствие диагностики дефектов и отчетов по тестированию. Программа JTAG Live задумывалась не как инструмент для производственного тестирования (хотя на ее основе можно создать серьезную тестовую станцию), а как недорогое решение для лабораторного использования, небольших разработческих компаний, ремонтных сервис-центров. Технически грамотный специалист, используя все возможности программы, сможет провести диагностику неисправностей самостоятельно, анализируя результаты выполненных тестовых приложений. Отчеты по тестированию главным образом затребованы в системах, которые находятся в производственных линиях, когда проверка каждого изделия должна быть зафиксирована в протоколе или маршрутном листе с привязкой к серийному номеру.

Типы приложений JTAG Live

JTAG Live использует проектный подход к организации работы с исследуемым изделием. Создаваемый пользователем проект содержит тестовые приложения, которые можно запускать на выполнение по отдельности. На данный момент доступны для использования шесть типов приложений:

1. Buzz. Инструмент для прозвонки и наблюдения за состоянием JTAG-цепей.
2. BuzzPlus. Инструмент для поиска всех цепей, соединенных с выбранной цепью.
3. AutoBuzz. Инструмент для обнаружения всех межсоединений изделия. Есть возможность обучения и последующего сравнения с «золотой» платой.
4. Clip. Инструмент для создания тестовых векторов вручную. Есть возможность графического отображения осциллограмм сигналов.
5. Script. Инструмент для написания сценариев на языке Python, позволяющих управлять выводами или группами выводов JTAG-компонентов.
6. CoreCommander. Инструмент, позволяющий контролировать процессорное ядро с помощью функций отладки.

Терминология внутри системы JTAG Live такова, что приложения здесь именуются задачами (Task). Это подчеркивает ориентированность программы на решение конкретных вопросов, возникающих в процессе отработки дизайна или отладки изделия.

Какие исходные данные потребуются?

Перед тем как создавать проект, разработчику необходимо получить BSDL-файлы (Boundary-Scan Description Language, язык описания периферийного сканирования) для всех JTAG-микросхем изделия с описанием архитектуры тестовой логики каждого компонента. Файлы эти не содержат какой-либо секретной информации, и их обычно можно свободно скачать с сайта производителя, после чего загрузить в проект. Информация о применяемых компонентах сохраняется в менеджере компонентов (Device Manager).



Рис. 1. Внешний вид демоплаты JT 2156 и контроллера JTAG Live USB

Так как JTAG Live не использует нетлист изделия, от пользователя требуется описать JTAG-цепочку (или цепочки), указав порядок следования микросхем в ней (в них). Необходимо сообщить дополнительно, что программа поддерживает работу одновременно с четырьмя цепочками.

Для того чтобы продемонстрировать работу в системе JTAG Live, мы выбрали новую демоплату компании JTAG Technologies — JT 2156 (рис. 1). Она содержит одну JTAG-цепочку, состоящую из двух компонентов: процессора NXP и FPGA Altera Cyclone III. Плата имеет массу других элементов, позволяющих оценить возможности технологии периферийного сканирования: память DDR2, микросхему EEPROM, контроллер Ethernet, контроллер USB, акселерометр, температурный датчик с интерфейсом I²C, контроллер сенсорной панели, логические буферы, кнопки, светодиоды и т. д. Вследствие того, что на плате находится одна цепочка сканирования, при подключении к тестовому разъему J2 достаточно будет воспользоваться простым однопортовым контроллером JTAG Live USB. Произведем коммутацию, после чего подадим питание на плату.

Создание первого проекта

Теперь обратим наше внимание на дисплей компьютера. При запуске программы JTAG Live Buzz первое, что мы видим, это окно помощника (рис. 2), спрашивающее, какое действие мы хотели бы предпринять. К нашему выбору один из трех вариантов: создать новый проект, открыть один из существующих проектов или открыть сделанный ранее архив JTAG Live. Выберем первый вариант и создадим абсолютно новый проект. В следующем окне помощник предложит нам ввести имя проекта и расположение папки, которая будет хранить всю информацию об исследуемом изделии. Выполним требуемое и перейдем к следующему шагу.

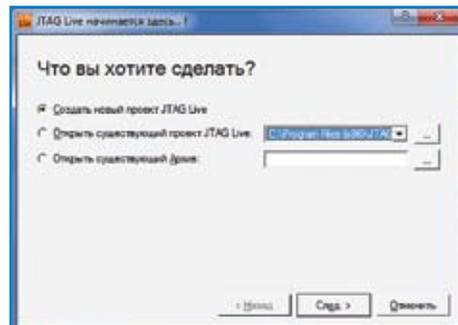


Рис. 2. Окно помощника

Перед нами стоит задача определить цепочки сканирования. Но сначала выберем подходящий тип контроллера, нажав на кнопку 1 «Свойства инструмента» (рис. 3), чтобы избежать возможных сообщений об ошибках. В поле «Предпочитаемый инструмент» теперь должна появиться надпись JTAG Live USB controller. Чтобы система начала автоматический поиск JTAG-компонентов, достаточно нажать на кнопку 2 — «Цепочки сканирования» (рис. 3).

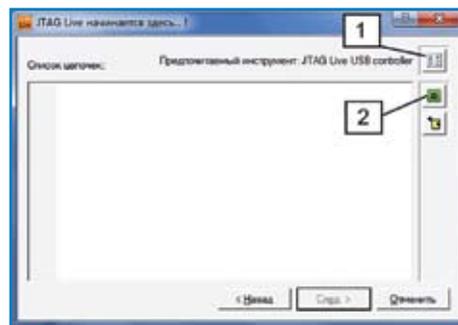


Рис. 3. Выбор инструмента

Через пару секунд в окне помощника появится найденная цепочка (рис. 4). Система JTAG Live сама определила не только порядок следования микросхем, но и получила содержимое их идентификационных регистров в виде ID-кода. 32-битный регистр идентификации содержит информацию о производителе микросхемы, которую программа отобразила для наглядности не в двоичном, а в понятном для нас текстовом виде: NXP Semiconductors и Altera.

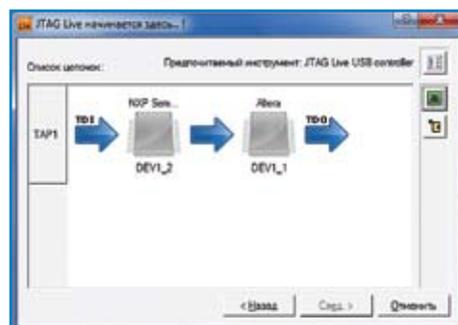


Рис. 4. Автоматический поиск JTAG-цепочки

Если дважды кликнуть на пиктограмму микросхемы из цепочки, то можно увидеть свойства компонента (рис. 5). Здесь нелишним будет в поле «Название компонента» заменить значение, заданное по умолчанию, на удобное для нас, например NXP для первой микро-

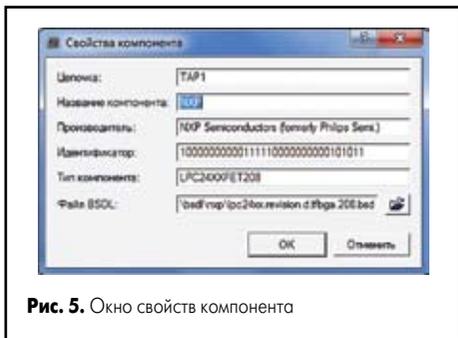


Рис. 5. Окно свойств компонента

схемы и Altera для второй. Если вам удобнее обращаться к устройствам по обозначениям из принципиальной схемы — U4 и U12, то выберите такой вариант.

И самое главное — в окне свойств компонента в поле «Файл BSDL» указать путь к этому файлу, который, повторимся, должен быть заранее скачан с сайта производителя микросхемы. Будьте внимательны, выбирая файл, так как для одного и того же компонента могут существовать варианты в разных корпусах (BGA, TQFP, SSOP и т. д.) и с разным количеством выводов. Структуры регистров периферийного сканирования, которые описываются в BSDL-файлах, для разных корпусов могут отличаться.

После того как вся необходимая информация введена, пользователю будет предложено завершить подготовку проекта нажатием соответствующей кнопки. Однако работа помощника на этом не будет окончена, и он предложит нам создать первое приложение (задачу).

Внешний вид программы

В бесплатной версии программы JTAG Live пользователь может создать только один тип приложения — Buzz. Выбор осуществляется в раскрывающемся списке «Тип задачи» (рис. 6). Дадим имя нашему приложению в поле «Название задачи». Будьте внимательны, не используйте здесь кириллицу и пробелы. Символ подчеркивания и цифры допустимы. Приложение готово к использованию уже после нажатия кнопки ОК.

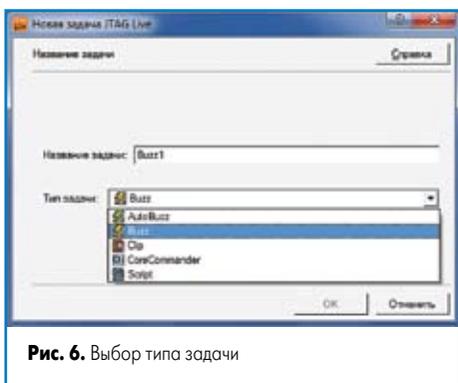


Рис. 6. Выбор типа задачи

В левой части окна программы JTAG Live располагается менеджер проекта (**Project Manager**), который наглядно отображает список всех задач (рис. 7). Внутри каждой задачи видна ее иерархическая структура: платы (Boards) и микросхемы (Devices). Список микросхем заключен между двумя

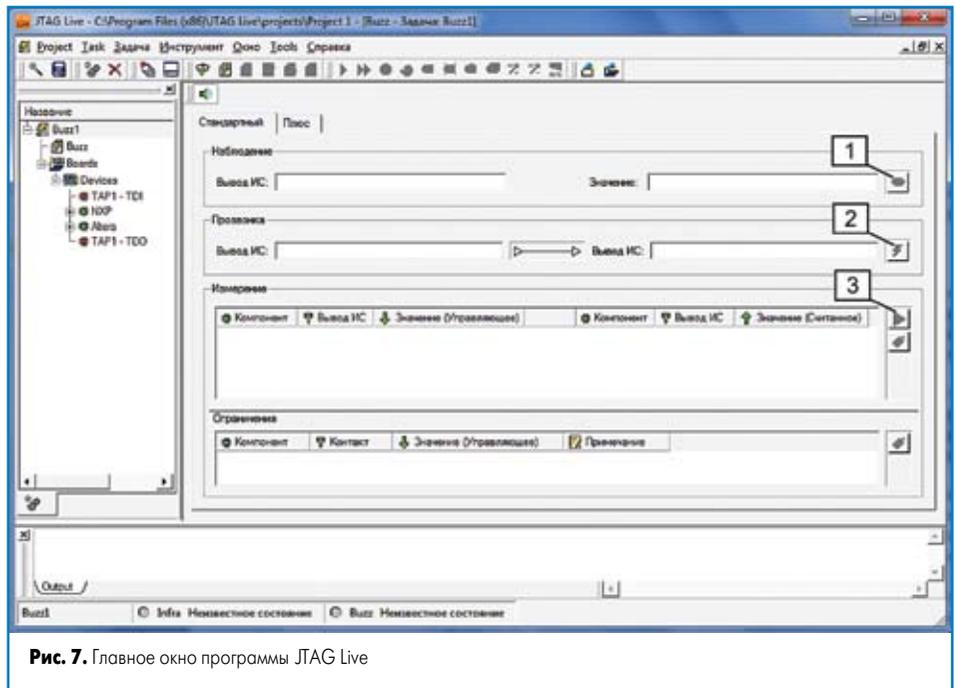


Рис. 7. Главное окно программы JTAG Live

элементами с пиктограммами красного цвета — TAP1. Они определяют цепочку периферийного сканирования нашего устройства и показывают порядок микросхем внутри нее. Тестовые векторы (последовательности нулей и единиц) будут проходить по следующему пути:

- выход JTAG-контроллера (TAP1-TDI),
- вход TDI первой микросхемы в цепочке (NXP),
- выход TDO первой микросхемы в цепочке (NXP),
- вход TDI второй микросхемы в цепочке (Altera),
- выход TDO второй микросхемы в цепочке (Altera),
- вход JTAG-контроллера (TAP1-TDO).

Обратите внимание, коммутация контроллеров для тестирования от JTAG Technologies осуществляется по принципу: TDI к «TDI платы», TDO к «TDO платы».

В нижней части окна ведется протоколирование работы программы (Logging). Причем к вкладке Output, где отражаются действия с проектом, добавляется вкладка Execution Log, в которой дается отчет по запуску отдельных приложений с фиксацией текущего времени.

Под протоколом также имеется стандартная строка статуса, в которой показана активная на данный момент задача и индикатор, сигнализирующий о том, что она запущена.

Для того чтобы открыть основное окно с редактором приложения, необходимо дважды кликнуть на элементе **Buzz** в менеджере проекта. В результате редактор займет всю оставшуюся часть окна программы JTAG Live. Рассматриваемое приложение позволяет выполнить три тестовые функции, соответственно, поле редактора разделено на три области (панели), каждая из которых озаглавлена: «Наблюдение», «Прозвонка», «Измерение». Далее мы более детально изучим содержимое каждой панели, связанное с конкретной функцией.

Тестирование инфраструктуры

Перед тем как запускать какой-либо тест, необходимо убедиться в том, что существующая на плате JTAG-цепочка (или цепочки, если их несколько) работоспособна, ведь в противном случае отправлять тестовые данные в никуда и анализировать неизвестно что не имеет практического смысла. Эта процедура называется тестом инфраструктуры, или инфрагестом. Проверка инфраструктуры не будет успешной в случаях, если на плату не подано питание или оно не доходит до JTAG-микросхем, осуществляется плохая или вовсе неверная коммутация контроллера с тестовым разъемом платы, установлены неверные JTAG-микросхемы (при совпадении корпусов для разных устройств), есть непройрка или иной дефект монтажа на выводах тестового порта.

Для того чтобы начать проверку, следует зайти в меню **Task** и выбрать «Тест инфраструктуры». Индикатор в статусной строке начнет мигать, и через две секунды мы увидим сообщение с результатом прохождения теста (рис. 8). В нашем случае все прошло успешно, цвет сообщения зеленый. Кроме того, нашему вниманию предлагается таблица тестовых векторов, чтобы мы проконтроли-

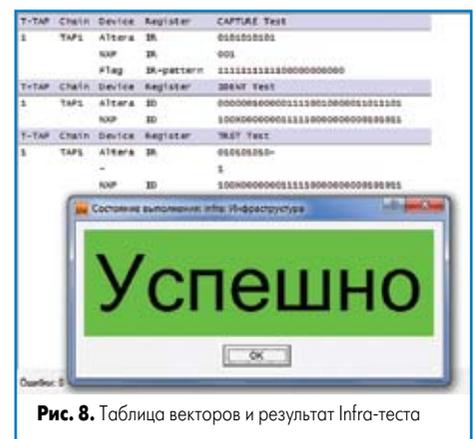


Рис. 8. Таблица векторов и результат Infra-теста

ровали, из чего состояла проверка. На первом этапе выполнялся CAPTURE-тест, он проверил целостность цепочки периферийного сканирования, то есть то, что TDO-выход одной микросхемы подсоединен к TDI-входу другой. В результате содержимое регистров инструкций, внутренних регистров JTAG-микросхем, последовательно выдвигалось по цепочке и достигло контроллера. Программа сравнила полученные данные с данными из BSDL-файлов на микросхемы и не нашла никаких отличий. На втором этапе проходит IDENT-тест, из компонентов цепочки извлекается содержимое регистров идентификации (ID) и также сравнивается с информацией из файлов BSDL. 32 бита, полученные в результате отправки инструкции IDCODE, содержат информацию о производителе микросхемы (11 бит), ее типе (16 бит) и версии кристалла (4 бита). Заключительный третий этап — TRST-тест. Он проверяет работу механизма сброса тестовой логики. Если у компонента имеется пин TRST, то программа произведет асинхронный сброс, используя его (NXP). Если нет, то сброс выполнится стандартным синхронным способом по четырехпроводному интерфейсу (Altera).

Функция «Наблюдение»

Первой функцией приложения Buzz, которую мы рассмотрим, будет «Наблюдение». При помощи нее можно узнать, какой логический уровень присутствует на выводе JTAG-микросхемы в реальном времени. Здесь необходимо сделать оговорку о реальном времени. Из-за ограниченности частоты сэмплирования, которая зависит от частоты ТСК и длины цепочки периферийного сканирования, быстрые колебания, скорее всего, зафиксировать не удастся, однако статические или медленно меняющиеся сигналы доступны для наблюдения.

Возвратимся к нашей демоплате и проверим, насколько хорошо функционирует такой простой элемент, как кнопка. В соответствии со схемой на рис. 9 пин F20 микросхемы Altera, к которому подключена кнопка SW4, подтянут к цепи питания, а второй порт кнопки соединен с «землей». Соответственно, в свободном состоянии на этом пине должен быть высокий уровень, а при нажатии кнопки — низкий.

Все, что нужно для начала тестирования, — это развернуть плюстик около выбранного компонента в менеджере проекта и найти там требуемый вывод. Если кликнуть на заголовки колонок «Название», то все выводы микросхемы отсортируются в алфавитном порядке, что упростит поиск. Также быстро переместиться на конкретный вывод можно, набрав на клавиатуре его номер или имя (для корпусов BGA). После этого «перетащите» пиктограмму пина в поле «Вывод ИС» функции «Наблюдение». Нажмите кнопку 1 (рис. 7) для того, чтобы запустить приложение. Теперь в поле «Значение» появилась та самая логическая единица, которую мы и ожидали увидеть. Сканирование выбранного пина осуществляется непрерывно. Это можно проверить, нажав на кнопку SW4 на плате несколько раз:

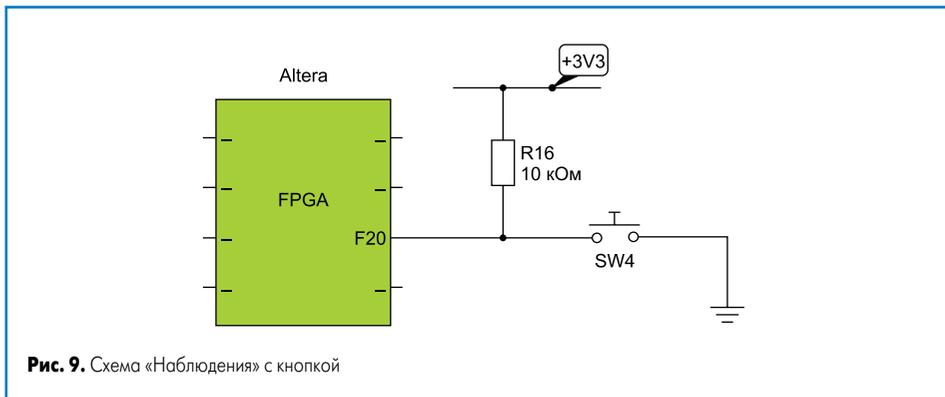


Рис. 9. Схема «Наблюдения» с кнопкой

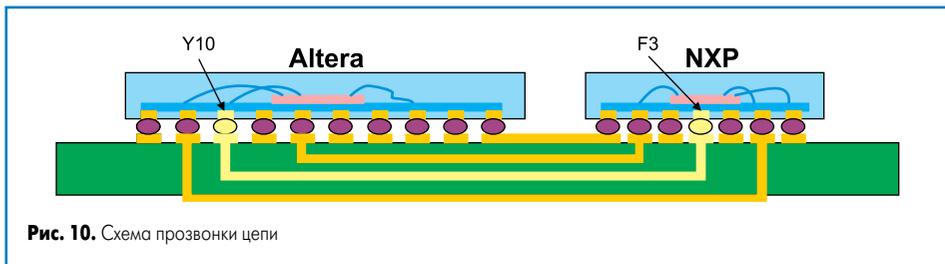


Рис. 10. Схема прозвонки цепи

считываемое значение будет в эти моменты сменяться на ноль.

В этом простом примере оказались успешно протестированными: монтаж шарика BGA-микросхемы, печатные проводники платы, по которым шел сигнал, правильность монтажа кнопки и ее работоспособность. Также прошел проверку резистор Pull Up, не на соответствие номиналу, естественно, а на наличие и выполнение своей функции. Инженер-аппаратчик теперь может быть уверен: если в функциональном режиме кнопка SW4 не работает, проблема точно в программе. И нужно связаться с программистом, писавшим «прошивку».

Функция «Прозвонка»

Перейдем к функции «Прозвонка», она позволит нам проверить наличие связи на печатной плате между двумя wybranными JTAG-пинами микросхемы. На рис. 10 в разрезе схематично представлена одна из таких связей, расположенная на нашей демоплате JT 2156. Электрическая цепь соединяет выводы Y10 у Altera и F3 у NXP. Обратите внимание, что цепь находится в слоях платы и наружу нигде не выходит. Если учесть, что мы имеем дело с корпусами BGA, то физически прозвонить такую цепь в процессе отладки или проверки устройства становится вообще невозможно.

JTAG Live Buzz — подходящее решение для такой задачи. Как и в предыдущем случае, найдем в дереве и «перетащим» два вышеуказанных пина в поле «Вывод ИС» функции «Прозвонка», а затем нажмем на кнопку 2 (рис. 7). Если связь наличествует, стрелка между выводами интегральных схем окрасится в зеленый цвет, если связи нет, то в красный. Дополнительно, если к вашему компьютеру подключена аудиосистема, вы услышите сигнал: в первом случае приятный, а во втором — похожий на сигнал тревоги.

Кроме печатной дорожки как таковой мы фактически протестировали не только монтаж шариков BGA-микросхем, но и разварку

кристалла внутри корпуса. Типичный случай использования функции «Прозвонка» — это поиск дефектов монтажа или печатной платы на неработающем устройстве.

Функция «Измерение»

Третья функция, доступная внутри задачи Buzz, — «Измерение». Она предназначена для выставления управляющего уровня и последующего считывания состояния цепи в случаях, когда цепь соединяет несколько JTAG-выводов. Также эта функция может работать с логическими узлами, которые позволяют описать свой функционал таблицей истинности (буферы, инверторы, ключи и т. д.). Таким образом, определенным логическим значениям на входах такого устройства всегда будут соответствовать определенные значения на выходах. Функция «Измерение», кроме того, дает возможность установить на JTAG-пине не только «1» или «0», но еще и перевести его в высокоимпедансное состояние, иначе называемое HighZ.

На нашей демоплате проверим цепь, соединяющую следующие пины: NXP.B17, NXP.E17, Altera.U11 (рис. 11). С цепью имеет соединение сигнал DQS с выхода микросхемы оперативной памяти U19. Обратите внимание, на схеме указана дополнительная управляемая с JTAG-компонента цепь, сигнал выбора микросхемы U19 — CS_. Снова «перетащим» выбранные пины из менеджера проекта на поле редактора (рис. 12). Причем слева расположим источник сигнала (драйвер), а справа — приемники (сенсоры). Очевидно, что в любой момент оперативная память может помешать корректно провести тест. Как минимизировать это влияние? Поле «Ограничения» служит как раз для таких целей, когда требуется принудительно выключить какой-либо компонент или удерживать какие-либо цепи в одном состоянии. Перетащим туда JTAG-пин Altera.G9, который управляет сигналом CS_ устройства ОЗУ. Установим управляющее значение равным «1», чтобы выбора микросхемы не происходило, так как активный уровень для этой

цепи — низкий. Дополнительно впишем примечание: «Отключение U19», чтобы при следующем обращении к проекту без труда вспомнить, для каких целей введено ограничение. Осталось установить на драйвере NXP. B17 «1», а затем нажать на кнопку 3 (рис. 7). Через мгновение мы получим в считанных значениях две единицы, то есть сигнал дошел до сенсоров. Если сменить уровень драйвера на «0», на приемниках также считаются нули. Полагаем тест завершенным успешно. Микросхема U19 при этом была отключена для обоих измерений.

Функция позволяет использовать одновременно не более 10 выводов для установки значений и не более 10 выводов для считывания, однако этого количества зачастую достаточно, чтобы провести проверку 8-разрядной шины.

Сохранение проекта

В процессе работы не забывайте сохранять свой проект. А для перенесения проекта на другой компьютер пользуйтесь функцией архивирования: в меню Project выбирайте пункт «Создать архив». В таком случае в одном файле будет сохранена вся структура внутренних папок и все пользовательские файлы, включая файлы BSDL. Для разархивирования проекта в заново открытой программе на первом шаге помощника необходимо выбрать пункт «Открыть существующий архив» и указать пути для извлечения файлов.

Заключение

Пользоваться программой JTAG Live Buzz чрезвычайно просто. Для этого нет необходимости разбираться в средах программирования ПЛИС или знать особенности компиляторов типа «С», загружать какую-либо прошивку в микросхемы. Поэтому мы советуем вступать в ряды пользователей JTAG Live тест-инженеров, инженеров по применению

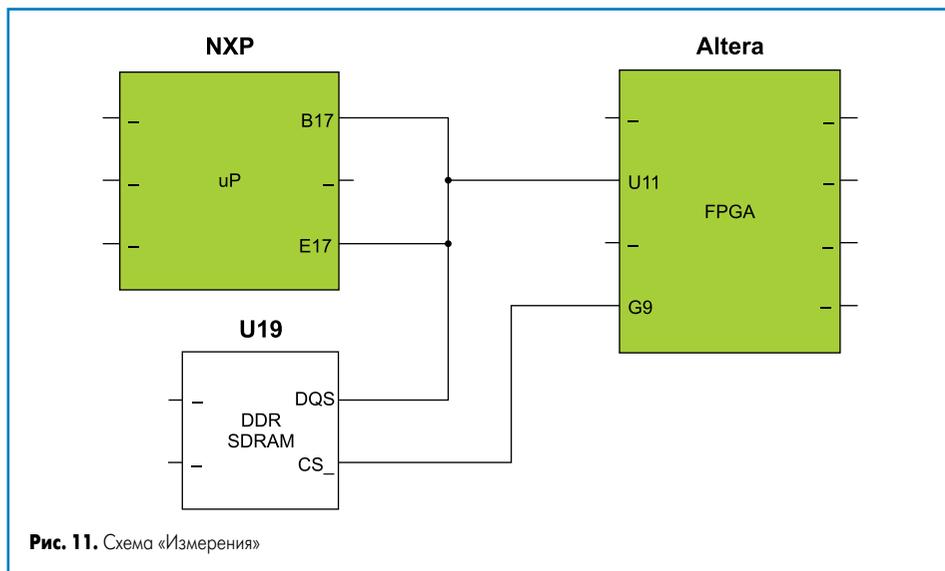


Рис. 11. Схема «Измерения»

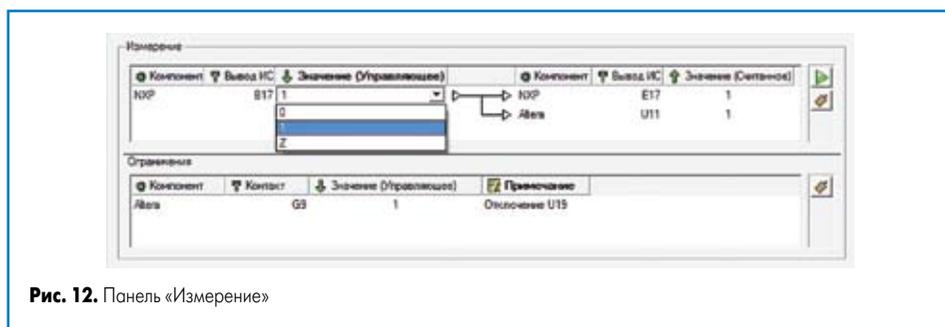


Рис. 12. Панель «Измерение»

и настройщиков радиоэлектронной аппаратуры. Каждый разработчик обязан иметь такой удобный инструмент в своем рабочем арсенале.

Если вы решили с программами JTAG Live использовать кабель ByteBlaster от Altera, то на компьютере должно быть установлено программное обеспечение Quartus. В случае использования адаптеров Xilinx должен быть установлен пакет ISE WebPack (включая iMPACT) версии не менее 11.1. Более подробно о программных требованиях к ка-

белям сторонних производителей можно посмотреть на сайте www.jtaglive.ru в разделе «Поддержка — FAQ».

Также следует отметить, что микросхемы Altera и Xilinx, как правило, не используют опциональный сигнал TRST. Поэтому в случае, когда ByteBlaster или Xilinx Platform применяются для работы с компонентами, имеющими в своем JTAG-интерфейсе этот сигнал (например, с сигнальными процессорами), необходимо убедиться, что он подтянут на высокий уровень.