



## НОВЫЕ ВОЗМОЖНОСТИ ТЕСТИРОВАНИЯ ПЛАТ ПРИ ПОМОЩИ ПЕРИФЕРИЙНОГО СКАНИРОВАНИЯ

Алексей Иванов  
alexey@jtag.com

**И**значально вариант заголовка этой статьи звучал так: «Новые возможности периферийного сканирования». Однако если вдуматься, то периферийное сканирование – это стандарт, который не менялся уже много лет. Определяет он логику регистров сканирования и JTAG-интерфейс микросхем, которые используются на тестируемых платах. Поэтому возможности периферийного сканирования изменяться никак не могут. Исключением может стать появление новых стандартов. Изменяются лишь системы проектирования тестов и оборудование для тестирования, предоставляющее все больше возможностей. Современные цифровые платы становятся настолько сложными и насыщенными, что в один момент стандартный набор автоматических процедур генерации JTAG-тестов исчерпал себя. В этой связи производители программного обеспечения и оборудования для периферийного сканирования стали искать новые подходы к данному вопросу. В статье приводятся наиболее интересные новые возможности при проектировании тестов в программном обеспечении компании JTAG Technologies, одна из которых – функциональный тест при помощи JTAG.

Итак, прежде чем перейти к новым возможностям, давайте разберемся со «старыми». Классические возможности создания тестов при помощи ATPG (Automatic Test Pattern Generation) можно продемонстрировать на примере проекта в системе JTAG ProVision. Проект строится на основе списка соединений из САПР (net-лист) и моделей компонентов. Этой информации обычно достаточно для генерации приложений для тестирования. При этом модели могут быть взяты из библиотеки или созданы при помощи редактора. Также существуют модели компонентов с поддержкой периферийного сканирования, называемые BSDL-файлами, которые можно загрузить с сайта производителя микросхем или которые входят в пакет документации. Классический процесс разработки тестов и, собственно, идеология ATPG показаны более подробно на рис. 1.

Данная структура уже доказала свою работоспособность, причем для достаточно больших участков платы, обеспечивая диагностику солидного процента неисправностей. В стандартные функции проектирования тестов на сегодня входят межэлементные связи, все виды памяти, логика, резисторы, тест разъемов и простые интерфейсные микросхемы. Основной вопрос, который возникает на данном этапе: достаточно ли всего вышеперечисленного для полноценного тестового покрытия платы?

Существуют участки схемы, где автоматическая векторная генерация с трудом выполняет свои задачи. С точки зрения технологии периферийного сканирования они называются «кластерами». Это могут быть аналого-цифровые компоненты, микроконтроллеры без поддержки JTAG, сложные интерфейсные микросхемы и т. д. Даже в том случае, когда к выводам таких «кластеров» есть прямой или косвенный доступ перифе-

рийного сканирования, для них не существует универсальных алгоритмов генерации тестовых векторов. Если взять обычную микросхему ОЗУ, то здесь все довольно просто: модель содержит информацию о циклах чтения/записи; проект содержит названия цепей, соединенных с данной ОЗУ. На основе этой информации генерируются сотни тестовых векторов, записывающих и считывающих данные в разных комбинациях, давая качественную диагностику дефектов. Если на плате установлена не одна такая микросхема ОЗУ, а, скажем, восемь, то тестовые приложения для остальных семи создаются одним щелчком с использованием той же самой модели.

Однако не все компоненты так же податливы для систем проектирования как память. Есть категория устройств, для тестирования которых

**Есть категория устройств, для тестирования которых требуется функциональный подход, как бы мы не стремились к идеологии структурного тестирования**

требуется функциональный подход, как бы мы не стремились к идеологии структурного тестирования. Такие задачи заставили искать новые методы «JTAG-общения» с «кластерами».

### НОВЫЕ «СТАРЫЕ» ВОЗМОЖНОСТИ

В этом разделе мы поговорим о программе ручного конструирования тестовых векторов Active Test от JTAG Technologies. Программа эта, нужно заметить, существует довольно давно. Просто до недавнего времени она представляла собой отдельное приложение, никак не связанное с системой проектирования JTAG ProVision. Теперь же, после интеграции, ее использование значительно упростилось. Связано это с тем, что в настоящее время Active Test является частью проекта ProVision и позволяет использовать оттуда готовую информацию (цепи, компоненты) для соз-

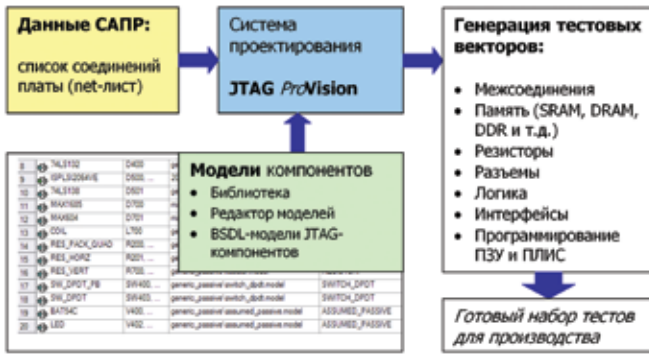


Рис. 1 Классический процесс разработки тестов в JTAG ProVision

дания тестовых векторов, а, кроме того, приложения Active Test теперь являются частью производственных последовательностей.

Выделим основную ценность этого «дополнительного» инструмента. Предположим, что в тестируемом изделии есть компонент, поддерживающий периферийное сканирование JTAG, и некий довольно сложный «кластер» (к примеру, микроконтроллер или ASIC, без таковой поддержки) (рис. 2).

Такой компонент, обозначенный на рисунке черным цветом, может создать трудности при автоматическом проектировании тестов. Ну, во-первых, его функциональность может зависеть от зашитой конфигурации, это может быть отечественный компонент, модель которого отсутствует в библиотеке ProVision (хотя в последнем случае можно создать самому). Собственно, это может быть СБИС своей разработки. Active Test в данном случае позволяет обойти «стандартную процедуру» и, к примеру, «сфотографировать» состояния цепей при определенных воздействиях на другие, идущие к «кластеру», и записать их в тестовые векторы. Это можно сделать на полностью исправной плате. Такая возможность вдвойне полезна для автономных производств, где порой бывает трудно получить от разработчиков описание функций того или иного компонента платы.

«Сфотографировать» результаты можно и на других выводах устройства, например, идущих на разъем или индикатор. В этом случае используется модуль I/O, подключаемый к разъему, или сам индикатор (светодиоды). Если светодиоды напрямую подключены к JTAG-компоненту, то задача зажечь их и убедиться в правильности установки с Active Test вообще представляется совершенно простой и заключается в создании одного вектора. При этом для оператора-тестирующего можно вывести предупреждение о проверке дисплея или светодиодов. На рис. 3 показан пример создания векторов с использованием Active Test. При считывании уровней сигналов с платы пользователь может выбрать на свое усмотрение любое количество цепей. После запуска теста пустые векторы заполняются реальными величинами, которые могут быть сохранены. Один вектор может содержать неограниченную цепочку бит, как для считывания, так и для записи.

## НОВЫЕ ПОДХОДЫ К ТЕСТИРОВАНИЮ «КЛАСТЕРОВ»

Только что рассмотренный пример с Active Test, несмотря на ручные методы работы, представляет собой векторный подход к тестированию



Рис. 2 Проблема с тестированием связей сложного «кластера»

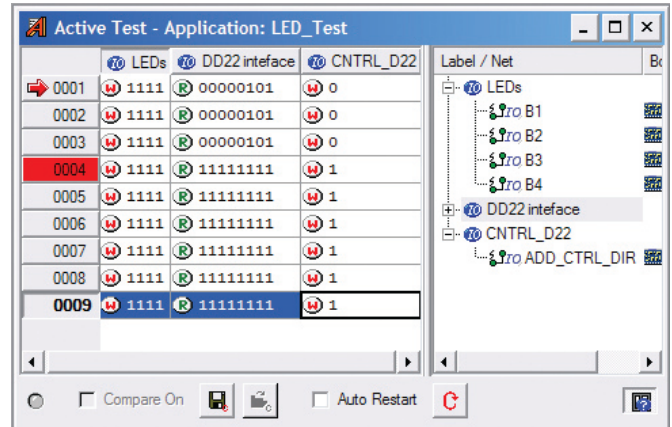


Рис. 3 Окно редактора тестовых векторов Active Test

при помощи периферийного сканирования. Просто векторы создаются не программами ATPG, анализирующими схематикку, а пишутся вручную или считываются с «золотой» платы.

Взглянем еще раз на рис. 2 и представим, что нам нужен полностью функциональный подход, симулируемый, однако, с помощью периферийного сканирования. Например, нужно имитировать какой-то интерфейс, произвести инициализацию регистров устройства, представляющего собой «кластер». А более наглядный пример – когда требуется протестировать работу АЦП и считать данные на цифровых выходах, сравнив их с предельно допустимыми значениями. Это уже совершенно точно функциональный подход.

В связи с этим появилось еще одно дополнение к JTAG ProVision – JFT (JTAG Functional Test). Данный программный инструмент использует встроенный язык программирования Python (Питон) – высокоуровневый открытый язык, развитие которого спонсируется такими гигантами как Microsoft и Google и поддерживается довольно большим сообществом программистов. Впрочем, Google сам использует Python для своих технических нужд.

Использование языка позволяет работать с переменными, что может оказаться очень удобно для вышеописанного случая с тестированием АЦП. При записи данных с тестируемых цепей в виде переменных с ними можно осуществлять разнообразные операции (рис. 4).

Для кластера, состоящего из одного или нескольких компонентов, можно создавать шаблонные модели на языке Python и использовать их в других проектах. В идеале, по замыслу разработчиков компании JTAG Technologies, вероятно образование интернет-сообщества инженеров, создающих модели кластеров на Python, которыми они могут делиться друг с другом.



Рис. 4 Пример программы для тестирования АЦП в JTAG Functional Test



Следует отметить, что описанные в данной статье методы, в особенности JFT, требуют уже некоторой фантазии и знания функциональности узлов схемы. Однако существует категория инженеров, которые любят «затачивать» даже «прагматичное» структурное тестирование под разнообразные нужды, в том числе и функциональную проверку.

Новые виды тестовых приложений Active Test и JFT можно также включать и в стандартные производственные последовательности. Приведем пример типичного набора для производства:

- тест JTAG-канала (проверка установки JTAG-компонентов);
- тест межкомпонентных соединений (может быть расширенным, с проверкой неподключенных выводов и т.д., с разъемами);
- тест соединений с памятью (может быть несколько в зависимости от количества и типа микросхем ОЗУ);
- тест нагрузочных резисторов, подключённых к шинам питания и земли;
- тест флэш-ПЗУ;
- Active Test: проверка светодиодов или сегментного дисплея с привлечением внимания оператора;
- программирование флэш-ПЗУ (с предварительным стиранием, если нет доверия к поставщику);
- программирование внутренней EEPROM контроллера без поддержки периферийного сканирования;
- JFT: проверка межсоединений с микроконтроллером без поддержки периферийного сканирования;
- сброс режима периферийного сканирования для запуска платы.

Все эти операции можно выполнить в виде единой последовательности (конечно при условии их создания) в JTAG ProVision и в отдельном секвенсоре AEX Manager. В каком порядке их выполнять и каковы условия перехода – дело предпочтений тестового инженера и элементарной логики.

На одном из пунктов последовательности, приведенной выше, следует остановиться более подробно. Это программирование внутренних EEPROM устройств (к примеру, специализированных небольших микроконтроллеров), не поддерживающих периферийное сканирование по стандарту IEEE 1149.1.

Для унификации производственного процесса в JTAG Technologies разработаны отдельные «специальные» программные модули для их прошивки, которые могут включаться в общую тестовую последовательность. Они не относятся к основному пакету проектирования и подключаются дополнительно. Список таких модулей и компонентов можно посмотреть на сайте [www.jtag-technologies.ru](http://www.jtag-technologies.ru). Здесь следует отметить, что данные программные модули не являются средствами разработки и отладки аппаратного ПО микроконтроллеров и разработаны лишь для того, чтобы адаптировать их прошивку к производственным пакетам JTAG Technologies для выполнения в общем цикле тестирования. В последовательность, приведенную выше, пункт с EEPROM микроконтроллера вставлен не случайно – после программирования микроконтроллера, не поддерживающего периферийное сканирование, можно использовать его функции для выполнения тестов JTAG Functional Test.

## НОВЫЕ АППАРАТНЫЕ ВОЗМОЖНОСТИ

В заключение поговорим о новых аппаратных решениях. Они коснулись и контроллеров периферийного сканирования. Например, недавно появились контроллеры серии RMI (Rack Mounting Instrument). Особенность их заключается в том, что помимо четырех TAP-портов у контроллера имеется еще 256 дополнительных цифровых каналов I/O (рис. 5). Это довольно важный шаг, так как современные цифровые платы обычно имеют много цепей, выходящих на внешние разъемы. Очень трудно найти изделие, в особенности среди модулей обработки данных, где количество таких внешних соединений было бы меньше 100. Опыт показывает, что использование модуля I/O при выполнении периферийного



Рис. 5 Контроллер JT3727/RMI. Слева находятся TAP-порты, а справа – цифровые каналы I/O

сканирования увеличивает тестовое покрытие в среднем на 40%. Кроме того, каналы I/O могут использоваться для включения/выключения каких-либо активных компонентов в процессе тестирования, подачи сигналов сброса на цепи, где нет доступа периферийного сканирования с компонентов самой платы и т.д., таким образом, обеспечивая более полноценный контроль платы при тестировании.

Отметим, что ранее модули ввода/вывода тоже существовали только

в виде отдельных автономных блоков, не встроены в контроллеры периферийного сканирования.

Теперь даже у «классических» контроллеров серии DataBlaster есть возможность заменить один из его четырех TAP-портов, которые имеют свойство отсоединяться, на небольшой модуль I/O JT2149 (рис. 6). В результате пользователь получает тестовую станцию уже с тремя (вместо четырех) JTAG-портами и 32-мя каналами цифрового ввода/вывода. Более того, модуль JT2149 поддерживает функцию SCIL (Scan-Configurable Logic). Это означает, что он может работать не только как набор параллельных I/O, но и симулировать, к примеру, какой-нибудь интерфейс. Функции SCIL могут быть заводскими или запрограммированы пользователем. Примеры реализации таких функций – программаторы устройств, не имеющие JTAG-протокола и использующие для прошивки другой интерфейс (например, SPI).

Завершая статью, хочется отметить, что в тестировании, как и в остальных инженерных сферах, очень многое зависит от фантазии и желания самого инженера увеличить тестовое покрытие. Современный инструментарий позволяет объединить все тестовые приложения периферийного сканирования и программирования устройств на плате в единые последовательности с разветвленной структурой, которые могут использоваться на производстве. Большая часть тестов для «стандартной» части платы может быть сгенерирована автоматически на основе моделей и схематики (60–80%), но обычно остается небольшая «нестандартная» часть, требующая функционального подхода, и вот теперь существует набор инструментов для его реализации. ■



Рис. 6 Модуль JT2149 с поддержкой функции SCIL