

Погружаясь в ПЛИС

Том ЛЕКЛИДЕР

С момента своего появления программируемые логические устройства (PLD) начали постепенно вытеснять дискретно-логические интегральные схемы. Дальнейшее развитие PLD до сложных PLD (CPLD) ускорило данную тенденцию, а последние разработки доступных, быстрых, больших матричных кристаллов с возможностью перепрограммирования (ПЛИС) почти завершили этот процесс.

Как удалось создать полноценный логический анализатор в таком компактном корпусе?

PLD и CPLD устройства могут работать очень быстро из-за своего сравнительно небольшого размера. Кроме того, их логические элементы включают комбинаторные ячейки с фиксацией состояний, что обеспечивает вычисление булевых уравнений. При создании файлов, необходимых для программирования PLD или родственных им программируемых логических матриц (PLA), программируемых массивов логики (PAL), или массивов логики общего назначения, разработчики используют языки ABEL и PALASM. Несмотря на различные варианты применения, способы декодирования адресов и диаграмм состояний используются одни и те же.

CPLD содержат множество областей, эквивалентных PLD, плюс необходимые средства коммутации. Однако не всегда использование таких устройств выгодно. Это связано с тем, что в комплексах CPLD невозможно реализовать логические комбинации с использованием всех элементов. Именно из-за этого они могут быть менее гибкими, чем PLD.

Безусловно, меньшие по размерам устройства все еще используются, но ведущие позиции занимают новые, большие и более быстрые ПЛИС. Причина для интенсивного и продолжающегося интереса к ним заключается в том, что ПЛИС преодолели точку, где затраты на их разработку и производство было трудно окупить. Для многих прикладных задач они являются вполне реальной альтернативой специализированным интегральным схемам (ASIC).

Существуют ПЛИС, которые, как и первые PLD, могут быть запрограммированы только однажды. Это так называемые однократно программируемые устройства, преимущество которых заключается в том, что они начинают работать с момента подключения к ним питания. Если они были запрограммированы некорректно или требовали доработки, но изменить что-либо уже невозмож-

но — и разработчики вынуждены программировать новое устройство.

Второй вид ПЛИС — это ПЛИС с возможностью хранения программ в статической оперативной памяти (SRAM). При этом конфигурация ПЛИС хранится во внутреннем ОЗУ, а инициализация осуществляется из внешней памяти. Преимущества данной технологии заключаются в нулевом времени программирования и нулевом складе уже запрограммированных микросхем. Кроме того, разработчики получают удобную возможность перепрограммирования таких устройств путем загрузки нового файла конфигурации.

Третий вид ПЛИС основан на использовании перепрограммируемой энергонезависимой памяти (флэш-памяти). В этих ПЛИС программа конфигурации загружается однажды, и инициализация и управление устройством обеспечивается этой программой. Устройства, базирующиеся на флэш-памяти и SRAM, перепрограммировать легко. Однако флэш-память обладает тем преимуществом, что не требует загрузки программы при каждом включении устройства.

В настоящее время стали доступны ПЛИС с комбинациями этих технологий. Например, семейство LatticeXP предлагает разработчику воспользоваться новой технологией Transparent Field Reconfiguration (TransFR или TFR). Эта технология обеспечивает реконфигурирование ПЛИС LatticeXP в «полевых» условиях без прекращения работы устройства, в котором она установлена. В ПЛИС LatticeXP установлены флэш-память и память типа SRAM. Линии управления поддерживают загрузку программ флэш-память в фоновом режиме через JTAG-порт или выделенный порт конфигурации, затем в выбранный момент времени фиксируются выходы, приостанавливается работа ПЛИС, обновляется содержимое SRAM из флэш-памяти, и управление возвращается к пользовательской логике.

Диапазон функциональных возможностей современных ПЛИС очень велик. Однако с этим связаны и новые требования. Главный вопрос, который приходится решать, должна ли схема устройства быть самостоятельной или

она будет выполнена на базе ПЛИС? В прошлом у разработчиков и руководителей проектов выбор был невелик. Если для конечного устройства была спроектирована программа большого объема, ASIC-решение часто оставалось лучшим способом обеспечить необходимую логику.

И сейчас ASIC остаются лучшим выбором для некоторых специфических изделий, но разница между ними и ПЛИС сокращается. Достоинства и недостатки двух подходов приведены в таблице 1. В случае создания устройств на основе ПЛИС небольшими компаниями, либо если разработчики плохо знакомы с возможностями ASIC, гибкость ПЛИС на основе перепрограммируемой флэш-памяти трудно переоценить. Когда возникает вопрос о сроках и стоимости обновления изделий на основе ASIC, решение на основе ПЛИС может значительно уменьшить затраты на разработку.

Именно из-за этого ПЛИС часто используются для моделирования решений на основе ASIC. Последствия ошибок, допущенных при проектировании ASIC, настолько велики, что разработка прототипа даже в дискретной логике может стать разумной альтернативой. ПЛИС допускает частые модификации проекта, так же как разделение его на части и отладку аппаратной реализации. Разработка прототипа с использованием ПЛИС облегчает решение проблем, связанных с параллель-

Таблица 1. Сравнение ПЛИС и ASIC (по данным компании Cadence Design Systems)

ПЛИС	ASIC
Более быстрая аппаратная часть	Более медленная аппаратная часть
Меньшее количество логических элементов	Большее количество логических элементов
Большая мощность	Меньшая мощность
Более высокая себестоимость	Более низкая себестоимость
Драйвер процессов ИС	Сложно отследить последний процесс
Не требует наличия топологии и физического воплощения	Наличие топологии и физического воплощения
Возможность внесения изменений: выполняется в течение нескольких минут	Возможность внесения изменений: выполняется в течение нескольких недель
Можно добавлять новые функции, менять протоколы, адаптировать в полевых условиях «на ходу»	Аппаратное устройство — нет возможности исправлений, пересмотра или адаптации

ной логикой, программным обеспечением и развитием системы в целом, поскольку различные части логики становятся доступными, как только завершается определенная секция проекта.

В результате для реализации все большего числа устройств разработчики используют ПЛИС, включая ПЛИС со встроенными процессорами и с небольшими объемами памяти. IP других поставщиков могут использоваться с логикой, разработанной специально для нового устройства.

Как узнать, что устройство работает?

Из-за того, что все больше функций устройства реализуется с помощью ПЛИС, отладка проекта становится серьезной проблемой. С аналогичной проблемой сталкивается любой неопытный разработчик ASIC, когда становится доступен первый кристалл. Пока он не стал частью проекта, его внутренние ресурсы почти недоступны. В то же время для отладки ПЛИС имеется множество решений, и главное требование, которое приходится учитывать разработчику, — это запланировать заранее возможности их использования.

ПЛИС, исходя из их названия, содержат множество программируемых ячеек, которые выполняют определенные логические функции. Каждый изготовитель имеет собственное представление о компромиссе между сложностью ячеек, гибкостью подключения и возможностями ввода-вывода. Рис. 1 демонстрирует некоторые элементы, из которых состоит LatticeXP. Большинство ПЛИС имеет либо флэш-память, либо память SRAM, но не оба типа памяти одновременно. Встроенные блоки RAM — относительно недавние дополнения ПЛИС. Также обратите внимание на встроенный JTAG-порт.

Архитектуру ПЛИС, которая имеет множество ячеек с относительно небольшим количеством элементов логики в каждой, называют мелкозернистой архитектурой. Поскольку в каждую ячейку включен один или несколько триггеров, мелкозернистые устройства хорошо подходят для описания диаграмм состояния, а также для конвейерных проектов, которые должны фиксировать промежуточные состояния, например для обеспечения синхронной работы. ПЛИС с большими ячейками (крупнозернистая архитектура), больше подходят для реализации широких комбинаторных логических функций.

Каждый производитель предлагает свой набор средств разработки для собственных изделий, и этот набор включает средства отладки. При этом некоторые из функциональных возможностей таких средств являются стандартными для технологий ПЛИС и не содержат почти ничего нового.

Например, серия antifuse (электрически замыкаемых перемычек) и Axcelerator FPGA

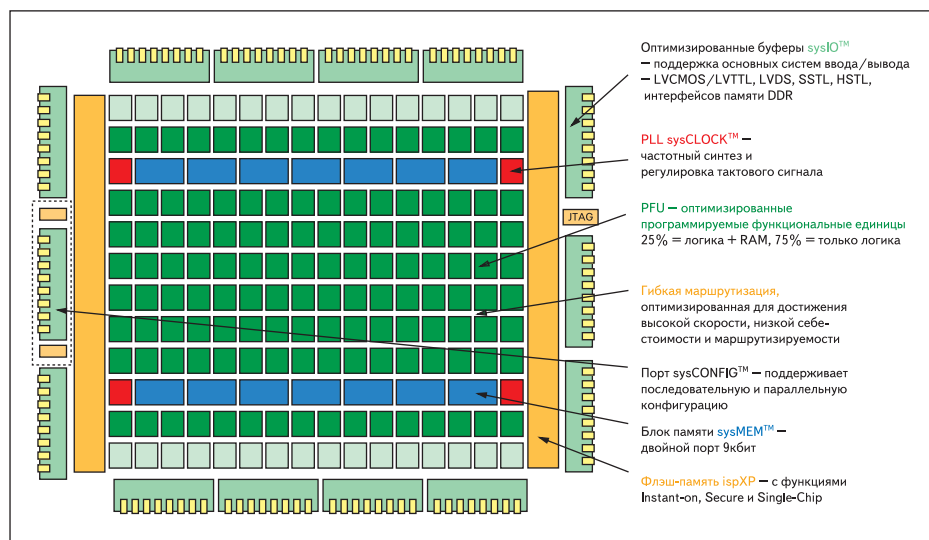


Рис. 1. Архитектура устройства LatticeXP (источник: компания Lattice Semiconductor)

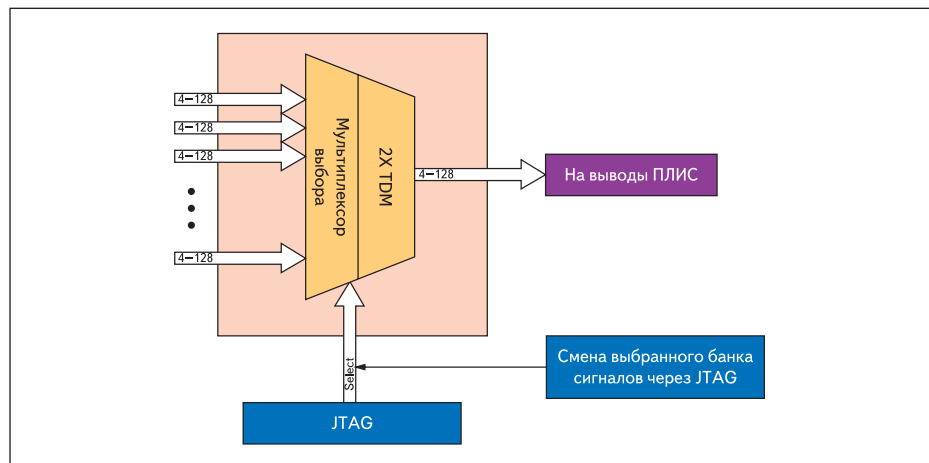


Рис. 2. Встроенное ядро Agilent Trace Core с управлением через JTAG-порт (источник: Agilent Technologies)

компании Actel имеют внутрисхемные средства отладки. Каждый узел схемы доступен для отладки благодаря фирменному решению Silicon Explorer и специализированному программному обеспечению. Аппаратная часть Silicon Explorer управляется через ПК и подключается к ПЛИС через JTAG-порт. В antifuse FPGA до появления серии Axcelerator были доступны два сигнала одновременно — как в синхронном режиме до 66 МГц, так и в асинхронном до 100 МГц. Для Axcelerator FPGA одновременно доступно 14 дополнительных сигналов.

Возможность исследования внутренних сигналов, реализованная по технологии antifuse, связана с выбором любой из электрически пережигаемых перемычек для программирования устройств. К сожалению, большинство других технологий ПЛИС, особенно на основе флэш-памяти и памяти SRAM, не имеют встроенных средств отладки. Разработчик должен создавать собственную схему отладки при проектировании ПЛИС.

В самой простой форме такая структура может быть реализована в виде мультиплек-

сора (mux), который подключается к одному внешнему выводу ПЛИС для передачи внутреннего сигнала и нескольким другим выводам для управления. Такая реализация занимает ценные выводы, количество которых ограничено. Вместо того чтобы непосредственно управлять мультиплексором, можно установить JTAG-контроллер. Регистр инструкций порта тестирования (TAP) JTAG-контроллера также может использоваться для управления мультиплексором.

Поскольку мультиплексор просто выбирает один сигнал из нескольких входных, выходные сигналы могут быть исследованы при помощи осциллографа или логического анализатора. Любые импульсные помехи или изменения длительности перепадов могут быть обнаружены этими приборами. Если есть возможность задействовать дополнительно другие выводы устройства, можно осуществить более детальный анализ с использованием двух, четырех или большего количества сигналов одновременно — с возможностью исследования их временных соотношений. Если один из сигналов будет так-

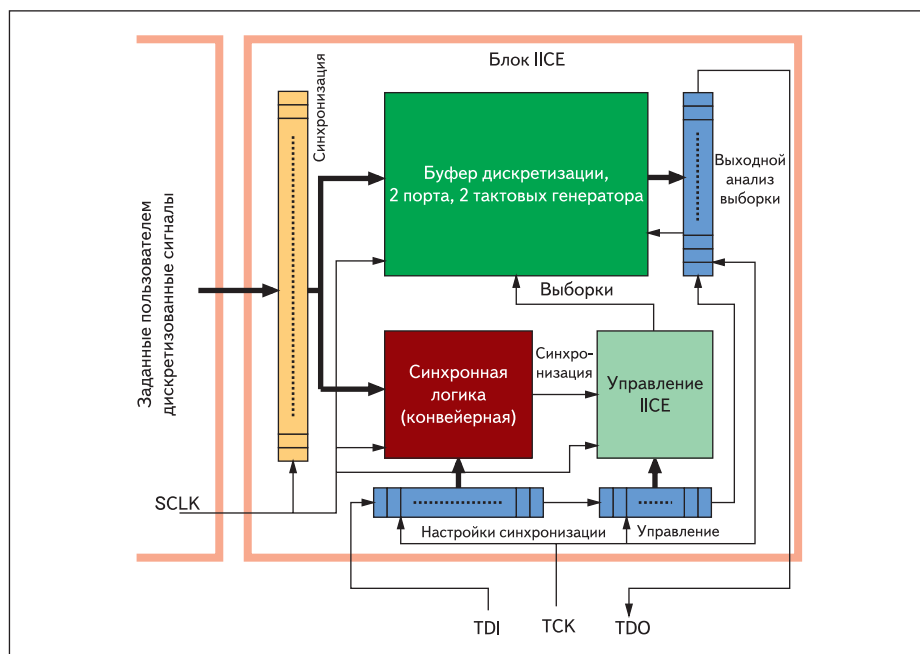


Рис. 3. Блок-диаграмма встроенного Intelligent In-Circuit Emulator (Источник: Synplicity)

товым, можно проанализировать диаграмму состояния устройства (рис. 2).

Кроме подхода, использующего для отладки внешние приборы, при наличии достаточного количества доступных свободных выводов можно создать встроенный анализатор состояний и реализовать некоторую возможность синхронизации в пределах ПЛИС. Благодаря этому подходу большая часть аппаратного обеспечения, необходимого для проверки правильности функционирования, становится частью законченного устройства.

Если одновременные состояния нескольких сигналов захватываются встроенными средствами, интерфейс JTAG может использоваться для их вывода через внешний информационный вывод (TDO). Так можно получить доступ к сотням различных внутренних сигналов (рис. 3), используя очень ограниченное число выводов. Внутренние средства отладки могут работать на полной скорости ПЛИС, потому что они фактически являются частью этого устройства.

Практическая сторона измерений

Измерения — важная и необходимая часть процесса разработки, и предусмотреть возможность тестирования в будущем — задача руководителя проекта. Проектирование и интеграция средств отладки для измерений не должны занимать много времени и требовать больших усилий: очень легко превратить в отдельный самостоятельный проект то, что начиналось как небольшой мультиплексор.

Необходимость проведения измерений и объем работ, связанный со всесторонним анализом функционирования устройств, стали учитываться производителями ПЛИС и из-

мерительного оборудования. Сотрудничество между двумя этими группами компаний привело к появлению различных средств отладки.

Изделия фирм Actel, QuickLogic и Atmel можно тестировать с помощью нескольких встраиваемых логических анализаторов на базе решения Logic Navigator IP компании First Silicon Solutions (FS2). Компании Agilent Technologies и Xilinx разработали встраиваемое ядро трассировки второго поколения (ATC2), поставляемое с набором средств разработки Xilinx ChipScope Pro. Компания Tektronix совместно с компанией FS2 разработала способ отладки ПЛИС Altera с использованием внешних измерительных приборов. Кроме того, некоторые из вышеперечисленных средств, а также некоторые решения других компаний можно применять к любым ПЛИС, хотя это и потребует от разработчика определенных усилий по их адаптации.

FS2

Решение FS2 Logic Navigator может осуществлять внешний или внутренний анализ ПЛИС. Этот так называемый «прибор-на-кристалле» (on-chip instrument, OCI) разработан для синхронизированной логики, а вывод сигналов осуществляется в соответствии с заданной пользователем тактовой плоскостью. Внутренний логический анализатор работает на полной скорости ПЛИС и выводит до 256 сигналов за один такт или до 4096 сигналов в общей сложности. Эти сигналы управляют регистром JTAG TAP. Количество выводов, необходимых для интеграции внутреннего средства Logic Navigator, пропорционально количеству транслируемых сигналов и сложности условий синхронизации.

Частота вывода сигналов на внешний логический анализатор ограничена 200 МГц и 32 сигналами за такт (до 128 сигналов

в общей сложности). Количество выводов ПЛИС для применения внешнего Logic Navigator пропорционально количеству транслируемых сигналов. Схемы синхронизации находятся вне кристалла в самостоятельном аппаратном пробнике, подключаемом к ПК.

Logic Navigator — это синтезированное аппаратно-программное средство, которое позволяет управлять «прибором-на-кристалле» и отображать транслируемые сигналы. Кроме этого, средства автоматического проектирования позволяют автоматизировать настройку, ввод и подключение схемы логики «прибора-на-кристалле» с остальной частью разработки на базе ПЛИС на уровне RTL при программировании с помощью VHDL или Verilog. Средства Logic Navigator могут быть использованы вместе со средствами отладки встроенных процессоров для некоторых процессорных архитектур ПЛИС.

Tektronix

Для ПЛИС Altera компания Tektronix разработала решение, использующее внешние приборы. Логический анализатор используется совместно с программным обеспечением FS2 FPGAView для анализа сигналов, которые извлекаются с помощью встроенного блока интерфейса связи с логическим анализатором (LAI). Altera выпустила этот блок в версии V5.1 программного обеспечения Quartus II, и он обладает функцией мультиплексирования. Этот блок соединяет большой набор внутренних сигналов ПЛИС с небольшим количеством внешних выводов, с которых логический анализатор Tektronix снимает и анализирует сигнал.

В дополнение к возможности минимизировать количество необходимых ресурсов самой ПЛИС, внешний логический анализатор также может использоваться для захвата сигналов с остальной части системы, находящейся за пределами кристалла. В результате складывается более полная картина функционирования устройства.

Один из основных этапов в конфигурировании блока LAI — это выбор тех сигналов, которые в дальнейшем будут анализироваться. Если будет обнаружено, что есть сигналы, не включенные в изначальную настройку блока LAI, но которые требуются в процессе отладки проанализировать, потребуется перенастройка этого блока и перезапуск процесса компиляции. Но так как LAI поддерживает инкрементную компиляцию, то перекомпилируются только сам блок LAI с установлением соединений между новыми сигналами и этим блоком. Это занимает немного времени и не влияет на временные параметры остальной части ПЛИС.

Программный пакет FPGAView позволяет задать нужное количество выводов для анализа, количество банков сигналов, которые необходимо мультиплексировать, частоту тактового генератора (если требуется получить диаграмму состояния), а также состояние выводов

в момент включения питания. Кроме этого, нужно задать параметры физического подключения логического анализатора Tektronix TLA к выводам ПЛИС. Это позволит логическому анализатору автоматически отобразить названия сигналов, которые были им присвоены при разработке ПЛИС.

Agilent Technologies, Xilinx

Средства, входящие в комплект Xilinx ChipScope Pro, предоставляют набор средств отладки и верификации, которые могут быть интегрированы на этапе разработки с использованием языка описания аппаратного обеспечения — Hardware Description Language (HDL) и вставки IP или напрямую в netlist перед началом фазы реализации проекта. Эти ядра позволяют промониторить или передать сигнал на любой внутренний узел ПЛИС, захватить сигналы и сохранить их в ячейках памяти на кристалле ПЛИС, а затем вывести их для анализа с помощью ChipScope Pro.

Средство Agilent Technologies ATC2 для ПЛИС Xilinx является частью средств ChipScope и может работать или в режиме анализа состояний или в режиме анализа временных диаграмм. В режиме анализа состояний захватываются сигналы на частоте внутреннего тактового генератора ПЛИС. В связи с применением конвейерного принципа сигналы передаются на внешние выводы с задержкой в четыре такта, а затем захватываются логическим анализатором Agilent или осциллографом смешанных сигналов MSO. Программное обеспечение FPGA Dynamic Probe (динамический пробник ПЛИС от Agilent), запущенное на любом из этих приборов, управляет ПЛИС.

В другом режиме (асинхронном) сигналы выводятся через мультиплексор ядра ATC2 на внешние выводы. Степень детализации захваченных сигналов в большей степени зависит от полосы пропускания и частоты дискретизации применяемого прибора.

Средство ATC2 обладает встроенной функцией генерации тестового сигнала, которая значительно упрощает задание параметров подключения логического анализатора. Режим автонастройки заставляет ядро сгенерировать уникальную последовательность тестовых сигналов, которые захватываются и распознаются внешним логическим анализатором или осциллографом смешанных сигналов. Далее прибор автоматически подключает внутренние узлы к соответствующим внешним выводам ПЛИС. Таким образом, пользователь может настроить ATC2, указав, какие сигналы надо анализировать и как они объединяются в группы и блоки. Благодаря этому настройка прибора осуществляется автоматически и потому быстро.

Аналогично решению Textronix и Altera, ядро ATC2 и внешний прибор позволяют выполнять одновременный анализ внутренних сигналов ПЛИС и внешних сигналов устройства. Ядро ATC2 позволяет выводить от одного до 128 сигналов на каждую группу сиг-

налов от 1 до 64 и работает как в синхронном, так и асинхронном режиме. Таким образом, можно анализировать до 8192 сигналов при наличии 128 свободных внешних выводов. Кроме этого, возможность мультиплексирования выводов во времени с соотношением 2:1 может удвоить количество выводимых сигналов для каждого вывода ПЛИС. Управление осуществляется через JTAG-порт.

По заверениям компании Agilent Technologies, ядро ATC2 было разработано таким образом, чтобы его можно было добавить с минимальными изменениями процесса проектирования или самого изделия. Главный секрет технологии заключается в уменьшенном размере ядра, использовании таких типов ресурсов ПЛИС, как LUT и триггеры, что является отличным решением как с точки зрения достаточного количества сигналов для анализа, так и с точки зрения минимизации затрат времени.

Synplicity

Решение Identify Intelligent In-Circuit Emulator (ИЦЕ) IP, предлагаемое компанией Synplicity, представляет собой встроенную синхронную систему дискретизации. Сигналы, выбранные пользователем для анализа, захватываются в соответствии с условиями синхронизации, хранящимися в памяти на кристалле, и подаются на выход со скоростью тестового тактового генератора (ТСК) JTAG-порта. Предполагается, что взаимосвязи между частотой дискретизации и частотой тактового генератора (ТСК) JTAG нет. Решение Identify можно использовать для всех типов ПЛИС.

В одном из приложений захват сигналов средством Identify был выполнен на частоте системной шины, умноженной на константу, что обеспечивало лучшую детализацию сигнала. Затем появилась возможность увидеть, что некоторые сигналы, которые должны были быть синхронными, иногда неожиданно изменяли свое состояние в течение одного такта. Добавление дополнительного уровня захвата сигнала к проекту гарантировало, что метастабильное состояние не будет повторяться, и, соответственно, не вызовет некорректных реакций [3].

Брошюры по примерам применения также помогают понять некоторые эффекты использования ИЦЕ, которые могут и не встретиться при использовании тех или иных конкретных приборов. «Средство синтеза оптимизирует комбинаторную логику между двумя регистрами, представляя ее в форме, которая сохраняет непосредственно логическую функцию, но не исходное представление на языке HDL. Затем это средство старается минимизировать схему с учетом заданных ограничений используемой технологии. Сигналы, описанные HDL, которые не связаны напрямую со входами или выходами регистров, могут не оказаться в этой реализации. Это особенно вероятно в случае с ПЛИС, где технологические элементы характеризуются высокой сложностью, например, LUT с четырьмя входами [3].

Процесс синтеза с применением и без применения внешних приборов может отличаться. Из-за увеличения размера добавленных цепей ИЦЕ и большого количества этапов захвата сигналов схема распределения тактового сигнала может стать перегруженной. Это означает, что частоту работы порта JTAG и частоту дискретизации прибора необходимо устанавливать с учетом корректно распределенных иерархий тактового сигнала, а не просто в зависимости ближайшего подходящего по параметрам тактового сигнала.

Решение Identify работает с именами сигналов, прописанных на уровне исходного текста, а не с теми, что прописаны в netlist. Этот момент имеет значение, потому что специалисты работают, в основном, с исходным кодом, и, следовательно, могут более эффективно применять это средство.

Temento Systems

Зачем ограничиваться сравнительно небольшим уровнем производительности только потому, что функция логического анализа встроена в ПЛИС? Компания Temento Systems «сшила» различные IP-модули для создания нового встроенного логического анализатора.

Решение Temento DiaLite предоставляет разработчикам ПЛИС линейку модульных IP-инструментов, которые можно соединить между собой для создания встраиваемого логического анализатора с заданными параметрами. Так, если требуется комбинация последовательного запуска, обнаружение импульсных помех либо встроенный генератор псевдослучайных последовательностей, все эти функции можно включить во встраиваемый логический анализатор. Существует даже модуль логических уравнений, позволяющий создать движок запуска на базе булевых уравнений.

Таким образом, DiaLite позволяет использовать в качестве основы для дискретизации любой тактовый сигнал. Например, находящийся на кристалле PLL можно использовать для создания высокоскоростного тактового сигнала с целью избыточной дискретизации (oversampling) сигналов и обеспечения высокой детализации в режиме асинхронного анализа. Этот подход используется в таких приложениях, как средство наблюдения за шиной VME bus monitor.

Заключение

С развитием ПЛИС растет и количество средств для их программирования. Также как и в случае со схемами с дискретной логикой, возможности аппаратных средств логического анализа помогают быстро выявить источник проблем. Но часто нелегко найти компромисс, учитывающий ограничения по числу выводов ПЛИС, полноту наблюдений, синхронный или асинхронный режим анализа, а также степень использования доступных ресурсов кристалла при встраивании измерительных средств.

Таблица 2

Производитель	Программный пакет	
Actel	Silicon Explorer	
Agilent Technologies	ATC2 Dynamic Probe	
Altera	FPGAView	
Atmel	FPGAs	
Cadence Design Systems	FPGA Design и Cadence Signal Processing Worksystem	
First Silicon Solutions (FS2)	Logic Navigator	
Lattice Semiconductor	LatticeXP FPGA	
QuickLogic	QuickCores	
Synplicity	Identify RTL Debugger	
Tektronix	Simplifying FPGA Debug Application Note	
Temento Systems	DiaLite Tools	
Xilinx	ChipScope Pro	

Теоретически все ПЛИС поддерживают встраиваемые средства отладки, и существует возможность применять программные средства, которые еще больше упрощают отладку ПЛИС. Если вы разрабатываете собственные схемы тестирования, подумайте о плюсах использования интегрированных решений. Это наверняка даст возможность более полного анализа сигналов. Документация, доступная по этим решениям, способна повысить эффективность разработки устройств на базе ПЛИС. К тому же применение универсальных и широко распространенных методов отладки ПЛИС обеспечивает воз-

можность верификации работы систем в полевых условиях. ■

Литература

1. Barr M. Programmable Logic: What's It to Ya? Embedded Systems Programming. June 1999.
2. Using the Identify Hardware Debugger to Catch Timing Problems.
www.synplicity.com/literature/pdf/identify_appnote05.pdf
3. The Impact of Instrumentation Logic.
www.synplicity.com/literature/pdf/debugger_impact.pdf