Тестирование PCI Express с введением ошибок

с целью повышения надежности

Йени ФУ (Yenyi FU)

По мере совершенствования любой технологии возникает необходимость в повышении общей надежности системы — для обеспечения ее работоспособности и устойчивости к внешним воздействиям. Это требование относится и к шине Peripheral Component Interconnect (PCI) Express. По мере того как PCI Express начинает использоваться во все большем числе приложений и растет число устройств, поддерживающих эту шину, следует ожидать от нее большей устойчивости к сбоям и более надежной работы.

Разработанная комитетом PCI-SIG, спецификация для шины PCI Express предусматривает ряд механизмов извещения пользователя о возникновении ошибок. При обнаружении сбоев, в зависимости от их типов, они могут обрабатываться аппаратными средствами или передаваться драйверу либо прикладной программе. Например, прикладная программа может выполнить переключение на работающее устройство, избегая тем самым обращения к устройству, работающему со сбоями. Теоретически, это отличный метод повышения надежности системы в целом. Проблема состоит в том, как сгенерировать такие ошибки, чтобы протестировать реакцию системы на их возникновение. В статье обсуждаются некоторые традиционные методы тестирования с введением ошибок и некоторые инновации в методике испытаний.

Тестирование с введением ошибок

Спецификации PCI-SIG для шины PCI Express предусматривают два механизма оповещения пользователя об ошибках:

- базовое оповещение об ошибках;
- расширенное оповещение об ошибках.

В то время как базовое оповещение об ошибках должно поддерживаться всеми устройствами PCI Express, расширенное оповещение об ошибках не обязательно. Тем не менее, расширенное оповещение об ошибках обладает большими возможностями обработки ошибок, которые могут возникнуть в устройстве. Различные ошибки, возникающие в канале PCI Express, делятся на три основные категории:

- исправимые;
- неисправимые — не фатальные;
- неисправимые — фатальные.

Исправимые ошибки могут обрабатываться аппаратными средствами и не оказывают функционального влияния на работоспособность системы или прикладной программы. В отличие от этого, неисправимые ошибки обрабатываются драйвером устройства или прикладной программой. Каждая прикладная программа и система обрабатывает ошибки по-разному, в зависимости от их типа.

Инженеры, занятые тестированием и проверкой, должны быть уверены в том, что устройство или система известным образом реагирует на возникновение ошибок разных типов, и должны гарантировать отсутствие перебоев в работе или ухудшения характеристик работающих систем. Проблема, стоящая перед инженерами, заключается в создании этих сценариев с ошибками, которые контролируются в лабораторных условиях. В настоящее время на рынке существует много инструментов, помогающих лучше понять поведение устройства при специфических комбинациях ошибок. Эти инструменты делятся на три основные категории:

- комплекты готовых тестов;
- имитаторы;
- генераторы помех.

Далее будут рассмотрены достоинства и преимущества каждой из этих категорий.

Комплекты готовых тестов

Лучшим примером комплекта готовых тестов являются тесты, выполняемые при помощи специальной тестовой платы (PTC), которая используется в лабораториях PCI-SIG для испытаний на совместимость. Комплект готовых тестов содержит, как правило, несколько сценариев тестирования. Например, PTC включает 13 сценариев, которые используются для тестирования функций уровня канала передачи данных и уровня транзакций PCI Express. Основным преимуществом комплекта готовых тестов является простота применения: тесты запускаются несколькими щелчками мыши и выполняются автоматически, при этом для каждого используемого сценария генерируется отчет о результатах.

Одно из ограничений комплектов готовых тестов связано с тем, что они проверяют только с одной стороны канала. Канал PCI Express состоит из конечных точек и комплекса маршрутизации, следовательно, для обеспечения гарантии полной работоспособности инженер должен тестировать обе стороны канала в реальных условиях работы. Более того, ему нужно ввести ошибки в конечные точки и в комплекс маршрутизации, а затем убедиться в правильной реакции обеих сторон на эти ошибки.

В типичном случае плата PTC заменяет материнскую плату и может создавать ошибки лишь в устройствах, расположенных в конечных точках (рис. 1). Это ограничивает применимость таких тестов проверкой только конечных точек. Кроме того, поскольку в реальной системе конечные точки недоступны, драйверы или тестовые программы установить нельзя. В результате комплекты готовых тестов нельзя использовать для тестирования на системном уровне.

Еще одним недостатком комплекта готовых тестов является ограниченный выбор тестов, и их трудно или невозможно изменить. И снова возьмем в качестве примера PTC. Эта плата содержит 13 тестов и охватывает лишь небольшую часть спецификаций. Этого достаточно для проверки операционной совместимости, но недостаточно для надежной проверки устройства в целом. Готовый комплект предлагает обычно фиксированный набор тестов, и конечному пользователю очень сложно расширить число сценариев тестирования в таком комплекте.

Рис. 1. Тестирование конечных точек устройства с помощью PTC

Конечно, на рынке имеются и другие продукты, охватывающие более широкий набор тестов, чем PTC, например, для PCI Express Gen2 имеются комплекты, включающие более 170 тестов. И хотя область тестирования была расширена, этот комплект тестов все-таки страдает от высокой степени статичности и неготовности адаптироваться к изменяющимся требованиям пользователей или к расширению области тестирования.

Имитаторы

Имитаторы — это инструменты, эмулирующие одну сторону канала. Некоторые из них можно запрограммировать так, чтобы они эмулировали конечную точку или комплекс маршрутизации канала, тогда как другие статически настроены на эмуляцию только конечной точки или комплекса маршрутизации (рис. 2а). Общий метод тестирования остается прежним. Для тестирования конечного устройства инженер заменяет комплекс маршрутизации имитатором, эмулирующим этот комплекс маршрутизации (рис. 2б). Для тестирования комплекса маршрутизации конечная точка заменяется имитатором, который затем эмулирует поведение этой конечной точки. Однако, в отличие от реального устройства или комплекса маршрутизации, имитатор можно запрограммировать как на правильную работу, так и на эмуляцию неисправности.

Имитаторы весьма важны на начальном этапе цикла разработки, когда доступно лишь ограниченное число устройств. Их можно использовать для начального функционального тестирования, известного также как первичный этап вывода на рынок системы или устройства, при этом они не зависят от наличия других устройств. Это важно в тех сценариях, где тестируемое устройство является первым устройством такого рода на рынке.

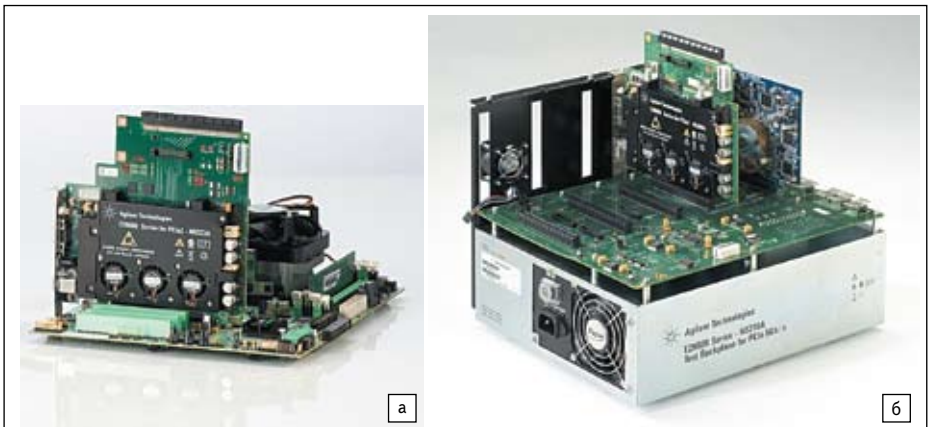

Рис. 2. а) Схема включения имитатора для тестирования комплекса маршрутизации;
б) имитатор с объединительной платой для тестирования конечной точки

Большинство имитаторов можно представить себе как некоторую машину состояний, работающую в соответствии с требованиями протокола и программируемую через графический интерфейс пользователя (GUI) или через интерфейс для программирования приложений (API). Инженер может запрограммировать имитатор через GUI или API так, чтобы он делал почти все, что угодно. В результате получается очень гибкий инструмент, который можно использовать для функционального тестирования, введения ошибок или измерения характеристик.

Но как всегда, за плюсами следуют минусы. Поскольку имитатор очень гибок, его трудней настраивать и контролировать. Например, представьте себе, что генерация ошибки заключается в том, что нужно случайным образом ввести поврежденный пакет уровня транзакций (TLP) после установки соединения и настройки устройства. В нормальной системе эту операцию выполняет комплекс маршрутизации и драйверы. Однако теперь, когда мы используем имитатор для эмуляции комплекса маршрутизации, мы должны воссоздать этот процесс инициации вручную, прежде чем можно будет начать введение ошибки. Эта задача очень громоздка.

Второе ограничение имитаторов тоже лежит в области системного тестирования. Имитатор — отличный инструмент для проверки разрабатываемых устройств или функционального тестирования. Однако в процессе тестирования важно увидеть работу всей системы в целом, включая комплекс маршрутизации, устройства, драйверы и прикладные программы. Один из ключевых вопросов, на который нужно ответить: регистрируется ли ошибка на конечном устройстве? Например, если сетевая карта сообщает о возникновении неисправимой ошибки, кто должен предпринимать соответствующие действия — драйвер или приложение? Используя имитатор, мы устраняем некоторые компоненты системы, и поскольку имитатор эмулирует часть системы, мы не можем проверить поведение системы в целом.

Генераторы помех

Генераторы помех — это третья категория инструментов для тестирования с введением ошибок. Хотя в некоторых отраслях генераторы помех применяются уже давно (например, генераторы радиочастотных помех в оборонных приложениях), в технологии PCI Express они являются сравнительно новой концепцией. Базовая концепция генераторов помех заключается в том, что они устанавливаются между комплексом маршрутизации и конечным устройством и, следовательно, прозрачны для топологии системы. Это значит, что генератор помех не виден ни устройству, ни комплексу маршрутизации.

Основное преимущество генератора помех в том, что он может тестировать всю систему: комплекс маршрутизации, конечное устройство, а также драйверы и прикладные программы. Генератор помех можно запрограммировать так, чтобы он ввел ошибки в комплекс маршрутизации и в конечное устройство. Поскольку при этом работает вся система целиком, инженер может наблюдать реакцию драйвера или программы на конкретную ошибку.

Другое преимущество генератора помех заключается в том, что его настроить проще, чем имитатор. Как упоминалось выше, используя имитатор, инженер вынужден вручную программировать весь процесс инициации. Генератор же прозрачен во время процесса инициации, поэтому последний автоматически выполняется комплексом маршрутизации и конечным устройством. После инициации пользователь может запрограммировать генератор помех на введение ошибок с помощью секвенсора. Секвенсор представляет собой машину состояний, в которой имеются условия ***if/else***. Если такое условие удовлетворяется, то выполняется конкретная операция.

В том же упомянутом выше примере случайное введение поврежденных TLP легко настраивается в генераторе помех. После настройки устройства генератор случайным образом вводит TLP с измененным заголовком (поврежденный TLP).

Поскольку генератор помех не создает каких-либо сообщений или трафика, ему необходимо присутствие обоих устройств на каждом конце канала. Это не всегда возможно на ранних этапах создания новой технологии. Поэтому генератор помех нельзя применять на ранних этапах разработки, в этом случае с задачей лучше справляется имитатор.

Заключение

На современном рынке имеются различные типы инструментов. Выбор соответствующего инструмента определяется многими факторами. Первый и самый важный из них — понимание уровня надежности, который ожидается от исследуемого устройства. При изготовлении сетевой карты для потребительской электроники тестирование всех аспектов технических характеристик и охват всех возможных негативных сценариев может не играть столь важной роли. В этом случае вполне подойдет быстрый и простой в применении комплект тестов с достаточно хорошим охватом (более 13 сценариев тестирования). Однако если инженер разрабатывает серверную платформу для ответственных приложений, очень важно протестировать надежность и способность восстановления всей системы. В таких случаях рекомендуется применять генератор помех, так как он позволяет тестировать совместную работу оборудования, программного обеспечения и драйверов.

Вторым фактором, который надо учитывать, является этап разработки тестируемого устройства: находится ли устройство на начальном этапе проектирования или вам нужно воссоздать инструмент, используемый в лаборатории PCI-SIG для тестирования на совместимость? Для раннего тестирования продукта, например, на этапе разработки, важно упростить рабочую среду. На этом этапе хорошим инструментом может оказаться имитатор, который эмулирует комплекс маршрутизации, обеспечивая полный контроль над тем, что он передает. Однако для тестирования системного уровня, или для создания ошибок в полной системе в лабораториях поддержки потребителей, или для воссоздания проблем, возникших у пользователя, лучшим выбором может оказаться генератор помех.

Универсального решения не существует. Всегда приходится выбирать нужный инструмент для каждой конкретной задачи.