Harris ускоряет верификацию ПЛИС в системах обработки сигналов

“HDL Verifier обеспечил непосредственный интерфейс для ко-симуляции между нашей моделью в MATLAB и логическим симулятором, что позволило нам раньше осуществить верификацию, быстрее идентифицировать проблемы, выполнить больше тестов и сократить весь цикл разработки в целом”.

Jason Plew, Harris Corporation

Инженеры в Harris Corporation привыкли разрабатывать сложные системы обработки сигналов на ПЛИС в кратчайшие сроки. Чтобы соответствовать строгим требованиям заказчиков и их собственным стандартам качества, инженеры должны тщательно верифицировать HDL-описания каждой системы, прежде чем осуществить синтез.

Раньше в процессе HDL-верификации некоторые этапы необходимо было выполнять вручную. Инженеры Harris автоматизировали процесс с помощью инструмента HDL Verifier™, который обеспечил двустороннюю связь между системной моделью в MATLAB® и HDL-проектом, симулируемым в Cadence® Incisive®. Новый процесс устраняет неоднозначность между спецификацией алгоритма и HDL-верификацией, уменьшает объёмы повторной работы и налаживает взаимопонимание между системными инженерами и разработчиками HDL.

“Ко-симуляция в MATLAB и HDL Verifier не только облегчает симуляцию на уровне подсистем, но также позволяет нам более полно верифицировать систему в целом” говорит Jason Plew, старший инженер в Harris. “Мы заметно сократили время, затрачиваемое на разработку экспериментальных стендов для подсистем, что позволило нам быстрее верифицировать и отлаживать наши проекты.”

Задача

Инженерам Harris требовалось разработать и верифицировать систему обработки сигналов, базирующуюся на нескольких ПЛИС Xilinx® Virtex®-5. В этом трудоёмком процессе по завершении симуляции модели подсистемы в арифметике с фиксированной точкой (fixed-point) команда формировала объемные текстовые файлы, содержащие входные данные симуляции и ожидаемые результаты, и импортировала их в Cadence Incisive. Опытный инженер затем писал в HDL тестовые вектора для считывания данных, применения входных воздействий и верификации результатов. В случае выявления

несоответствий, команде требовалось рыться в огромных файлах с результатами, чтобы выяснить, где и почему произошла ошибка симуляции. И при каждом изменении параметров тестирования им приходилось вновь экспортировать данные для симуляции, что означало ещё 8 часов утомительного ручного труда.

“Мы тратили уйму времени, повторяя вручную этапы для каждой подсистемы, и в результате к заявленному сроку успевали выполнить лишь 30% требуемых тестов на системном уровне” вспоминает Plew.

Корпорация Harris нуждалась в возможности вводить входные сигналы в логический симулятор напрямую, чтобы избавиться от необходимости писать тестовые вектора на HDL и работать с огромными текстовыми файлами данных для тестирования.

Решение

Инженеры Harris осуществляли моделирование и симуляцию их систем обработки сигналов в MATLAB. Модель MATLAB являлась универсальным экспериментальным стендом, который они могли использовать интерактивно при ко-симуляции в Cadence Incisive, используя инструмент HDL Verifier.

С помощью MATLAB и DSP System Toolbox™ инженеры разработали в арифметике с плавающей точкой идеальную модель операционного автомата обработки немодулированного сигнала, которую они использовали для верификации проекта на соответствие функциональным требованиям.

После преобразования модели в арифметику с фиксированной точкой, разработчики оптимизировали свои алгоритмы под аппаратную реализацию. В MATLAB осуществлялись дополнительные этапы симуляции, задачей которых было подтвердить, что fixed-point-реализация также удовлетворяет спецификации.

Используя fixed-point-модель в качестве исполняемой спецификации, инженеры Harris затем реализовали систему в HDL.

Разработчики использовали HDL Verifier, чтобы позволить коду MATLAB служить в качестве экспериментального стенда. Интерфейс ко-симуляции позволил использовать MATLAB для формирования входных воздействий и анализа результатов HDL-симуляции.

Чтобы ускорить выполнение множества тестовых заданий, инженеры Harris разработали управляющий скрипт MATLAB, который контролировал параллельные процессы моделирования в вычислительном гриде Linux.

Разработчики использовали данный подход для быстрой верификации как отдельных подсистем, так и цепи обработки сигналов в целом, после чего осуществили синтез и успешно продемонстрировали работающее устройство. Инженеры Harris начали применять подобный подход на других проектах, в том числе и на проекте, требующем реализации быстрого преобразования Фурье с миллионом точек.

Результаты

- Время выполнения функциональной верификации сократилось на 85%. “Раньше опытному инженеру требовалось 8 часов для подготовки экспериментального стенда подсистемы,” говорит Plew. “Теперь с HDL Verifier мы можем создавать стенды за 1 час, и поскольку MATLAB генерирует входные воздействия и осуществляет анализ, все результаты тестирования также доступны в среде MATLAB для постобработки.”
- Выполнено 100% всех запланированных тестов. “Раньше процесс был настолько сложным, что мы зачастую осуществляли очень мало верификации на уровне подсистем, и наш график не позволял нам выполнять более 30% требуемых тестов на системном уровне,” говорит Plew. “С инструментами MathWorks мы провели генерацию и симуляцию 100% требуемых тестов, используя автоматизированный процесс, который позволил нам обнаруживать большинство ошибок на уровне подсистем.”
- Реализованный проект не содержит ошибок. “Мы сэкономили недели отладки в лаборатории, потому что так тщательно верифицировали проект с использованием HDL Verifier для ко-симуляции HDL-кода в MATLAB и Cadence Incisive,” замечает Plews. “Фактически, элементы операционного автомата ПЛИС функционировали правильно с самого начала”