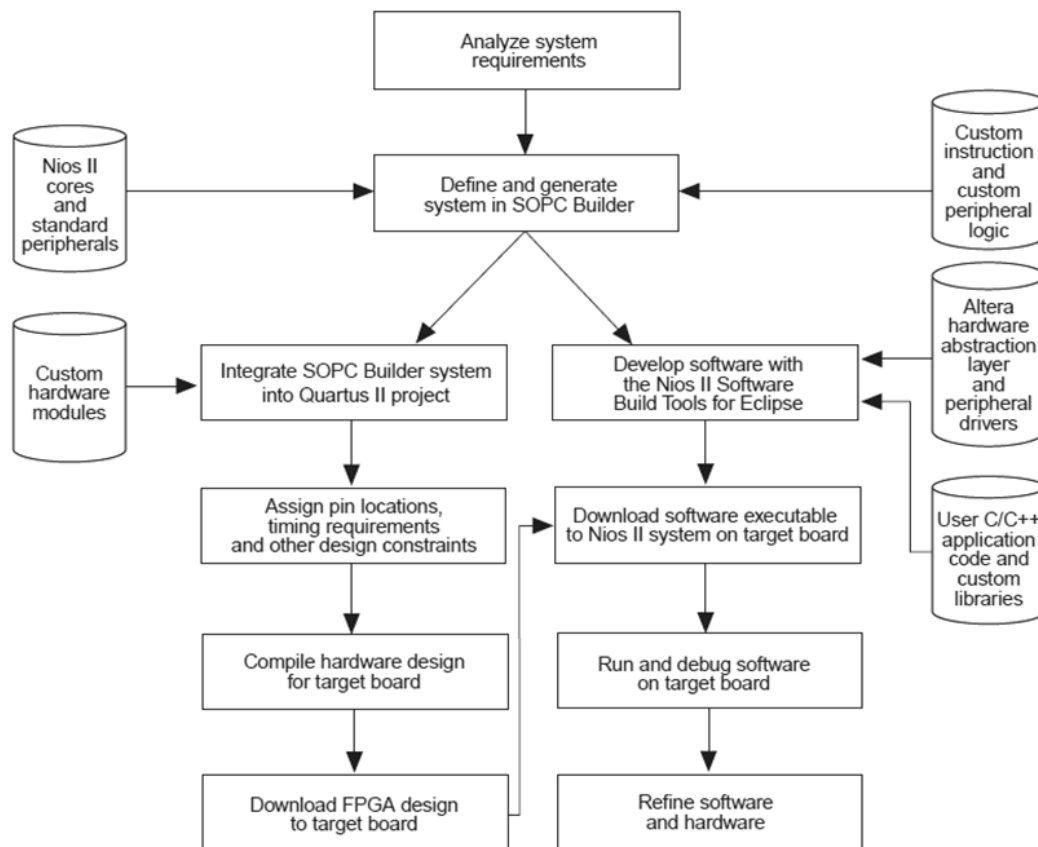


Процесс разработки системы Nios II

В этой секции описывается полный процесс разработки для создания системы Nios II и прототипирования её на выбранной плате. На рисунке 1-2 показан процесс разработки системы Nios II.

Figure 1–2. Nios II System Development Flow



Процесс разработки системы Nios II состоит из трёх типов разработки: части аппаратных средств, части программных средств и части системных средств, включающих в себя аппаратные и программные средства. Для простой системы Nios I, один человек может выполнить все части. Для большинства комплексных систем, необходимо разделение разработчиков аппаратных и программных частей. Часть проектирования системы, состоящая из аппаратных и программных средств, потребует участия обеих групп разработчиков. В этом случае, отдельные группы разработчиков аппаратных и программных средств должны знать о том, какие файлы и информация должна быть представлена группами на точке объединения процесса разработки.

Особое внимание в этом учебном пособии уделено части разработки аппаратных средств, и дано только небольшое введение, касательно разработки программных средств.

После завершения этого учебного пособия обратитесь к Настольной книге разработчиков программного обеспечения под Nios II, особенно к учебному пособию "Начало работы с графическим интерфейсом пользователя" за последующими пояснениями процесса разработки программного обеспечения. Настольная книга – это полное пособие для разработчиков программ под процессор Nios II.

Анализ системных требований

Процесс разработки начинается с рассмотрения проекта как такового, включая анализ требований к приложению, отвечающий на следующие вопросы:

- Какие вычислительные характеристики требуются в приложениях?
- Какая полоса пропускания или производительность требуется в приложениях?

-
- Такие типы интерфейсов требуются в приложениях?
 - Требуется ли многопоточное программное обеспечение в приложениях?

Основываясь на ответы на эти вопросы, вы можете конкретизировать системные требования, такие как:

- Какое ядро процессора Nios II использовать: малое или быстрое.
- Какие компоненты понадобятся и сколько конкретно каждого типа.
- Какую операционную систему реального времени (RTOS), если нужно.
- Какая логика ускорения аппаратных средств может существенно улучшить характеристики системы. Например:
 - Может ли компонент DMA устранить пустые процессорные циклы копирования данных?
 - Могут ли специальные инструкции переместить критические петли DSP алгоритма?

Для анализа этих требований необходимо задействовать группы разработчиков аппаратных и программных средств.

Определение и генерирование системы в SOPC Builder

После анализа требований к аппаратной части системы, вы используете SOPC Builder для определения ядра (ядер) процессора Nios II, памяти и прочих компонентов, необходимых в вашей системе. SOPC Builder автоматически генерирует логику внутренних соединений для интегрирования компонентов в аппаратную часть системы.

Вы можете выбрать из списка стандартных ядер процессора и компонентов, предлагаемых в Nios II EDS. Вы можете также добавить свои собственные аппаратные компоненты для ускорения характеристики системы. Вы можете добавить логику собственных инструкций для ядра Nios II, которая улучшает характеристики CPU, или добавить собственный компонент, разгружающий CPU от задач. Это учебное пособие охватывает задачи установки стандартного процессора и компонентов, и не охватывает задачи установки собственной логики в систему.

Главной задачей SOPC Builder является получения следующих типов файлов:

- Файл проекта SOPC Builder (**.sopc**) — Содержит аппаратную часть системы SOPC Builder.
- Файл информации SOPC Builder (**.sopcinfo**) — Содержит удобочитаемое описание содержимого файла **.sopc**. Nios II EDS использует файл **.sopcinfo** для компиляции программного обеспечения для выбранного устройства.
- Файлы языка описания аппаратуры (HDL) — Все файлы проектирования аппаратных средств, описывающих систему SOPC Builder. Программа Quartus II использует HDL файлы для компиляции всего проекта FPGA в выходной файл SRAM (**.sof**).

За подробной информацией о ядрах процессора Nios II, обратитесь к Настольной книге процессора Nios II. За подробной информацией о SOPC Builder и разработке собственных компонентов, обратитесь к тому 4: SOPC Builder в Настольной книге Quartus II. За подробной информацией о собственных инструкциях, обратитесь к Руководству пользователя собственных инструкций Nios II.

Интегрирование системы SOPC Builder в проект Quartus II

После генерирования системы Nios II с использованием SOPC Builder, вы интегрируете его в проект Quartus II. Используя программу Quartus II, вы выполняете все задачи, необходимые для создания конечного аппаратного проекта FPGA.

Как показано на рисунке 1-1 на странице 1-2, большинство проектов FPGA имеют логику вокруг системы Nios II. Вы можете интегрировать свои собственные аппаратные модули в проект FPGA или другие готовые модули проекта, защищённые авторским правом (IP), поставляемые Altera или другими IP разработчиками. Это учебное пособие не рассматривает добавление прочей логики вокруг системы Nios II.

Используя программу Quartus II, вы можете также назначить размещение выводов для I/O сигналов, определить временные ограничения и прочие ограничения для проекта. В завершении, вы компилируете проект Quartus II, чтобы получить **.sof** файл для конфигурирования FPGA.

Вы загружаете **.sof** файл в FPGA на выбранной плате с помощью загрузочного кабеля, например USB-Blaster™. После конфигурирования, FPGA преобразуется в специальный аппаратный проект, в нашем случае это система с процессором Nios II.

За дополнительной информацией об использовании программы Quartus II, обратитесь к "Введению в программу Quartus II", к "Настольной книге Quartus II" и к "Интерактивному учебному пособию по программе Quartus II" в секции учебных курсов на веб-сайте Altera.

Разработка программного обеспечения с помощью инструмента создания программы Nios II под Eclipse

Используя инструмент создания программы Nios II под Eclipse, вы выполните все задачи разработки программного обеспечения для вашей системы с процессором Nios II. После того, как вы сгенерируете систему с помощью SOPC Builder, вы можете сразу же начать разработку вашего C/C++ программного кода с помощью инструмента создания программы Nios II под Eclipse. Altera предоставляет драйверы компонентов и уровень аппаратных абстракций (HAL), позволяющие вам быстро писать программы для Nios II, не зависимо от низкоуровневых особенностей аппаратных средств. В дополнении в вашему программному коду, вы можете разрабатывать и использовать собственные библиотеки в ваших проектах с помощью инструмента создания программы Nios II под Eclipse.

Для создания нового проекта приложения Nios II C/C++, инструмент создания программы Nios II под Eclipse использует информацию из **.sopcinfo** файла. Вам также необходим **.sof** файл для конфигурирования FPGA перед запуском и отладкой прикладного проекта в выбранном устройстве.

Инструмент создания программы Nios II под Eclipse может создать несколько выходных файлов, описанных ниже. Не всем проектам требуется всё перечисленное ниже:

- Файл **system.h** – определяет идентификатор для обращения к аппаратной части системы. Инструмент создания программы Nios II под Eclipse автоматически создаёт этот файл, когда вы создаёте новый проект.
- Файл формата исполнения и связи (**.elf**) — это результат компиляции проекта приложения C/C++, который вы загружаете прямо в процессор Nios II.
- Шестнадцатеричный файл (формат Intel) (**.hex**) — содержит информацию об инициализации памяти на чипе. Инструмент создания программы Nios II под Eclipse

генерирует эти файлы инициализации для памяти на чипе, поддерживающей инициализацию содержимого.

■ Данные для программирования флеш-памяти – это код самозагрузки и другие произвольные данные, которые вы должны загрузить во флеш-память. Инструмент создания программы Nios II под Eclipse имеет флеш-программатор, который позволяет вам записывать вашу программу во флеш-память. Флеш-программатор добавляет соответствующий загрузочный код, позволяя вам запрограммировать загрузку из флеш-памяти. Вы можете также использовать флеш-программатор для записи нужных данных во флеш-память.

Это учебное пособие сосредоточено на загрузке только **.elf** прямо в систему Nios II. Полная информация от разработке программного обеспечения для процессора Nios II, обратитесь к Настольной книге разработчиков программного обеспечения под Nios II.

Запуск и отладка программы в выбранной плате

Инструмент создания программы Nios II под Eclipse предоставляет полный набор средств для загрузки программы в выбранную плату, запуска или отладки программы в устройстве. Инструмент создания программы Nios II под Eclipse позволяет вам запускать и останавливать процессор, пошагово исполнять код, устанавливать точки останова и анализировать переменные по ходу выполнения программы.

Дополнительная информация о запуске и отладке программ Nios II, находится в учебном пособии "Начало работы с графическим интерфейсом пользователя" – главе в Настольной книге разработчиков программного обеспечения под Nios II.

Варьирование процессом разработки

Процесс разработки не строго линеен. В этой главе описаны обычные вариации.

Уточнение программных и аппаратных средств

После запуска программы в выбранной плате, вы обнаружили, что системе Nios II потребовались лучшие характеристики. В этом случае, вы можете повторить часть разработки программных средств, чтобы улучшить алгоритм работы программы. С другой стороны, вы можете повторить часть разработки аппаратных средств, чтобы добавить ускоряющую логику. Если система выполняет взаимно независимые задачи, вы должны разделить их на два (и более) процессора Nios II, которые разделяют рабочую нагрузку и улучшают характеристики каждого отдельного процессора.

Итерационное создание системы Nios II

Большинство технологий создания комплексных систем Nios II начинают с простой системы SOPC Builder, затем итерационно прибавляют к ней. На каждой итерации, вы можете проверить правильность функционирования системы. Вы можете выбрать верификацию основных компонентов системы: процессора, памяти и каналов коммутации, прежде чем добавлять прочие комплексные компоненты. Когда разрабатывается собственный компонент или собственная инструкция, то сначала нужно интегрировать собственную логику в минимальную систему, чтобы проверить её функционирование, а затем вы можете интегрировать собственную логику в более комплексную систему.

Altera предлагает несколько базовых рабочих проектов Nios II, которые вы можете использовать в качестве начальной точки для своего собственного проекта. После инсталляции Nios II EDS, обратитесь к директории *<Nios II EDS путь установки>/examples/verilog* или к *<Nios II EDS путь установки>/examples/vhdl*. Демонстрация приложений также доступна при инсталляции в новых наборах разработчика.

Верификация системы с помощью инструментов симуляции аппаратных средств

Вы можете выполнить симуляцию аппаратных средств или исполнение программы в системе Nios II, используя инструменты, такие как RTL симулятор ModelSim®. Аппаратная симуляция прекрасно подходит для получения точных требований, включая следующие случаи:

- Для верификации точных циклических характеристик системы Nios II прежде, чем будет доступна аппаратная часть.
- Для верификации функционирования собственных компонентов или собственных инструкций Nios II прежде, чем испытывать их в аппаратной части.

Часть с аппаратной симуляцией не показана на рисунке 1-2 на странице 1-4. Если вы создали систему Nios II, основываясь на стандартных компонентах, предлагаемых в Nios II EDS, то самый простой путь для функциональной верификации – это загрузка аппаратной и программной частей непосредственно в плату разработки.

За подробной информацией о выполнении аппаратной симуляции системы Nios II, обратитесь к *AN351: Симуляция проектов со встроенным процессором Nios II*.