



Debugging Nios II Systems with the SignalTap II Embedded Logic Analyzer

AN-446-2.0

Application Note

Этот документ научит вас отлаживать ваш проект системы, используя динамическую информацию, появляющуюся во время работы процессора Nios[®] II. В коротком учебном пособии показано, как использовать плагин Nios II, встроенный логический анализатор SignalTap[®] II и инструмент создания программы под Nios II на Eclipse для защёлкивания, захвата и отвода сигналов, которые включают светодиоды.

Введение

Так как проекты системы на FPGA становятся очень сложными, а сами системы сосредотачивают в себе большое количество процессоров, периферии, шин и мостов, - разработчикам требуется лучший, более современный и сложный инструмент отладки на уровне системы. Встроенный логический анализатор Altera SignalTap II предлагает возможность аппаратной отладки в реальном времени, встроив логический анализатор в систему. Плагин Nios II для встроенного логического анализатора SignalTap II расширяет возможности отладки системы в SignalTap II, разрешая захват исполнения программы процессором Nios II.

Плагин Nios II расширяет возможности отладки системы в SignalTap II, разрешая вам запросто защёлкивать и захватывать инструкции, а также отводить данные, исполняемые в ядре Nios II. Вы можете задать триггер отведения инструкции, который позволяет защёлкивать встроенный логический анализатор SignalTap II, когда процессор обращается по определённому адресу, введя определённое имя символа из своей программы, или задав собственное состояние триггера. Плагин Nios II автоматически коррелирует отведение процессора с помощью заданного образа программы, предоставляя вам имя символа или офсет для отображения отведения вместе с декодированным с машинного языка Nios II операционными кодами (opcodes).

Необходимо условие

Задача этого документа - это обучить вас работе с плагином Nios II в среде SignalTap II. Этот документ подразумевает, что вы знакомы с основными программными инструментами Altera[®], включая основы владения встроенным логическим анализатором SignalTap II.

Этот документ подразумевает, что вы знакомы с работой в программе Quartus[®] II, с Qsys и со встраиваемым процессором Nios II, а также с основными возможностями и средствами встроенного логического анализатора SignalTap II. Поскольку плагин Nios II является расширением логического анализатора SignalTap II, основы использования плагина идентичны работе в логическом анализаторе SignalTap II.

В дополнении ко всему, учебное пособие этого документа подразумевает наличие у вас доступа к основным инструментам Altera и интеллектуальных ядер (IP).

За дополнительной информацией об опциях конфигурации SignalTap II и пользовательских моделях обратитесь к главе "[Отладка проекта с помощью встроенного логического анализатора SignalTap II](#)" в томе 3 настольной книги Quartus II.

Требования к системе

Использование плагина Nios II требует следующие инструменты:

- программа Quartus II версии 11.0 и старше,
- пакет разработчика Nios II EDS версии 11.0 и старше,

Выполнение учебного пособия в конце этого документа потребует наличия следующих инструментов:

- набор разработчика Nios II Kit,
- кабель USB-Blaster™,
- файлы **an446_signal_tap_test.zip** (которые [можно загрузить](#) вместе с этим документом)

За более подробной информацией об аппаратных и программных требованиях для этого учебного пособия, обратитесь к секции: "Учебное пособие: Использование плагина Nios II" на странице 10.

Плагин Nios II

Плагин Nios II - это отладочное дополнение для встроенного логического анализатора SignalTap II. Он разрешает вам захватывать операционные коды (opcodes), исполняемые встроенным процессором Nios II. Плагин Nios II работает после установки отладочного узла внутри АЛУ процессора Nios II. Плагин Nios II поддерживает все варианты реализации ядра Nios II.

Каждый инсталлированный плагин Nios II ассоциируется с заданным процессором Nios II, и может работать с другими инсталлированными плагинами Nios II и другими модулями SignalTap II.

Вы можете задавать следующие типы состояний триггера:

- Адрес инструкции
- Имя символа (функции), представленной в исполняемом или загружаемом формате файла Nios II (.elf)
- Имя символа плюс офсет

Плагин декодирует все инструкции в удобочитаемый формат. Адреса инструкций транслируются в имя символа плюс офсет (если это возможно), а операционные коды инструкций транслируются в эквивалентную им мнемонику ассемблера.

Установки в проекте Quartus II и в логическом анализаторе SignalTap II

Вы должны создать и сконфигурировать SignalTap II файл (.stp) в вашем Quartus II проекте для использования плагина Nios II. Для добавления .stp файла в вашу систему, выполните следующие пункты:

1. В меню **File** программы Quartus II кликните **New**.
2. В диалоговом окне **New** в категории **Verification/Debugging Files** кликните **SignalTap II Logic Analyzer File**.
3. Кликните **OK**.

Или вы можете создать новый или открыть существующий .stp, выполнив следующие пункты:

- В меню **Tools** программы Quartus II кликните **SignalTap II Logic Analyzer**.

После того, как вы сделаете этот шаг, откроется окно SignalTap II.

Плагин Nios II пока не поддерживается проектом Quartus II, если в нём разрешена опция инкрементной компиляции. Поэтому вы должны запретить инкрементную компиляцию в вашем проекте Quartus II. Для запрещения инкрементной компиляции выполните следующие шаги:

1. В меню **Assignments** программы Quartus II кликните **Settings**.
2. В диалоговом окне **Settings** в категории **Compilation Process Settings** кликните **Incremental Compilation**.
3. В **Incremental Compilation** выберите **Off**.
4. Кликните **OK**.

Добавление плагина Nios II

Когда вы добавляете плагин Nios II в вашу систему, вы должны задать ему процессор Nios II, который он будет отслеживать, и, опционально, образ программы процессора (в виде .elf файла). Плагин Nios II обрабатывает .elf файл для извлечения символьной информации. Плагин использует .elf файл во время конфигурирования для задания позиций триггера.

Для добавления плагина Nios II в вашу систему, выполните следующие пункты:

1. В меню **Processing** программы Quartus II выберите **Start** кликните **Start Analysis & Elaboration**.
2. В окне SignalTap II кликните правой клавишей на список узлов SignalTap II. Выберите **Add Nodes with Plug-In** и кликните **Nios II**. Раскроется диалог **Select Hierarchy Level**.
3. В диалоге **Select Hierarchy Level** выберите модуль процессора Nios II, за которым вы хотите наблюдать с помощью плагина.
4. Кликните **OK**. Раскроется диалог **Plug-In Options**.
5. В диалоге **Plug-In Options** опционально задайте путь к .elf файлу.
6. Кликните **OK**.

В любое время вы сможете изменить используемый плагином Nios II **.elf** файл, выполнив следующие пункты:

1. В окне SignalTap II кликните на вкладку **Setup**.
2. В списке узлов SignalTap II правым кликом на модуль плагина Nios II, который вы хотите изменить, затем кликните **Plug-In Options**. Раскроются опции конфигурирования плагина Nios II.

Nios II SBT на Eclipse генерирует файл во время процесса сборки программы. Обычно **.elf** файл вашего проекта расположен в директории **software/app/**, которую создаёт Nios II SBT на Eclipse. Также вы можете создавать **.elf** файл в процессе Nios II SBT.

Определение состояний триггера

В отличие от стандартных состояний триггера в логическом анализаторе SignalTap II, которые описываются как аппаратные или логические события, состояния триггера в плагине Nios II определяются адресом инструкции. Плагин Nios II защёлкивает значение, когда процессор Nios II обращается по определённом адресу инструкции во время исполнения программы.

Базовый триггер

В режиме базового триггера, плагин Nios II использует для защёлкивания видимый процессору системный адрес, от которого отводятся значения. Для задания триггера, кликните на столбец **Trigger Conditions** в плагине Nios II и введите адрес инструкции.

Вы можете ввести только следующие поддерживаемые состояния триггера:

- целое шестнадцатеричное число: `0x<32-битное число>` (например, `0x20000000`),
- символ **.elf** файла: `<строка>` - символьное имя C/C++ функции, которая выводится символом в **.elf** файле (например, `foo`),
- символ **.elf** файла + офсет: `<строка> + <целое шестнадцатеричное число>` (например, `foo+0x80`)

Для задания состояний триггера с опцией имени символа, вы должны сконфигурировать плагин Nios II **.elf** файлом процессора Nios II. Файл **.elf** может не иметь заданного имени символа, поэтому появляется сообщение об ошибке, а поле состояния триггера задаётся неопределённым значением. Когда вы вводите значение в поле состояния триггера, плагин Nios II исследует содержимое соответствующего **.elf** файла на предмет изменений. Такая проверка гарантирует, что состояние триггера будет синхронизировано с содержимым **.elf** файла, даже если **.elf** файл часто меняется, например, во время отладки программы.

Для задания символов в **.elf** файле используйте следующий метод генерации **objdump** файла:

- Включите опцию **CREATE_OBJDUMP** в **makefile**, затем соберите проект.
- После того, как вы создадите **.elf** файл, в командной среде Nios II, введите следующую команду:

```
nios2-elf-objdump -s <filename>.elf ↵
```

За дополнительной информацией обратитесь к секции "Сборка программы Nios II" на стр. 11.

Вы можете задавать стандартные состояния триггера логического анализатора SignalTap II в окне SignalTap II, правым кликом на столбец **Trigger Conditions** и выбрав заполнение триггера. Однако для ясности, вам потребуется вводить ваши состояния триггера в плагине Nios II.

Не заданный адрес состояния триггера для плагина Nios II - это одно и то же, что и не заданное состояние триггера в обычном элементе SignalTap II. Если вы не задаёте состояние триггера, логический анализатор защёлкивается постоянно, а отображаемые данные - бесполезны.

Несколько триггеров

Плагин Nios II поддерживает средство нескольких триггеров SignalTap II, которые являются частью более сложной процедуры захвата. Вы можете использовать последовательность триггеров плагина Nios II как часть любого состояния триггера SignalTap II.

Расширенные триггеры

Вы можете создавать сложные триггеры с опцией SignalTap II Advanced Trigger. Однако плагин Nios II теряет большинство своих свойств при использовании этой опции. Группы сигналов из плагина Nios II находятся в редакторе состояний расширенного триггера (Advanced Trigger Configuration Editor), где они отображаются как обычные сигналы. Редактор состояния расширенного триггера теряет свойства плагина Nios II захватывать по адресу или символу.

Триггеры включения питания

Вы можете использовать плагин Nios II вместе со средством триггера включения SignalTap II. Поскольку триггеры включения разрешены до ручного запуска логического анализатора SignalTap II, они прекрасно подходят для мониторинга системы с работающим процессором Nios II в режиме самозагрузки, то есть после конфигурирования FPGA. В этом случае процессор Nios II запускает исполнение программы прямо из системной памяти без помощи отладчика (запуск, остановка и загрузка памяти).

Назначение такта захвата

Вы должны задать тактовый сигнал для контроля над захватываемыми отсчётами. Задайте тактовый сигнал на вкладке Signal Configuration в окне SignalTap II. Altera рекомендует выбирать тактовый сигнал, который использует процессор Nios II, в качестве такта захвата SignalTap II. Использование такта процессора Nios II гарантирует вам, что вы будете захватывать инструкции в соответствии с их исполнением в процессоре Nios II.

Выбор глубины отсчётов, типа памяти и режима буфера захвата

Для плагина Nios II, как и для логического анализатора SignalTap II, вы должны сконфигурировать глубину отсчётов, тип памяти и режим буфера захвата для текущей сессии сбора данных. Эти настройки такие же, как и в обычной сессии сбора данных логического анализатора SignalTap II, они доступны на панели конфигурации сигнала (Signal Configuration).

Существенное значение имеет выбор глубины отсчётов (**Sample Depth**). Плагину Nios II для каждого отсчёта требуется захватывать несколько сигналов, что быстро сокращает доступные ресурсы памяти. Используйте встроенный в SignalTap II счётчик ресурсов (resource estimator) для лучшего понимания того, как влияет параметр глубины отсчётов на ваш проект.

Компиляция проекта и программирование выбранного чипа

Плагин Nios II компилируется в проекте Quartus II вместе с логическим анализатором SignalTap II. Однако, поскольку плагин Nios II пока не поддерживает инкрементную компиляцию Quartus II, вы должны выполнять полную компиляцию проекта после добавления плагина.

После компиляции вы сможете запрограммировать чип FPGA объектным файлом SRAM (.sof) в окне SignalTap II точно так же, как если бы вы не использовали плагин Nios II.