

Программирование флеш-памяти

Многие процессорные системы Nios II используют внешнюю флеш-память для хранения одного или нескольких следующих объектов:

- Программного кода
- Программных данных
- Данных конфигурации FPGA
- Файловой системы

Nios II SBT на Eclipse предлагает утилиты флеш программатора, чтобы помочь вам управлять и программировать содержимым флеш-памяти. Флеш программатор позволяет вам программировать любую комбинацию программы, аппаратной части или двоичных данных в флеш-память за одну операцию.

Запуск флеш программатора

Вы запускаете флеш программатор, кликнув на **Flash Programmer** в меню Nios II. Когда вы впервые открываете флеш программатор, никакие средства контроля не доступны, пока вы не откроете или не создадите файл настроек флеш программатора (**.flash-settings**).

Создание файла настроек флеш программатора

Файл **.flash-settings** описывает ваши установки в графической оболочке флеш программатора, для программирования флеш-памяти. Эта информация включает в себя файлы, необходимые для программирования во флеш-память, описание конфигурации аппаратной части в системе SOPC Builder, и местоположение программ. Вы должны создать или открыть файл настроек флеш программатора, прежде чем сможете запрограммировать флеш-память.

Вы создаёте файл настроек флеш программатора в меню File. Когда вы кликаете **New**, открывается диалог **New Flash Programmer Settings File**.

Спецификация аппаратной конфигурации

Вы задаёте аппаратную конфигурацию, открывая файл **.sopcinfo**. Вы можете локализовать размещение **.sopcinfo** файла двумя способами:

- Указав на файл настроек BSP. Флеш программатор находит **.sopcinfo** файл, ассоциированный с этим BSP.
- Указав прямо на **.sopcinfo** файл.

После того, как вы идентифицируете аппаратную конфигурацию, подробная информация о выбранной аппаратной реализации появляется в верхней части экрана флеш программатора Nios II.

Также в верней части экрана флеш программатора Nios II находится кнопка **Hardware Connections**, открывающая диалог **Hardware Connections**. В этом диалоге вы выбираете загрузочный кабель и контролируете поведения ID системы, как описано в секции "Вкладка Target Connection" на странице 2-19.

Вид экрана флеш программатора

Экран флеш программатора делится на две части. Верхняя часть – область команд, а нижняя – область консоли. Подробное описание областей экрана флеш программатора описывается в этой секции.

Внизу области консоли находится кнопка **Start**. Эта кнопка разрешена, если параметры флеш программатора верные. С её помощью запускается процесс программирования флеш-памяти.

Область команд

В области команд вы задаёте настройки и прочие параметры, определяющие файл настроек флеш программатора. Область команд имеет одну или несколько вкладок. Каждая вкладка описывает компонент флеш-памяти, доступный в аппаратной части. Каждая вкладка позволяет вам видеть параметры компонента памяти, а также видеть и редактировать список файлов, которые будут запрограммированы в компоненте памяти.

Кнопки **Add** и **Remove** позволяют вам создавать и редактировать список файлов, которые будут запрограммированы в компоненте памяти.

Вкладка **File generation command** показывает команды, используемые для генерации файлов Motorola S-record (**.flash**), которые используются для программирования флеш-памяти.

Вкладка **File programming command** показывает команды, используемые для программирования **.flash** файлов в флеш-память.

Кнопка **Properties** открывает диалог **Properties**, который позволяет вам видеть и модифицировать информацию о каждом конкретном файле. В случае с **.elf**, кнопка **Properties** обращается к адресу сброса проекта, к основному адресу флеш-памяти и конечному адресу, а также к файлу загрузки операционной системы (если имеется).

Флеш программатор определяет необходимость загрузки операционной системы, основываясь на размещении загрузки и запуска в секции **.text**. Вы можете использовать диалог **Properties**, чтобы переписать конфигурацию по умолчанию для загрузчика операционной системы.

Область консоли

Область консоли показывает результаты настроек и команд, которые вы выбираете в области команд. Область консоли состоит из следующих вкладок:

- Вкладка **Information** (информация)
- Вкладка **Problems** (проблемы)
- Вкладка **Processing** (процесс)

Вкладка Information

Вкладка **Information** показывает высокоуровневые изменения, которые вы делаете с вашим файлом настройки программатора.

Вкладка Problems

Вкладка **Problems** показывает сообщения о предупреждениях и ошибках во время процесса создания файла настроек программатора.

Вкладка Processing

Когда вы программируете флеш-память, на вкладке **Processing** отображаются отдельные действия процесса программирования.

Сохранение файла настроек флеш программатора

Когда вы закончите конфигурировать входные файлы, размещение и прочие настройки для программирования вашего проекта во флеш-память, вы можете сохранить настройки в **.flash-settings** файле. С помощью **.flash-settings** файла вы можете повторно запрограммировать проект без обновления конфигурации настроек. Сохранение **.flash-settings** файла в меню File.

Опции флеш программатора

В меню Options вы можете контролировать некоторые глобальные аспекты поведения флеш программатора, что описывается в этой секции.

Подробнее об этих средствах в "Руководстве пользователя флеш программатором Nios II".

Staging Directories

В диалоге **Staging Directories** (каскадирование директорий) вы контролируете, где флеш программатор создаёт свои скрипты и **.flash-settings** файлы.

Generate Files

Если вы запрещаете эту опцию, флеш программатор не может генерировать программные файлы, но программные файлы находятся в директории. Вы должны использовать это средство для перепрограммирования набора недавно созданных файлов.

Program Files

Если вы запрещаете эту опцию, флеш программатор генерирует программные файлы и скрипты, но не программирует флеш-память. Вы можете позже использовать эти файлы для программирования флеш-памяти, предварительно выключив опцию **Generate Files**.

Erase Flash Before Programming

Когда разрешено, флеш-память стирается перед программированием.

Run From Reset After Programming

Когда разрешено, процессор Nios II сбрасывается и запускается после программирования флеш-памяти.

Создание файла внутренней инициализации загрузочной памяти

Nios II SBT на Eclipse может создавать шестнадцатеричный (Intel-Format) файл (**.hex**), требуемый для загрузки Nios II напрямую из внутренней памяти (без внешней загрузки). Для создания **.hex** файла вашего проекта, правым кликом на проект приложения, выберите **Make Targets**, и кликните **Build**, чтобы открыть диалог **Make Targets**. Выберите и соберите задание **mem_init_install**.

Задание **mem_init_install** создаёт файл, называемый *<имя компонента памяти>.hex*, и размещает его в директории вашего проекта Quartus. После создания **.hex** файла, перекомпилируйте ваш проект Quartus.

За подробной информацией о создании файла инициализации памяти, обратитесь к секции "Общие задачи BSP" в главе "Инструменты создания программы Nios II" в "Настольной книге программиста Nios II".

Управление Toolchains в Eclipse

Nios II EDS содержит две версии GNU коллекций компилятора (GCC) toolchain (цепи инструментов): GCC 3.4.6 и GCC 4.1.2. Начиная с версии 10.0, EDS использует GCC 4 для всех новых SBT проектов. EDS использует GCC 3 по умолчанию только для проектов, созданных до версии 10.0, и только для проектов, конвертированных в SBT из Nios II IDE.

В большинстве случаев, вы можете спокойно обновить проекты с GCC 3 на GCC 4. Для изменения версии GCC toolchain в Eclipse, правым кликом на проект выберите **Properties**. В диалоге **Properties** разверните вкладку **C/C++ Build** и выберите **Tool Chain Editor**. Выберите соответствующую Nios II GCC toolchain, в зависимости от операционной системы хост-машины. Например, для использования GCC 4, выберите одну из следующих toolchains:

- В операционной системе Windows выберите MinGW Nios II GCC 4.
- В операционной системе Linux выберите Linux Nios II GCC 4.

После того, как вы выберите toolchain, SBT на Eclipse продолжит использовать эту toolchain в вашем проекте, пока вы снова его не измените.

Если вы переносите проект на другую хост-платформу, вы должны вручную изменить соответствующую toolchain под новую хост-платформу. Например, если вы переносите GCC 4 с хоста Windows на хост Linux, используйте диалог **Properties** для выбора Linux Nios II GCC 4.

Основная информация о GCC toolchains находится в секции "Поставляемые Altera инструменты разработки" в главе "Инструменты создания программы Nios II" в "Настольной книге программиста Nios II". За информацией о выборе toolchain в командной строке, обратитесь к главе "Начало работы в командной строке" в "Настольной книге программиста Nios II".

Режим использования Eclipse

Поведение основной программы Eclipse и средств CDT изменяется Nios II SBT на Eclipse. Если вы попытаетесь использовать эти средства тем же способом, что и для не Nios II проектов, у вас могут возникнуть проблемы с конфигурированием или сборкой вашего проекта. В этой секции обсуждаются названные средства.

Конфигурирование свойств приложения и библиотеки

Для конфигурирования свойств проекта специально для приложения Nios II SBT и проектов библиотек, используйте вкладки **Nios II Application Properties** и **Nios II Library Properties** диалога **Properties**. Для открытия соответствующей вкладки, правым кликом на приложение или библиотеку, затем кликните **Properties**. В зависимости от типа проекта, откроются вкладки **Nios II Application Properties** или **Nios II Library Properties**. Кликните на нужную вкладку, чтобы открыть её.

Вкладки **Nios II Application Properties** и **Nios II Library Properties** очень похожи. Эти вкладки позволяют вам контролировать следующие свойства проекта:

- имя требуемого **.elf** файла (только в проекте приложения);
- имя библиотеки (только в проекте библиотеки);
- список символов, определяемых в **makefile**;

- список символов, неопределяемых в makefile;
- список флагов ассемблера;
- флаги уровня предупреждений;
- список пользовательских флагов;
- генерация отладочных символов;
- уровень оптимизации компилятором;
- генерация объектного файла дампа (только для проекта приложения);
- управление исходными файлами;
- путь до ассоциированного BSP (необходим для приложений, опционален для библиотеки).

Конфигурация свойств BSP

Для конфигурирования настроек и свойств BSP используется редактор Nios II BSP. За подробной информацией о редакторе Nios II BSP, обратитесь к секции "Использование редактора BSP" на странице 2-11.

Исключение из сборки не поддерживается

Команда **Exclude from Build** (исключение из сборки) не поддерживается. Вы должны использовать взамен команды **Remove from Nios II Build** и **Add to Nios II Build**.

Такое поведение отличается от поведения Nios II SBT на Eclipse в версии 9.1.

Выбор корректного запуска типа конфигурации

Если вы попытаетесь отлаживать программный проект Nios II, запуская тип конфигурации как CDT локальное C/C++ приложение, вы увидите сообщение об ошибке, а перспектива отладки Nios II не откроется. Это ожидаемое поведение CDT на платформе Eclipse. Локальное C/C++ приложение запускает тип конфигурации для стандартного CDT проекта. Вызывайте плагин Nios II, вы должны использовать запуск типа конфигурации - Nios II.

Переименование проектов Nios II

Для переименования проекта в Nios II SBT на Eclipse, выполните следующие шаги:

1. Правым кликом на проект выберите **Rename**.
2. Введите новое имя проекта.
3. Правым кликом на проект выберите **Refresh**.

Если вы пропустите обновление (**Refresh**) проекта, при попытке собрать его, вы увидите следующее сообщение об ошибке:

Resource *<original_project_name>* is out of sync with the system

"Ресурс *<оригинальное имя проекта>* вышел из синхронизации с системой".

Запуск основных сценариев (скриптов) из SBT на Eclipse

Большинство SBT утилит реализовано в виде основных скриптов. Вы можете использовать конфигурацию внешних инструментов Eclipse для запуска основных скриптов. Однако вы должны проверить, что основная среда корректно настроена.

Для запуска основных скриптов из SBT на Eclipse, выполните следующие пункты:

1. Запустите командный процессор Nios II, как описано в главе "Начало работы в командной строке" в "Настольной книге программиста Nios II".
2. Запустите Nios II SBT на Eclipse, введя следующую команду:

```
eclipse-nios2
```

Вы должны запускать SBT на Eclipse из командной строки и в Linux, и в Windows, чтобы создать корректную среду командного процессора.

3. В меню Eclipse Run выберите **External Tools**, и кликните **External Tools Configurations**.
4. Создайте новую конфигурацию инструментов или откройте существующую.
5. На вкладке **Main** установите **Location** и **Argument** как показано в таблице 2–3.

Table 2–3. Location and Argument to Run Shell Script from Eclipse

Platform	Location	Argument
Windows	<code>\${env_var:QUARTUS_ROOTDIR}\bin\cygwin\bin\sh.exe</code>	<code>-c "<script name> <script args>"</code>
Linux	<code>\${env_var:SOPC_KIT_NIOS2}/bin/<script name></code>	<code><script args></code>

Например, для запуска команды `elf2hex --help`, установите **Location** и **Argument** как показано в таблице 2–4.

Table 2–4. Location and Argument to Run `elf2hex --help` from Eclipse

Platform	Location	Argument
Windows	<code>\${env_var:QUARTUS_ROOTDIR}\bin\cygwin\bin\sh.exe</code>	<code>-c "elf2hex --help"</code>
Linux	<code>\${env_var:SOPC_KIT_NIOS2}/bin/elf2hex</code>	<code>--help</code>

6. На вкладке **Build** проследите, чтобы **Build before launch** и связанные опции были установлены правильно.

По умолчанию, новая конфигурация инструментов собирает все проекты в вашей рабочей области, прежде чем исполнять команду. Это не является необходимым.

7. Кликните **Run**. Команда выполнится в командном процессоре Nios II, а результат появится во вкладке Eclipse **Console**.