

3. Использование Flash Programmer из командной строки

Инструменты разработки Nios II предлагают утилиты командной строки, предоставляющие вам полный контроль над средствами Nios II Flash Programmer. Вы можете создавать собственные скрипт файлы для автоматизации задач программирования флеш.

Также вы можете использовать графическую оболочку флеш программатора. В главе 2, "Использование Flash Programmer GUI" описывается графическая оболочка.

Nios II Flash Programmer GUI программирует флеш память путём создания скриптов, основанных на утилитах командной строки. Скрипты правильно организованы, подогнаны под ваш проект и читабельны. Вы можете использовать их как справку о синтаксисе командной строки флеш программатора. Сгенерированные в GUI скрипты очень полезны, если вам требуется использовать параметр **--instance** из табл. 3-2.

После того, как вы успешно запрограммируете флеш память, используя Nios II Flash Programmer GUI, вы сможете найти скрипты флеш программатора в директории, заданной вами в диалоговом окне **Staging Directories**, доступного из меню **Options**. Скрипты флеш программатора - это файл с именем flash_programmer.sh.

В табл. 3-1 перечислены утилиты командной строки Nios II Flash Programmer.

Табл. 3-1. Утилиты командной строки Flash Programmer

nios2-flash-programmer	Программирует S-record файл во флеш память. Также может вычитывать данные, верифицировать данные, давать отладочную информацию о чипе флеш памяти и т.д.
sof2flash	Конвертирует SRAM объектный файл (.sof) в S-record файл.
elf2flash	Конвертирует исполняемый файл Nios II (.elf) в S-record файл.
bin2flash	Конвертирует произвольный файл данных в S-record файл.

Основная утилита для программирования флеш памяти из командной строки - это nios2-flash-programmer. Ей требуется наличие входных файлов стандарта S-record. Все эти утилиты предназначены для того, чтобы входной файл был совместим с Nios II Flash Programmer. Имена входных файлов для всех утилит должны иметь определённое расширение, такое как .elf или .flash.

На компьютерах с Windows, когда вы запускаете Nios II Command Shell, утилиты флеш программатора доступны в вашей директории поиска по умолчанию.

За дополнительной информацией о Nios II Command Shell, обратитесь к главе "[Начало работы из командной строки](#)" в Настольной книге программиста Nios II.

В следующих параграфах описываются утилиты и их функции.

nios2-flash-programmer

Утилита nios2-flash-programmer программирует подготовленный файл в определённую флеш память. В качестве загрузочного файла - файл стандарта S-record, обычно создаваемый одной из утилит конвертации sof2flash, elf2flash или bin2flash. В качестве загрузочного файла nios2-flash-programmer может использовать любой S-record файл с заявленным офсетом от начала флеш памяти. Nios II Flash Programmer GUI создаёт файлы флеш программатора с расширением .flash.

Утилита nios2-flash-programmer позволяет программировать, стирать или читать из любой CFI-совместимой флеш памяти или чипа EPCS из целевого аппаратного проекта.

Синтаксис командной строки nios2-flash-programmer следующий:

```
nios2-flash-programmer [--help] [--cable=<cable name>]\
[--device=<device index>] [--instance=<instance>]\
[--sidp=<address>] [--id=<id>] [--timestamp=<time>]\
[--accept-bad-sysid] --base=<address> [--epcs]\
{ <file> } [--go]
```

Прежде чем вы сможете запрограммировать флеш память на вашей плате, вы должны сконфигурировать ваш FPGA с помощью целевого проекта флеш программатора, который состоит из одного из минимальных наборов компонентов, перечисленных в табл. 1-1 на стр. 1-3.

За инструкциями о конфигурировании FPGA, обратитесь к главе "[Quartus II Programmer](#)" в томе 3 Настольной книги Quartus II.

В табл. 3-2 перечислены часто встречаемые параметры, используемые с nios2-flash-programmer.

Табл. 3-2. Параметры nios2-flash-programmer (часть 1 из 3)

Имя	Требуется	Описание
Основные параметры		
--cable=<cable name>	Требуется, если к хосту подключено несколько загрузочных кабелей.	Определяет, какой кабель использовать. (1)
--device=<device index>	Требуется, если несколько чипов находятся в цепи JTAG.	Определяет порядковый номер чипов FPGA в цепи JTAG. Этот номер определяет чип, который флеш программатор рассматривает в качестве отладочного модуля Nios II JTAG. Чипы JTAG нумеруются в порядке цепи JTAG, начиная с 1. (2)
--instance=<instance>	Требуется, если несколько процессоров с отладочным модулем JTAG находятся в целевом проекте FPGA.	Определяет, какой отладочный модуль Nios II JTAG рассматривать в этом FPGA. ID элемента определяет отладочный модуль JTAG, используемый для программирования флеш памяти. (3)
--sidp=<address>	Опционально; требуется для проверки ID системы.	Содержит базовый адрес компонента System ID вашей системы. Это значение приводится в шестнадцатеричном формате, например 0x01000000. (4)

Табл. 3-2. Параметры nios2-flash-programmer (часть 2 из 3)

Имя	Требуется	Описание
Основные параметры		
--id=<id>	Опционально; требуется для проверки ID системы.	Содержит значение ID, программируемое в компонент System ID вашей системы. Это значение выбирается случайным образом каждый раз, при регенерации системы SOPC Builder. Это значение имеет беззнаковый десятичный формат, например, 2056847728u. (5)
--timestamp=<time>	Опционально; требуется для проверки достоверности времени создания.	Содержит значение времени создания, программируемое в компоненте System ID вашей системы. SOPC Builder определяет это значения по времени генерации системы. Это значение имеет беззнаковый десятичный формат, например 1177105077u. Использование этого параметра аналогично выключению Ignore mismatched system timestamp в диалоговом окне аппаратных подключений Nios II Flash Programmer GUI. (6)
--accept-bad-sysid	Опционально; по умолчанию выключено.	Используется для пропуска проверки ID системы. Принуждает флеш программатор загрузить образ флеш. Включение этого параметра аналогично включению Ignore mismatched system timestamp в диалоговом окне аппаратных подключений Nios II Flash Programmer GUI.
--erase=<start>,<size>	Опционально; по умолчанию выключено.	Стирает диапазон байтов во флеш памяти.
--erase-all	Опционально; по умолчанию выключено.	Стирает выбранную флеш память целиком. Операция стирания происходит перед программированием, если предоставляется входной файл для программирования.
--program	Опционально; по умолчанию включено, если задан входной файл для программирования.	Программирует флеш память входным файлом для программирования.
--no-keep-nearby	Опционально; по умолчанию выключено.	Не использует определённый сектор данных. Если данные для программирования занимают первый или последний сектор не полностью, флеш программатор обычно сохраняет и перепрограммирует оригинальные данные в этих секторах. Параметр --no-keep-nearby запрещает эту функцию. Эта опция ускоряет процесс программирования, но её используют, когда содержимое во флеш памяти не актуально.
--verify	Опционально; по умолчанию выключено.	Верифицирует содержимое флеш памяти с входными файлами.
{ <file> }	Опционально	Определяет имена входных файлов для программирования и верификации. Разделяйте имена нескольких файлов пробелом.
--read=<file>	Опционально; по умолчанию выключено.	Читает содержимое памяти в файл.
--read-bytes=<start>,<size>	Опционально, если --read задано, по умолчанию выключено	Определяет адресное пространство для чтения (побайтовая адресация).
--go	Опционально; по умолчанию выключено.	Запускает процессор по вектору сброса после окончания программирования.
Параметры CFI		
--debug	Опционально; по умолчанию выключено.	Выводит отладочную информацию, включая таблицу запроса флеш памяти.

Табл. 3-2. Параметры nios2-flash-programmer (часть 3 из 3)

Имя	Требуется	Описание
Параметры CFI		
--base=<address>	Требуется	Определяет базовый адрес CFI-флеш памяти. Этот параметр является абсолютным адресом в адресном пространстве целевого проекта. nios2-flash-programmer обращается к адресам в S-record файлах как к офсетам по отношению к базовому адресу.
Параметры EPCS		
--epcs	Требуется при программировании EPCS; по умолчанию выключено.	Определяет, какая из целевых чипов флеш памяти является чипом EPCS.
--debug	Опционально; по умолчанию выключено.	Выводит отладочную информацию о содержимом физической памяти внутри чипа EPCS.
--base=<address>	Требуется	Определяет базовый адрес чипа EPCS,

Примечания к табл. 3-2:

- (1) Параметр --cable необходим только при наличии нескольких загрузочных кабелей, подключенных к хосту. Для определения имён кабелей запустите **jtagconfig**.
- (2) Параметр --device необходим только при наличии двух и более процессоров в различных чипах с одинаковыми ID элемента. Для определения индекса устройства JTAG запустите **jtagconfig**.
- (3) Существуют два способа найти правильное значение ID элемента для процессора. Простейший - это использовать Nios II Flash Programmer GUI для создания примера скрипта флеш программатора. Обратитесь к главе 2 "Использование Flash Programmer GUI" за подробной информацией. Другой способ - это открыть файл <Quartus II project name>.jdi в директории проекта Quartus II. Найти узел процессора Nios II, путём поиска значения hpath, состоящего из <processor module name>=. ID элемента определяется как instance_id.
- (4) В файле system.h вашего пакета поддержки платы (BSP), базовый адрес ID системы определяется SYSID_BASE.
- (5) В файле system.h и в вашем BSP значение ID системы определяется SYSID_ID.
- (6) В файле system.h и в вашем BSP время создания ID системы определяется SYSID_TIMESTAMP.

Чтобы получить дополнительные параметры, введите в командной строке nios2-flash-programmer --help.

Примеры командной строки nios2-flash-programmer

```
nios2-flash-programmer --cable="Usb-blaster [USB-0]" --base=0x200000\
--program ext_flash.flash
```

Программирует CFI флеш память по базовому адресу 0x200000 входным файлом ext_flash.flash, используя загрузочный кабель "Usb-blaster [USB-0]".

```
nios2-flash-programmer --epcs --base=0x02100000 epcs_controller.flash
```

Программирует чип EPCS по базовому адресу 0x02100000 входным файлом epcs_controller.flash.

```
nios2-flash-programmer --base=0x200000 --read=current.srec \
--read-bytes=0,0x10000
```

Читает 0x10000 байт из CFI флеш памяти по базовому адресу 0x200000 и записывает содержимое в файл с именем current.srec.

```
nios2-flash-programmer --base=0x200000 --erase=0x8000,0x10000
```

Стирает адресное пространство с 0x8000 по 0x10000 в CFI флеш памяти по базовому адресу 0x200000.

```
nios2-flash-programmer --base=0x200000 --debug
```

Запрашивает CFI флеш память по базовому адресу 0x200000 и выдаёт результат. Дамп этой команды - таблица запроса флеш памяти.

sof2flash

Утилита sof2flash берёт объектный файл SRAM и транслирует его в S-record файл, подходящий для программирования флеш памяти.

В табл. 3-3 перечислены основные параметры, используемые с sof2flash.

Табл. 3-3. Параметры sof2flash

Имя	Требуется	Описание
Основные параметры		
--compress	Опционально	Включает компрессию. Доступен в чипах Cyclone® II, Cyclone III, Stratix® II и Stratix III.
--input=<file>	Требуется	Имя входного объектного файла SRAM.
--output=<file>	Требуется	Имя выходного файла.
Параметры CFI		
--offset=<addr>	Требуется	Офсет внутри чипа флеш памяти, по которому будет запрограммирована конфигурация FPGA.
Параметры EPCS		
--epcs	Требуется для EPCS; по умолчанию выключено	Определяет, что выходной файл предназначен для чипа EPCS.
Специальные параметры чипа		
--activeparallel	Опционально	Создаёт содержимое для параллельной флеш памяти, совместимое с активно-параллельным режимом конфигурации. Доступно только для FPGA, которые поддерживают активно-параллельную конфигурацию.
--pfl (1)	Требуется, если ваша конфигурация использует PFL	Определяет, что флеш программатор использует параллельный загрузчик флеш (PFL). Требуется, когда FPGA использует конфигурацию с PFL.
--optionbit=<optionbitaddr> (1)	Требуется, если ваша конфигурация использует PFL	Определяет опцию побитной адресации в вашем чипе флеш памяти. Когда используется эта опция, команда sof2flash генерирует два файла .flash и .map.flash. Когда вы программируете флеш память .map.flash файлом, он переписывает биты опции по умолчанию. Почти всегда, биты опции по умолчанию задействованы, и вам не нужно программировать этот файл во флеш память.

Примечание к табл. 3-3:

(1) Использование опций командной строки --pfl и --optionbits показано ниже. За дополнительной информацией о том, как использовать опции командной строки --pfl и --optionbits, обратитесь к [AN386](#): Using the Parallel Flash Loader with the Quartus II Software.

За дополнительными параметрами, наберите sof2flash --help.

Примеры командной строки sof2flash

```
sof2flash --offset=0x0 --input=standard.sof --output=standard_cfi.flash
```

Конвертирует standard.sof в S-record файл с именем standard_cfi.flash, предназначенный для CFI флеш памяти. Офсет S-record начинается с 0x0.

```
sof2flash --epcs --input=standard.sof --output=standard_epcs.flash
```

Конвертирует standard.sof в S-record файл с именем standard_epcs.flash, предназначенный для EPCS.

```
sof2flash --optionbit=0x18000 --pfl --input=standard.sof \
--output=standard.flash --offset=0x640000
```

Конвертирует standard.sof в S-record файл с именем standard.flash, предназначенный для параллельного загрузчика флеш, и генерирует файл перезаписи опциональных бит standard.map.flash для опциональных бит по офсету 0x18000 в этом чипе флеш памяти.

Altera рекомендует вам не использовать standard.map.flash файл, даже если вы его генерируете. Опция командной строки --optionbit необходима для корректного функционирования опции --pfl, но результирующий файл перезаписи должен игнорироваться.

elf2flash

Утилита elf2flash берёт исполняемый файл .elf и транслирует его в S-record файл, подходящий для программирования флеш памяти. Также эта утилита, если потребуется, может вставить код копировщика загрузки во флеш файл. Код копировщика загрузки вставляется перед кодом приложения в следующих случаях:

- Адрес сброса процессора попадает в адресное пространство флеш памяти для программирования.
- Исполняемый код линкован в адресном пространстве за пределами флеш памяти для программирования.

Если утилита elf2flash вставляет копировщик загрузки, она также транслирует исполняемый .elf файл в загрузочную запись, используемую копировщиком загрузки. Эта загрузочная запись содержит весь код приложения, но не исполняется. После сброса, копировщик загрузки читает загрузочную запись из флеш памяти и копирует код приложения по корректному линковочному адресу, а затем выполняет переход на только что скопированный код приложения.

В табл. 3-4 перечислены основные параметры elf2flash.

Табл. 3-4. Параметры elf2flash (часть 1 из 2)

Имя	Требуется	Описание
Основные параметры		
--input=<file>	Требуется	Имя входного объектного .elf файла.
--output=<file>	Требуется	Имя выходного файла.
Параметры CFI		
--base=<addr>	Требуется	Базовый адрес компонента флеш памяти. Утилита elf2flash использует этот параметр с ключами --end и --reset при необходимости копировщика загрузки в системе.
--end=<addr>	Требуется	Конечный адрес компонента флеш памяти. Утилита elf2flash использует этот параметр с ключами --base и --reset при необходимости копировщика загрузки в системе.

Табл. 3-4. Параметры elf2flash (часть 2 из 2)

Имя	Требуется	Описание
Параметры CFI		
--reset=<addr>	Требуется	Адрес сброса процессора, заданный в SOPC builder. Утилита elf2flash использует этот параметр с ключами --base и --end при необходимости копировщика загрузки в системе.
--boot=<file>	Требуется в следующих случаях: <ul style="list-style-type: none"> Адрес сброса процессора попадает в адресное пространство флеш памяти для программирования. Исполняемый код линкован в адресном пространстве за пределами флеш памяти для программирования. 	Определяет файл объектного кода копировщика загрузки. Игнорируется, если копировщик загрузки не требуется. Если утилита elf2flash определяет необходимость копировщика загрузки, но параметр --boot отсутствует, elf2flash выдаёт сообщение об ошибке. Поставляемый Altera копировщик загрузки находится в директории <Nios II EDS install path>/components/altera_nios2/boot_loader_cfi.srec.
Параметры EPCS		
--epcs	Требуется при создании файлов для чипа EPCS, по умолчанию выключено.	Определяет, что выходной файл предназначен для чипа EPCS.
--after=<file>	Требуется при программировании аппаратной и программной частей в чип EPCS.	Утилита использует этот параметр для размещения исполняемого кода Nios II в чипе EPCS сразу за конфигурацией FPGA. Более подробно в параграфе "Программирование аппаратной и программной частей в чип EPCS".

За дополнительными параметрами, наберите elf2flash --help.

Программирование аппаратной и программной частей в чип EPCS

Параметр --base не доступен для чипов EPCS, поскольку в чипах EPCS данные конфигурации FPGA должны начинаться по адресу 0x0. Однако, если вы программируете аппаратную и программную конфигурацию в чипе EPCS, то параметр --after прямо указывает вам на адрес позиции исполняемого программного кода, находящийся сразу после данных конфигурации FPGA.

Сконвертируйте сначала файл конфигурации FPGA, используя sof2flash. Во время конвертации исполняемого файла Nios II, используйте параметр --after и задайте имя S-record файла конфигурации FPGA. Выходной исполняемый S-record файл программы начинается с первого адреса, не используемого конфигурацией FPGA. Посмотрите второй пример параграфа "Примеры командной строки elf2flash".

Утилита elf2flash не может разместить конфигурацию FPGA в выходном файле. Она может просто зарезервировать место с адреса 0x0 достаточного размера под данные конфигурации.

Примеры командной строки elf2flash

```
elf2flash --base=0x0 --reset=0x0 --boot=boot_loader_cfi.srec \
--input=myapp.elf --output=myapp.flash
```

Конвертирует myapp.elf в S-record файл с именем myapp.flash, предназначенный для CFI флеш памяти, размещающейся по адресу 0x0. Содержит копировщик загрузки (из boot_loader_cfi.srec), который необходим для этого примера, потому что ключи --base и --reset эквивалентны.

```
elf2flash --epcs --after=standard.flash --input=myapp.elf \
--output=myapp.flash
```

Конвертирует myapp.elf в S-record файл с именем myapp.flash, предназначенный для чипа EPCS. Выходной S-record файл начинается по первому неиспользуемому файлом standard.flash адресу.

bin2flash

Утилита bin2flash конвертирует любой файл в и S-record файл, подходящий для использования флеш программатором. Вы можете использовать утилиту bin2flash для конвертирования бинарных файлов только для чтения, содержащих данные для Nios II программы, например, таблицы программной конфигурации.

В зависимости от задач вашего приложения, вы можете находить более уместным использовать файловую систему только для чтения Altera Zip, выполняющую те же функции.

За дополнительной информацией о файловой системе только для чтения Altera Zip, обратитесь к главе [Read-Only Zip File System](#) в Настольной книге программиста Nios II.

Не используйте утилиту bin2flash для конвертирования исполняемых программных файлов или файлов конфигурации FPGA. Для конвертирования программных файлов используйте elf2flash. Для конвертирования файлов конфигурации используйте sof2flash.

В табл. 3-5 перечислены основные параметры утилиты bin2flash.

Табл. 3-5. Параметры bin2flash

Имя	Требуется	Описание
Основные параметры		
--location=<addr>	Требуется	Офсет внутри флеш памяти, куда должны быть запрограммированы данные.
--input=<file>	Требуется	Имя входного бинарного файла.
--output=<file>	Требуется	Имя выходного файла.

За дополнительными параметрами, наберите bin2flash --help.

Примеры командной строки bin2flash

```
bin2flash --location=0x40000 --input=data.bin --output=data.flash
```

Конвертирует data.bin в S-record файл с именем data.flash. Размещает эти данные в S-record файле по адресу 0x40000 от начала флеш памяти.