Отладка копировщиков загрузки

Рассмотрим некоторые дополнительные возможности, появляющиеся во время добавления отладчика Nios II IDE к процессору с запущенным кодом копировщика загрузки. В следующей секции обсуждаются требования к отлаживаемым копировщикам загрузки.

Аппаратные и программные точки останова

Копировщики загрузки часто запускаются из энергонезависимой памяти, что влияет на тип точек останова, которые устанавливаются в коде. Два типа точек останова используются отладчиком Nios II: аппаратные и программные точки останова. Программные точки останова замещают инструкцию процессора в месте точки останова другой инструкцией, которая передаёт контроль отладчику. Для этого метода замещения необходимо, чтобы память программ была доступна для записи (чтобы записать инструкцию точки останова). Поскольку копировщики загрузки часто запускаются из энергонезависимой памяти, такой как флеш память, программные точки останова не могут быть установлены в копировщиках загрузки.

Аппаратные точки останова детектируют значение адреса точки останова по шине адреса инструкций, а затем передают контроль отладчику аппаратным способом. Поэтому аппаратные точки останова могут быть установлены во временной или энергонезависимой памяти. Только аппаратные точки останова могут быть установлены в копировщике загрузки, запускаемом из флеш памяти.

Разрешение аппаратных точек останова

Для разрешения аппаратных точек останова в процессоре Nios II:

1. В SOPC Builder откройте интерфейс Nios II MegaWizard, дважды кликнув на процессор Nios II в системе.
2. В интерфейсе Nios II MegaWizard кликните на вкладку **JTAG Debug Module**.
3. Выберите **debugging level 2** или выше. Уровень отладки 2 разрешает одновременно две аппаратные точки останова, которые отладчик Nios II использует автоматически.

Останов перед main()

Во время отладки копировщика загрузки вы можете пожелать запустить отладчик сразу после сброса, вместо того, чтобы ждать начала функции main(). Некоторые копировщики загрузки не содержат функцию main() вообще. В этих случаях, проинструктируйте отладчик установить точку останова на точки входа программы.

Установка отладчика

Для конфигурирования отладчика Nios II для отладки копировщика проекта:

1. Импортируйте ваш проект копировщика загрузки в Nios II IDE, выполнив следующие пункты:
    a. Откройте Nios II SBT на Eclipse™ (выберите папку рабочего пространства на свой вкус).
    b. Выберите **File** > **Import**. Откроется диалог **Import**.
    c. Раскройте папку **Nios II Software Build Tools** и выберите **Import Nios II Software Build Tools Project.**
    d. Кликните **Next**.
    e. В **Project location** отыщите папку с вашим проектом копировщика загрузки.
    f. Введите имя проекта.
    g. Выключите **Clean project** и **Managed project**.
    h. Кликните **Finish**.
2. В Nios II SBT на Eclipse™ выделите имя импортированного проекта копировщика загрузки, а в меню **Run** кликните **Debug Configurations**.
3. В диалоге **Debug Configurations** кликните на иконку **New**, чтобы создать новую конфигурацию отладки.
4. Кликните на вкладку **Target Connection**.
5. В окне **Download**:
    a. Если ваш копировщик загрузки запускается из ROM, включите **Start processor**, включите **Reset the selected target system** и выключите **Download ELF to selected target system.**
    b. Если ваш копировщик загрузки запускается из RAM, включите **Start processor**, выключите **Reset the selected target system** и включите **Download ELF to selected target system.**
6. Кликните на вкладку **Debugger**.
7. Включите **Stop on Startup at: main**.
8. Кликните **Apply**.
9. Кликните на кнопку **Debug** для запуска отладчика. Однажды подключенный, отладчик остановится на точке входа копировщика загрузки.

Внешний контроль над процессом загрузки Nios II

Другим способом загрузки процессора Nios II является наличие другого компонента, например, другого процессора, контролирующего процесс загрузки извне. В этой ситуации, внешний процессор читает код приложения Nios II из любого источника и загружает его в память программ Nios II. Внешний процессор может извлекать код приложения Nios II из различных источников. Например, он должен читать код из любой энергонезависимой среды хранения, такой как жёсткий диск, или загружать код через Ethernet соединение.

Методы, с помощью которых внешний процессор извлекает код приложения Nios II, выходят за рамки этого документа. В этой секции обсуждается процесс безопасной загрузки кода приложения в память программ Nios II, а затем перенаправление процессора Nios II на правильное исполнение приложения.

Общее представление

Доступны два независимых метода реализации внешне контролируемой загрузки системы Nios II:

- Внешний процессор распаковывает образ загрузки Nios II и записывает исполняемый код приложения в память программ Nios II.
- Внешний процессор только копирует образ загрузки в RAM. Процессор Nios II принимает управление с этого момента и распаковывает образ загрузки самостоятельно.

Последний метод позволяет процессору Nios II распаковывать и загружать приложения из образа загрузки, он похож на процесс запуска обычного копировщика загрузки в процессоре Nios II. Одно отличие заключается в том, что флеш программатор помещает образ загрузки во флеш память, а внешний процессор копирует образ загрузки в RAM. После того как внешний процессор разблокирует процессор Nios II после сброса, всё происходит так, как будто процессор Nios II загружался из флеш памяти.

В этой секции обсуждается первый метод, при котором внешний процессор распаковывает образ загрузки Nios II, записывает исполняемый код приложения в память программ Nios II и переводит процессор Nios II на точку входа приложения.

Есть одно общее требование, не зависящее от метода внешней загрузки. Вы должны запретить процессору Nios II исполнение любого кода в пространстве памяти, пока в него происходит запись внешним процессором, в процессе копирования и распаковки. В противном случае вы можете столкнуться с состоянием соперничества и проблемами повреждения данных. Процесс, описанный в этой секции, предотвращает исполнение кода процессором Nios II, держа его в сбросе, пока код приложения не скопируется в память программ Nios II. После завершения копирования кода приложения, процессор Nios II освобождается от сброса и производит исполнение приложения.

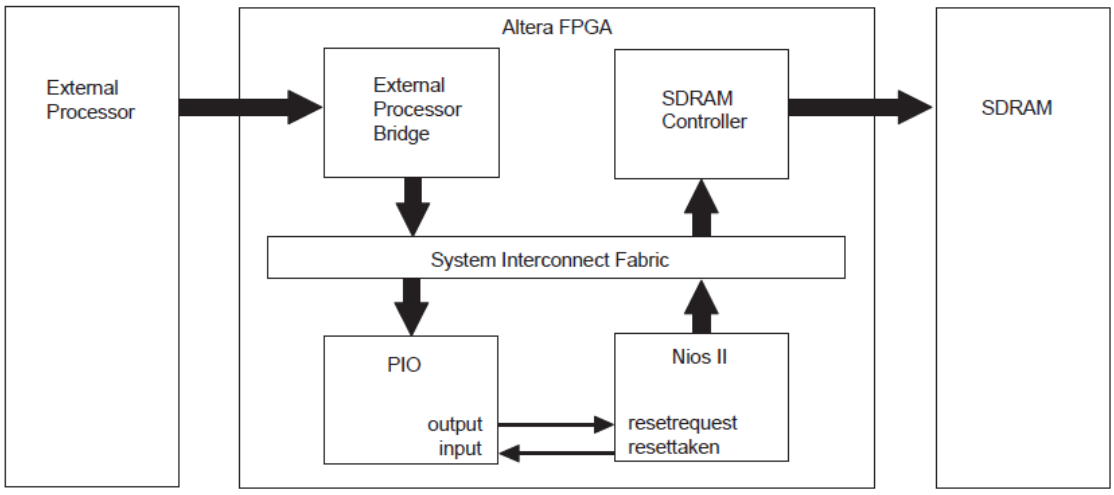
Сборка соответствующей системы SOPC Builder

Прежде чем успешно реализовать внешне контролируемую загрузку Nios II, вы должны убедиться, что ваша система SOPC Builder содержит необходимые устройства. Внешний процессор должен иметь доступ к соответствующей периферии системы и контролировать состояние сброса процессора Nios II. Следующий список приводит минимальный набор аппаратных элементов, необходимый для внешне контролируемой загрузке Nios II:

- Мост внешнего процессора
- Процессор Nios II со следующими средствами:
  - Сигнал cpu_resetrequest
  - Адрес сброса указывает на RAM
  - Однобитовое периферийное устройство параллельных IO (PIO)

На рис. 5 показана блок-схема системы, которая контролирует загрузку процессора Nios II извне.

Figure 5. Block Diagram of Externally Controlled Nios II Boot System



Мост внешнего процессора

Чтобы позволить внешнему процессору доступ к периферии вашей системы SOPC Builder, система должна иметь мост между структурой Avalon и шиной внешнего процессора.

Мосты с внешними процессорами могут быть приобретены в качестве IP ядер или разработаны собственными средствами. Многие проектировщики разрабатывают свои собственные компоненты моста внешнего процессора для SOPC Builder, поскольку это обычно относительно прямой мост между архитектурой структуры Avalon и другими протоколами шины. Редактор компонентов, доступный в SOPC Builder, прекрасно подходит для создания IP, таких как мосты внешнего процессора.

Список IP мостов, поставляемых Altera, находится на странице "Интеллектуальная собственность и связанные проекты" на веб сайте Altera.

Сигнал cpu_resetrequest

В версии процессора Nios II 6.0 и старше опциональный сигнал cpu_resetrequest стал доступен для контроля над состоянием сброса процессора. Этот сигнал отличается от обычного сигнала сброса системы SOPC Builder - reset_n – сигнал cpu_resetrequest сбрасывает только процессор Nios II. Остальная система SOPC Builder продолжает функционировать. Этот сигнал удерживает процессор Nios II в сбросе, пока код перемещается в память программ Nios II.

Сигнал cpu_resetrequest не может немедленно перевести процессор Nios II в состояние сброса. Когда сигнал cpu_resetrequest переходит в 1, процессор Nios II заканчивает исполнение начатой инструкции в конвейере, а затем переходит в сброс. Этот процесс может занять неопределённое количество тактовых циклов, пока сигнал статуса cpu_resettaken не будет переведён в 1 процессором Nios II, когда тот перейдёт в состояние сброса. Процессор удерживает этот сигнал в 1 один цикл. Сигнал cpu_resettaken продолжит периодически появляться, когда сигнал cpu_resetrequest переходит в 1.

Чтобы разрешить сигнал cpu_resetrequest, откройте проект в SOPC Builder, который содержит процессор Nios II. Дважды кликните на компонент Nios II, чтобы открыть интерфейс Nios II MegaWizard, затем кликните на вкладку **Advanced Features**. Включите **Include cpu_resetrequest and cpu_resettaken signals**, чтобы разрешить эти сигналы. Они появятся в качестве портов в верхнем уровне вашей системы SOPC Builder после регенерации системы.

Адрес сброса Nios II

Адрес сброса Nios II – это адрес первой инструкции, исполняемой процессором после возвращения из состояния сброса. Поэтому в системе Nios II с возможностью внешнего контроля загрузки, адрес сброса Nios II должен указывать на доступную для записи память (RAM). Этот класс адресов сброса обычно не используется в традиционном сценарии загрузки, но в случае ситуации внешнего контроля над загрузкой, описываемой в этой секции, важно, чтобы адрес сброса Nios II указывал на RAM.

Адрес сброса Nios II должен указывать на RAM, поскольку это направляет процессор Nios II на код приложения, который уже скопирован в RAM, а внешний процессор должен уметь записать первую инструкцию (или инструкции), которые будет исполнять процессор Nios II после сброса. Обычно инструкция, записанная по адресу сброса, - это безусловный переход (br) на точку входа приложения.

Вы можете выбрать любое незанятое 32-битное место в RAM в качестве адреса сброса для процессора Nios II, но базовый адрес (офсет 0x0) памяти программ Nios II – регион памяти, содержащий секцию .text – это наилучший выбор. По умолчанию, таблица исключений Nios II размещена по офсету 0x20 в памяти программ, а остаток кода приложения размещён следом за таблицей исключений в непрерывной памяти. Такой порядок оставляет доступными офсеты с 0x0 по 0x1C. Адрес сброса по офсету 0x0 гарантирует, что разность между адресом сброса и точкой входа приложения – предполагается вначале кода приложения – никогда не будет превышать 64 КБайта, что необходимо для работы этого процесса. За разъяснением, почему разница не может превышать 64 КБайта, обратитесь к п.1 на стр. 29.

Однобитовая PIO периферия

Однобитовая PIO периферия необходима внешнему процессору для контроля над сигналом Nios II cpu_resetrequest. Внешний процессор получает доступ к распределённой по Avalon PIO периферии через мост внешнего процессора. Внешний процессор пишет 1 в PIO для назначения вывода cpu_resetrequest или 0 – для снятия назначения.

Внешний процессор может также читать состояние сигнала cpu_resettaken, используя ту же PIO периферию. Однако, процессор Nios II назначает сигнал cpu_resettaken только на один тактовый цикл за раз. Поэтому программный захват этого сигнала, чтобы увидеть установку сигнала сброса, не работает. Сигнал может запросто появиться и исчезнуть между отсчётами так, что истинное назначение сигнала cpu_resettaken процессором Nios II может никогда не быть захваченным внешним процессором.

Компонент PIO, включённый в систему SOPC Builder, имеет средство захвата фронта, которое можно использовать в этой ситуации. Средство захвата фронта устанавливает бит в регистре фронта захвата PIO, когда фронт предопределённого типа наблюдается на этом бите входного порта PIO. Внешний процессор может прочитать регистр захвата фронта в любое время после назначения сигнала cpu_resetrequest. Если появляется сигнал cpu_resettaken в любое время после назначения cpu_resetrequest, соответствующий бит в регистре фронта захвата PIO также устанавливается.

Чтобы добавить компонент PIO, сконфигурированный для использования средства захвата фронта, чтобы детектировать появление сигнала cpu_resettaken в вашей системе, выполните следующие пункты:

1. Откройте вашу систему SOPC Builder.
2. Кликните на вкладку **System Contents** >**Peripherals**> **Microcontroller Peripherals,** кликните на компонент **PIO (Parallel I/O)**.
3. Кликните **Add**.
4. В интерфейсе **PIO** MegaWizard установите ширину в один бит и выберите **Both input and output ports.**
5. Выберите вкладку **Input Options**, отметьте **Synchronously Capture** и выберите **Rising Edge**
6. Кликните **Finish**, чтобы добавить компонент PIO в вашу систему.

В вашей системе теперь есть компонент PIO, совместимый с назначением сигнала cpu_resetrequest в процессоре Nios II и детектировании нарастающего фронта сигнала cpu_resettaken.

Система SOPC Builder не можете автоматически подключать входные и выходные порты компонента PIO к сигналам Nios II cpu_resettaken и cpu_resetrequest. После генерации SOPC Builder вы должны создать эти подключения в верхнем уровне проекта Quartus II.

Процесс загрузки

Сейчас вы изучили важные аппаратные аспекты внешне контролируемого процесса загрузки Nios II, в этой секции описываются те же процессы загрузки, относительно программы, запущенной во внешнем процессоре.

Образы загрузки

Процедуры, описанные здесь, подразумевают, что вы имеете образ загрузки в формате, описанном в секции "Образы загрузки" на стр. 5.

Пример Си кода

В директории **boot_copier_src/app/external_boot**, вы можете найти пример исходного Си кода, который вы можете запустить во внешнем процессоре для контроля над загрузкой процессора Nios II. Код часто комментирован, что упрощает его модификацию и кастомизацию. Пример кода извлекает образ загрузки по офсету 0x0 в CFI флеш, но в реальной системе, образ загрузки может быть где угодно. Эта часть процесса слева от вашего отделения.

Процесс внешней загрузки

В следующей секции описан процесс загрузки, реализованный в примере Си кода, приведённого в предыдущей секции. Эти пункты написаны для соответствующей программы, запущенной во внешнем процессоре, ответственном за контроль над процессом загрузки Nios II.

1. Извлечение образа загрузки Nios II

Программа может извлекать образ загрузки Nios II любым способом. Общие методы включают в себя чтение образа загрузки из энергонезависимой памяти (такой как жёсткий диск или флеш память), загрузка его через соединение Ethernet или простое указание его размещения в RAM. Самое главное, чтобы образ был локально доступен полностью, прежде чем вы приступите к распаковке и копировании его в программную память Nios II.

2. Удержание процессора Nios II в сбросе, используя однобитный PIO, выполнив следующие действия:

- Записать любое 32-битное число по офсету 0x3 компонента PIO, для сброса регистра захвата фронта. Использовать регистр захвата фронта для детектирования перехода сигнала cpu_resettaken в 1, необходимо для сброса регистра захвата фронта, чтобы не пропустить появление нового события.
- Записать число 1 по офсету 0x0 в компоненте PIO, чтобы назначить сигнал Nios II cpu_resetrequest.
- Непрерывно опрашивать по офсету 0x3 компонента PIO, пока он удерживает значение 1. Это значение отображает, что сигнал Nios II cpu_resettaken продолжает оставаться 1, что означает, что процессор Nios II находится в состоянии сброса, и вы можете безопасно начать копирование кода приложение в его память программ.

3. Копирование приложения по адресу назначения в пространство памяти.

Распаковка записи загрузки для копирования каждой секции кода приложения в соответствующее место программной памяти Nios II. В конце каждой записи в записи загрузки есть запись перехода. Проследите за тем, чтобы у вас было значение перехода; оно содержит точку входа кода приложения. В следующем пункте вы должны перенаправить процессор Nios II на точку входа приложения.

4. Создание инструкции ветвления для размещения по адресу сброса Nios II

Создание инструкции ветвления (br) Nios II позволяет Nios II перейти со своего адреса сброса на точку входа кода приложения. Поскольку инструкция ветвления Nios II относительная, значение её ветвления относится к текущей инструкции, вам требуется знать и адрес сброса Nios II, и адрес точки входа приложения.

В примере кода, адрес сброса Nios II просто определён вверху файла. Если вы изменяете адрес сброса Nios II, вы должны также изменить соответствующее **#define** в коде примера.

Вычете адрес сброса из адреса точки входа (сохранённой в предыдущем пункте), чтобы получить офсет. Для конвейера Nios II, инструкции ветвления необходим офсет для связи со следующей инструкцией, так вычитая 4 из офсета, получается текущий офсет, необходимый для инструкции.

Поскольку инструкции Nios II 32-битные, каждый адрес и офсет должен быть кратным 4. Офсет, используемый в инструкции ветвления 16-битный, поэтому ваш адрес сброса должен быть менее 64 Кбайт от точки входа кода приложения в адресном пространстве памяти.

Используя кодирование инструкции ветвления из табл. 2, создайте инструкцию ветвления из офсета. Следующие Си операторы создают правильно кодированную инструкцию из адреса сброса и точки входа:

```
int offset = entry_point - reset_address;
unsigned int inst = ((offset - 4) << 6) | 0x6;
```

Table 2. Nios II Branch Instruction Encoding

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0					0					16-bit offset - 4																0x06					

5. Запишите инструкцию ветвления по адресу сброса Nios II
6. Выведите процессор Nios II из сброса

Запишите нуль по офсету 0x0 периферии PIO, чтобы снять назначение сигнала Nios II cpu_resetrequest. Процессор Nios II должен выйти из сброса, выполнить инструкцию ветвления, перейти на точку входа приложения и начать его исполнение.

Загрузка завершена. Процессор Nios II отключен и запущен, так что внешний процессор может быть задействован для исполнения других системных задач.