



8. Exception Handling

NII52006-10.0.0

8. Обработка исключений

Введение

В этой главе рассказывается, как писать программы для обработки исключений в архитектуре процессора Nios[®] II. Особое внимание уделяется тому, как процесс запроса аппаратного прерывания регистрируется пользовательской программой обработки прерываний (ISR) в слое аппаратной абстракции (HAL). Эта информация применима к программным проектам, созданным с помощью инструментов разработки программ под Nios II (SBT) либо в Eclipse[™], либо в командной строке.

Эта глава состоит из следующих секций:

- "Введение" на странице 8-1
- "Общее представление об обработке исключений в Nios II" на странице 8-1
- "Программа обработки прерываний" на странице 8-7
- "Улучшение характеристик ISR" на странице 8-19
- "Отладка ISR" на странице 8-26
- "Реализация HAL системы обработки исключений" на странице 8-26
- "Родственные инструкции обработки исключений" на странице 8-34

За подробной информацией об аппаратной реализации обработки исключений и аппаратных прерываниях в архитектуре Nios II, обратитесь к главе "[Программная модель](#)" в справочном пособии по процессору Nios II.

Общее представление об обработке исключений в Nios II

Процессор Nios II предлагает следующие типы исключений:

- Аппаратные прерывания
- Программные исключения, которые делятся на следующие категории:
 - Нереализуемые инструкции
 - Прерывание при возникновении непредусмотренной ситуации
 - Смешанные исключения

Процессор Nios II предлагает два разных подхода обработки аппаратных прерываний:

- Внутренний контроллер прерываний (IIC)
- Интерфейс с внешним контроллером прерываний (EIC)

Контроллеры прерываний обсуждаются в секции "Контроллеры прерываний на стр. 8-3.

Термины обработки исключений

Следующий список HAL терминов даёт представление о концепте обработки исключений:

- Контекст приложения – Статус процессора Nios II и HAL во время обычного исполнения программы, вне направителей (funnels) и обработчиков исключений.
- Переключение контекста – Процесс сохранения состояния регистров процессора Nios II при возникновении программных исключений и аппаратных прерываний, и их восстановление после выхода из программы обработки исключений или ISR.
- Исключение – Переход контроля от программы с нормальным исполнением кода, вызванный событием, внешним или внутренним, к процессору, когда требуется немедленное рассмотрение. Исключения являются программными исключениями и аппаратными прерываниями.
- Контекст исключения – Статус процессора Nios II и HAL после программного исключения или аппаратного прерывания, когда исполняется код направителя (funnel code) – обработчик программного исключения - или ISR.
- Система обработки исключений – Полная система программ, обслуживающих все исключения, включая аппаратные прерывания, и предоставляющая контроль программным обработчикам исключений и ISR.
- Задержка исключения (или прерывания) – Время с момента возникновения исключения (такого как нереализуемая инструкция или запрос на прерывание) до исполнения первой инструкции по адресу исключения (или вектору прерывания).
- Время отклика исключения (или прерывания) - Время с момента возникновения исключения до исполнения обработки.
- Служебные исключения – Дополнительные процессы, необходимые для обслуживания программных исключений или аппаратных прерываний, включая специальные процессы HAL и специальные процессы RTOS (операционная система реального времени), если они применимы.
- Код направителя (funnel code) – Предлагаемый HAL код, который устанавливает корректную конфигурацию процессора под специальный обработчик исключений, например ISR.
- Обработчик – Специфический код под тип исключения. Код обработчика отличается от кода направителя (funnel code), который озабочен задачами служебных исключений.
- Аппаратное прерывание – Исключение, явно возникающее по сигналу аппаратного запроса из внешнего устройства. Аппаратное прерывание переводит работу процессора на программу обработки прерываний, чтобы добиться обработки состояния устройства в современной манере.
- Инструкция, зависящая от реализации – Инструкция процессора Nios II, которая не поддерживается всеми реализациями ядра Nios II. Например, инструкции mul и div зависят от реализации, поскольку не поддерживаются ядром Nios II/e.
- Прерывание – аппаратное прерывание.
- Контроллер прерывания – аппаратное разрешение процессору Nios II по запросу на прерывание переводить контроль на программу обработки прерываний.
- Запрос прерывания (IRQ) – Аппаратное прерывание.

- Программа обработки прерываний (ISR) – Программа, обрабатывающая индивидуальные аппаратные прерывания.
- Неверная инструкция – Инструкция, которой не определена реализация в процессоре Nios II.
- Маскируемые исключения – Исключения, которые могут быть запрещены по status.PIE флагу, включая внутренние аппаратные прерывания, маскируемые внешние прерывания и программные исключения, но не включая внешние немаскируемые прерывания.
- Максимальное время блокировки – максимальная длительность времени, затрачиваемое системой на блокировку маскируемого исключения.
- Максимальное время маскирования – максимальная длительность времени, затрачиваемое системой на маскирование одного прерывания.
- Смешанные исключения – Программные исключения, которые не являются нереализуемыми инструкциями и не командой системного прерывания. За подробной информацией обратитесь к секции "Смешанные исключения" на стр. 8-33.
- Вложенные прерывания – смотри приоритетные прерывания.
- Приоритетные прерывания – Процесс, когда передаётся контроль высоко приоритетным прерываниям, когда низко приоритетные ISR уже запущены.
- Программные исключения – Исключения, возникающие при работе программы; т.е. все исключения, кроме аппаратных прерываний. Они включают в себя нереализуемыми инструкциями и команды системного прерывания.
- Нереализуемая инструкция – Инструкция, зависящая от реализации, которая не поддерживается текущей реализацией ядра Nios II, установленного в вашей системе. Например, в ядре Nios II/e, инструкции mul и div являются нереализуемыми.
- Наихудшая задержка исключения (или прерывания) – Значения задержки исключения (или прерывания), включая максимальное время блокировки или максимальное время маскирования. Включение максимального времени блокировки или максимального времени маскирования учитывается в тех случаях, когда происходит исключение (или прерывание) во время маскирования или блокировки.

Контроллеры прерываний

Конфигурация процессов исключений Nios II зависит от типа аппаратного контроллера прерываний. Вы выбираете аппаратный контроллер прерываний, когда устанавливаете процессор Nios II в SOPC Builder. В этой секции описываются типы контроллеров прерываний, доступные в процессоре Nios II.

За подробной информацией о выборе аппаратного контроллера прерываний обратитесь к главе "[Инсталляция процессора Nios II в SOPC Builder](#)" в настольной книге по процессору Nios II.

Внутренний контроллер прерываний

С помощью IIC обработка прерываний в Nios II реализуется по классической схеме RISC. Все типы исключений, включая аппаратные прерывания, координируются одним направителем (funnel) исключений верхнего уровня. Таким образом, все исключения (программные и аппаратные) обрабатываются кодом, расположенным в одном месте, по адресу исключений.

IIC – это простой не векторный аппаратный контроллер прерываний. После получения запроса на прерывание, IIC перемещает контроль на основной адрес

исключения. Аппаратная часть показывает, какой IRQ сейчас назначен, и позволяет программе маскировать отдельные прерывания.

С помощью IIC направитель прерывания HAL идентифицирует событие аппаратного прерывания в программе и координирует регистрацию ISR.

Внутренний контроллер прерываний доступен во всех версиях процессора Nios II.

Концепты внешних прерываний

Интерфейс EIC позволяет процессору Nios II работать с отдельным внешним компонентом контроллера прерываний. EIC может быть любым разработанным вами компонентом. Altera предлагает пример EIC – векторного контроллера прерываний (VIC).

За подробной информацией о VIC обратитесь к главе "[Векторный контроллер прерываний](#)" в руководстве пользователя по встроенной периферии IP.

С помощью EIC, аппаратные прерывания могут обрабатываться отдельно от программных исключений. Аппаратные прерывания имеют отдельные векторы и направители (funnels). Каждое прерывание может иметь собственный обработчик, или обработчики могут быть общими. Обработка программных прерываний такая же, как в IIC.

Интерфейс EIC предлагает широкие возможности для переделки ваших аппаратных прерываний. Вы можете разработать, подключить и сконфигурировать контроллер оптимально для вашего приложения.

Когда происходит внешнее аппаратное прерывание, процессор Nios II переводит контроль по индивидуальному адресу вектора, уникальному для каждого прерывания. HAL предлагает следующие сервисы:

- Регистрация ISR
- Установка таблицы вектора
- Перемещение контроля с таблицы вектора на вашу ISR

EIC может быть использовано с набором теневых регистров. Набор теневых регистров – это полная альтернатива набору регистров общего назначения, которые могут быть использованы для получения отдельного контекста в период прогона для ISR.

EIC предлагает следующую информацию о каждом аппаратном прерывании:

Адрес запрашиваемого обработчика

Адрес запрашиваемого обработчика (RHA) – задаёт адрес направителя, ассоциированный с аппаратным прерыванием. Доступность RHA для каждого прерывания позволяет процессору Nios II перепрыгивать прямо на заданный направитель прерываний для устройства, вызывающего прерывание, уменьшая задержку прерываний.

Уровень запрашиваемых прерываний

Процессор Nios II использует уровень запрашиваемых прерываний (RIL) для установки приоритета версии запроса аппаратного прерывания для любого прерывания в текущем процессе. Во время обработки прерываний, процессор Nios II обычно выбирает только прерывания с высоким приоритетом.

Набор запрашиваемых регистров

Если набор теневых регистров реализован в ядре Nios II, EIC задёт набор запрашиваемых регистров (RRS), когда им назначается запрос прерываний. Когда процессор Nios II получает запрос на прерывание, процессор переключается на набор запрашиваемых регистров. Когда прерывание имеет набор специальных регистров, ISR избегает избыточности при сохранении регистров для контекста переключения.

Множество аппаратных прерываний могут быть сконфигурированы в наборе регистров общего назначения. Однако, во время прогона программы, процессор Nios II может не разрешить приоритет между прерываниями, назначенными одному набору регистров, если это свойство специально не разрешено. В этом случае, ISR должны быть записаны, чтобы не повредить регистр.

Обратитесь к главе "[Ядро векторного контроллера прерываний](#)" в руководстве пользователя по встроенной IP периферии за примером драйвера, управляющего приоритетом внутри набора регистров.

Режим запрашиваемых NMI

Если прерывания сконфигурированы как немаскируемые прерывания (NMI), то EIC назначает запрашиваемые NMI (RNMI). Любое аппаратное прерывание может быть немаскируемым, в зависимости от конфигурации EIC. NMI обычно сигнализирует о критическом событии в системе, требующим немедленной обработки, чтобы добиться либо стабильности системы, либо вычисления характеристик в реальном времени.

Набор теневых регистров

Несмотря на то, что набор теневых регистров может быть реализован независимо от интерфейса EIC, обычно два этих средства используются вместе. Комбинируя набора теневых регистров соответственно с EIC, вы сможете минимизировать или ограничить избыточность контекста переключения для критичных аппаратных прерываний.

Задержка и время отклика

Задержка исключения (прерывания), как говорилось в предыдущей секции, это время, требуемое аппаратуре на отклик на исключение. В отличие от задержки, время отклика – это время, требуемое для начала исполнения кода по событию исключения, такому как специфическое ISR. Время отклика содержит задержку плюс время, необходимое HAL для выполнения одной или всех следующих служебных задач:

- Сохранение контекста – сохранение регистров в стеке
- Контекст переключения RTOS – вызов функций контекста переключения, если они реализованы в RTOS
- Координирование обработчика – определение источника исключения и передача контроля заданному обработчику или ISR

Если вы заинтересованы в характеристиках системы, то время отклика для вас важнее задержки, поскольку оно отображает время между физическим событием и заданным откликом системы на это событие.

В этой секции обсуждаются возможные опции обработки исключений, и их влияние на задержку и время отклика.

Внутренний или внешний контроллеры прерываний

Nios II IIC является не векторным, нуждающимся в процессоре для координирования ISR с помощью программы. В отличие от него, EIC является векторным. С помощью векторного EIC, такого как Altera® VIC, координирование ISR управляется аппаратно, освобождая процессорное время, требуемое для координирования ISR, и существенно экономя время отклика на аппаратное прерывание.

EIC не влияет на задержку или время отклика программного исключения.

Набор теневых регистров

Совместно с EIC, набор теневых регистров ускоряет отклик от аппаратного прерывания, делая ненужным сохранение регистров в стек. Это свойство не влияет на задержку прерывания, но существенно уменьшает время отклика на прерывание.

Набор теневых регистров не влияет на время отклика программного исключения.

Устройство работы

Процессор Nios II может откликаться на исключения, включая программные исключения и аппаратные прерывания. Когда процессор Nios II отвечает на исключение, он выполняет следующие задачи:

1. Сохраняет регистр status в estatus. Это означает, что если разрешены аппаратные прерывания, то установлен бит PIE в регистре estatus.
2. Запрещает аппаратные прерывания.
3. Сохраняет следующий адрес исполнения в ea (r29).
4. Передаёт контроль на соответствующий адрес исключения, то есть:
 - Программное исключение или внутреннее аппаратное прерывания – на основной адрес исключения процессора Nios II.
 - Внешнее аппаратное прерывание – заданный для этого устройства адрес прерывания.

Все типы исключений Nios II являются точными. Это означает, что после обработки исключения, процессор Nios II может возобновить исполнение инструкции, прерванное исключением.

Процессор Nios II всегда возобновляет инструкцию после завершения работы обработчика программного исключения или ISR, когда система процессов исключения возвращается в контекст приложения.

Несколько видов исключений, таких как расширенные исключения, являются опциональными для ядра процессора Nios II. Присутствие этих типов исключений зависит от аппаратной конфигурации ядра Nios II на стадии генерации системы.

Отклик процессора на аппаратные прерывания зависит от реализации контроллера прерываний. В следующих секциях описано аппаратное поведение каждого контроллера прерываний.

За подробной информацией о контроллере исключений процессора Nios II и аппаратных контроллерах прерываний, включая список опциональных типов исключений, обратитесь к главе "[Архитектура процессора](#)" в настольной книге по процессору Nios II.

Работа внутреннего контроллера прерываний

В IIC доступно до 32 независимых сигналов аппаратных прерываний. Эти сигналы прерываний позволяют программно задавать приоритет прерываний, даже если сами по себе сигналы прерываний не имеют свойства приоритета.

В IIC исключения Nios II являются не векторными. Поэтому один адрес исключений принимает контроль над всеми типами исключений. Основным направителем (funnel) исключений по этому адресу должен определять тип программного исключения или аппаратного прерывания.

Работа внешнего контроллера прерываний

В EIC процессор Nios II поддерживает произвольное количество независимых сигналов аппаратных прерываний. Обычно прерывания векторные с уровнем приоритета, реализованным аппаратно. Вектор позволяет процессору Nios II переводить контроль непосредственно на каждый ISR. Уровни аппаратного прерывания позволяют самым критичным прерываниям замещать низко приоритетные. Поскольку оба этих средства реализованы аппаратно, то система может обрабатывать прерывания без исполнения основного кода направителя (funnel) исключений.

Детали реализации векторных и приоритетных аппаратных прерываний зависят от реализации EIC. Чтобы увидеть пример реализации EIC, обратитесь к главе "[Ядро векторного контроллера прерываний](#)" в руководстве пользователя встроенной IP периферии.

HAL поддерживает внешние контроллеры прерываний, только если они подключены одним из следующих способов:

- Прямо к интерфейсу Nios II EIC
- Через порт последовательного соединения с другим EIC