

## Отладка ISR

Вы можете отладить вашу ISR, установив в ней точки останова. Отладчик полностью останавливает процессор при возникновении точки останова. Между тем, как остальные устройства в вашей системе продолжают работать. Тем не менее, неизбежно возникающие прерывания от них будут игнорированы, поскольку процессор остановлен. Вы можете использовать отладчик для пошагового исполнения вашего ISR кода, но статус других поступивших прерываний от драйверов устройств абсолютно ошибочный по состоянию на время возобновления обычного исполнения процессора. Вы должны сбросить процессор для возобновления правильного состояния системы.

В ICC, регистр `ipending (ctl4)` маскирует всё нулями на время пошагового исполнения. Такое маскирование защищает процессор от обслуживания прерываний, возникающих при пошаговом исполнении кода. В результате, если вы пошагово исполняете часть системы обработки исключений, читающей регистр `ipending`, такие как `alt_irq_entry()` или `alt_irq_handler()`, то код не детектирует никакие отложенные прерывания. Подобное решение не может повлиять на программные исключения. Вы можете устанавливать точки останова в код вашей ISR (и пошагово исполнять её), поскольку `направитель (funnel)` всегда использует `ipending` для определения устройства, вызвавшего прерывание.

## Реализация HAL системы обработки исключений

В этой секции описывается реализация HAL системы обработки исключений. Это одна из множества возможных реализаций системы обработки исключений для процессора Nios II. Некоторые средства системы обработки исключений HAL ограничены устройством Nios II, тогда как другие предлагают полный набор сервисов.

Вы можете использовать преимущества системы обработки исключений HAL без необходимости вникать в её реализацию. Подробнее о том, как устанавливать ISR, используя HAL API, в секции "Программа обработки прерываний" на стр. 8-7.

### Структура системы обработки исключений

Система обработки исключений состоит из следующих компонентов:

- Главный направитель (funnel) исключений
- Направитель (funnel) программного исключения
- Направитель (funnel) аппаратных прерываний

В IIC есть один направитель аппаратных прерываний. Направитель управляет переключением контекста процессора и службами RTOS (если операционная система имеет место). Он определяет источник IRQ и вызывает правильную ISR.

В EIC направители аппаратных прерываний конфигурируются драйвером EIC. В векторном внешнем контроллере прерываний, таком как Altera VIC, есть несколько направителей аппаратных прерываний. Каждый направитель управляет переключением контекста процессора и службами RTOS (если операционная система имеет место). Вызов ISR управляется аппаратно.

В IIC, когда процессор Nios II генерирует исключение, главный направитель исключений принимает контроль. Главный направитель исключений передаёт контроль либо направителю аппаратных прерываний, либо направителю программного исключения. Направитель аппаратных прерываний передаёт контроль одной или несколькими ISR.

Всякий раз, когда происходит исключение, система обработки исключений обслуживает либо программное исключение, либо аппаратное прерывание, при этом аппаратное прерывание имеет наивысший приоритет. HAL IIC не поддерживает вложенные прерывания, но может обрабатывать несколько аппаратных прерываний за одно переключение контекста. Подробнее в секции "Направитель аппаратных прерываний" на стр. 8-28.

В EIC главный направитель исключений обрабатывает только программные исключения. IRQ вынуждает процессор передать контроль одному из направителей прерываний, которые подведены непосредственно к ISR.

### Главный направитель исключений

Главный направитель (funnel) исключений, предлагаемый с HAL, расположен по адресу исключений процессора. Когда происходит программное исключение или внутреннее аппаратное прерывание, и контроль передаётся главному направителю исключений, он делает следующее:

1. Переключает на отдельный стек исключений (если он разрешён).
2. Сохраняет значения регистров в стеке.
3. Определяет тип исключения и передаёт контроль направителю программного исключения или направителю аппаратных прерываний.

### ***Передача аппаратного прерывания во внутреннем контроллере прерываний***

В IIC главный направитель исключений передаёт аппаратные прерывания на манер программных исключений. На рис. 8-1 показан алгоритм, который использует главный направитель исключений HAL, чтобы различать аппаратные прерывания и программные исключения.

Главный направитель исключений просматривает регистр `estatus` для определения статуса разрешения прерывания. Если бит `PIE` установлен, произошли аппаратные прерывания, разрешённые во время исключения. Если так, то главный направитель исключений передаёт контроль в направитель аппаратных прерываний. Направитель аппаратных прерываний просматривает `IRQ` биты в регистре `ipending`. Если `IRQ` назначены, направитель прерываний вызывает соответствующий обработчик прерываний.

Если аппаратные прерывания не разрешены на время исключений, не требуется просматривать `ipending`.

Если нет активных `IRQ`, т.е. нет аппаратных прерываний, то исключение является программным. В этом случае, главный направитель исключений вызывает направитель программного исключения.

Все аппаратные прерывания имеют более высокий приоритет над программным исключением.

В `EIC` `IRQ` передаются аппаратно. Главный направитель исключений `HAL` обрабатывает только программные исключения.

За подробной информацией о регистрах процессора `estatus` и `ipending` обратитесь к главе "[Программная модель](#)" в настольной книге по процессору `Nios II`.

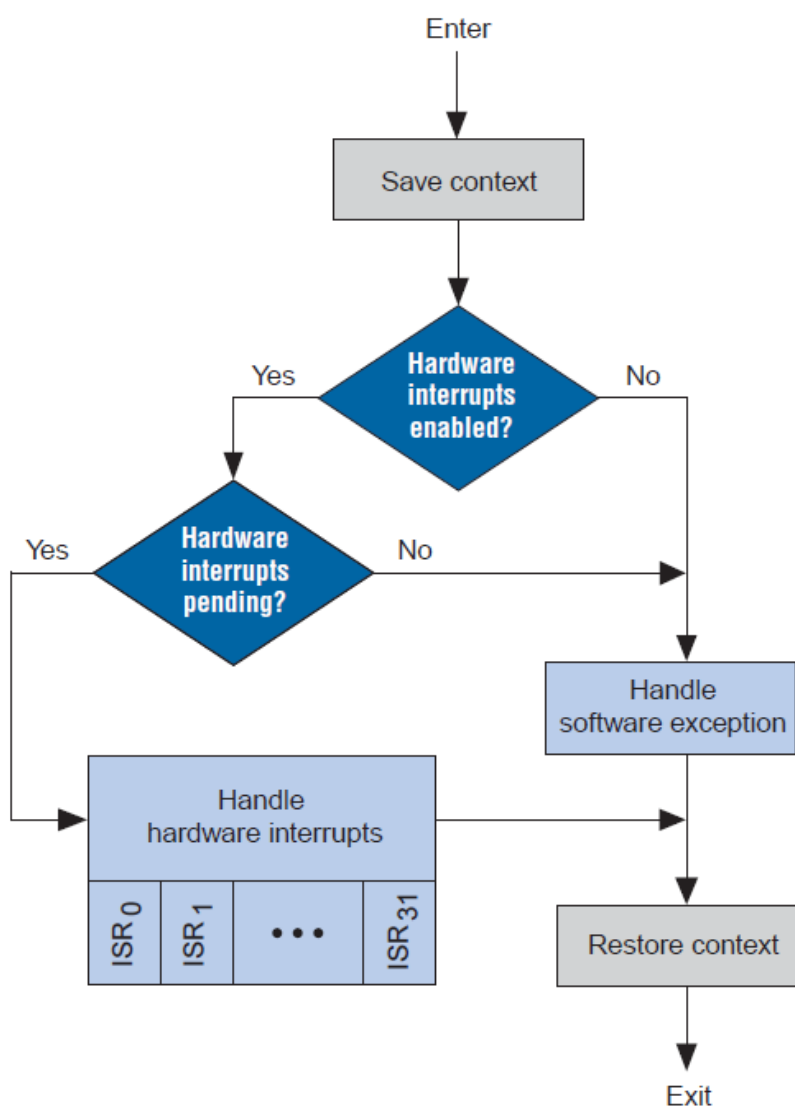
### ***Возврат из исключений***

После возврата из `ISR` или обработчика программных исключений, главный направитель (`funnel`) исключений выполняет следующие задачи:

1. Восстановление указателя стека, когда используется отдельный стек исключений.
2. Восстановление регистров из стека.
3. Выход по исполнению инструкции `eret` (`exception return`).

### **Направитель аппаратных прерываний**

Конфигурация `HAL` направителя аппаратных прерываний зависит от реализации контроллера прерываний в ядре процессора `Nios II`.

**Figure 8–1.** HAL Exception Handling System with the Internal Interrupt Controller

### **Направитель прерываний для внутреннего контроллера прерываний**

В IIC процессор Nios II поддерживает 32 аппаратных прерывания. В направителе (funnel) HAL аппаратное прерывание 0 имеет более высокий приоритет, а 31 – наименьший. Приоритеты – это средство HAL направителя, а не свойство контроллера прерываний Nios II.

Направитель аппаратных прерываний вызывает зарегистрированную пользователем ISR. Он проходит по регистру ipending, начиная с 0, и находит первое активное прерывание (наивысшего приоритета). Затем он вызывает соответствующую зарегистрированную ISR. После исполнения этой, направитель начинает снова сканировать IRQ, начиная с IRQ0. В этом случае, прерывания с наивысшим приоритетом всегда исполняются перед прерываний с низшим приоритетом. Когда сброшены все IRQ, направитель аппаратного прерывания возвращается в верхний уровень. На рис. 8-2 показана блок-схема HAL направителя аппаратного прерывания.

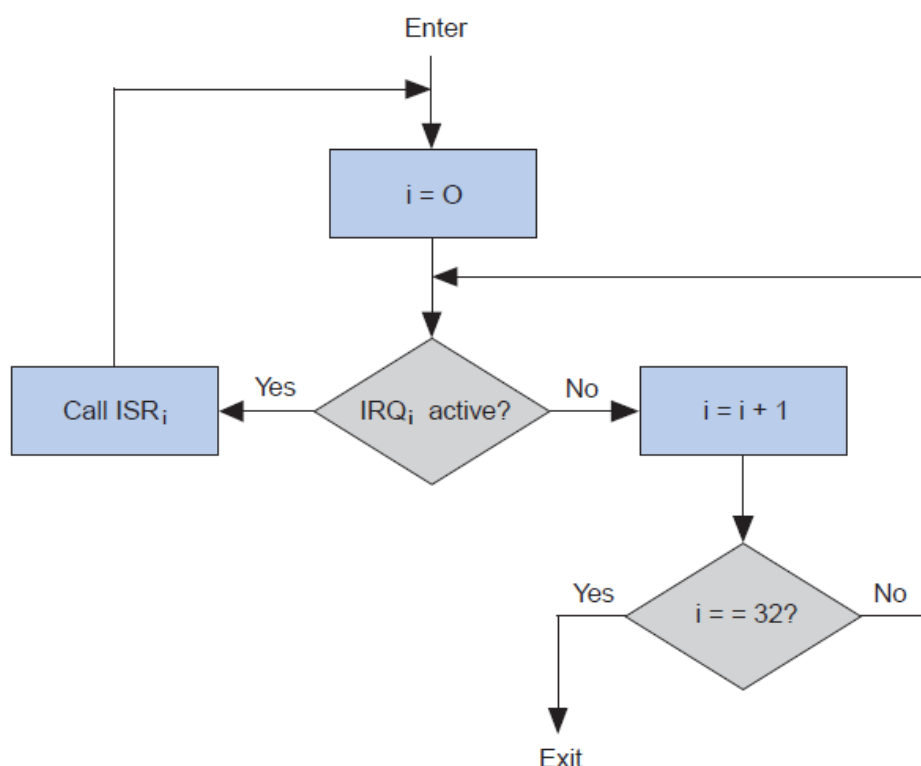
Если в процессоре Nios II представлен вектор прерываний собственных инструкций, исходный код HAL детектирует его на стадии компиляции и генерирует код, использующий собственные инструкции. За подробной информацией обратитесь к секции "Использование вектора прерываний собственных инструкций" на стр. 8-25.

**Направители прерываний для внешних контроллеров прерываний**

В EIC процессор Nios II может поддерживать потенциально неограниченное количество аппаратных прерываний от последовательных EIC. Уровень приоритета прерываний может быть сконфигурирован программно. Подробности настройки приоритета прерываний зависят от конкретной реализации EIC. Первенство в обработке высокоприоритетных прерываний отслеживается аппаратно.

Вы регистрируете ISR на стадии инициализации системы. Посылка прерывания обрабатывается аппаратно.

За подробной информацией обратитесь к секции "Структура системы обработки исключений" на стр. 8-27.

**Figure 8-2.** HAL Hardware Interrupt Funnel for the Internal Interrupt Controller

HAL предлагает следующие направители прерываний:

- набор теневых регистров, приоритетное обслуживание запрещено
- набор теневых регистров, приоритетное обслуживание разрешено
- немаскируемые прерывания

За подробной информацией обратитесь к секции "Использование направителей прерываний" на стр. 8-12.