Сигналы сброса и отладки

Ядро процессора Nios II поддерживает два типа сигналов сброса:

■ reset – Это глобальный сигнал сброса, непосредственно принуждающий ядро процессора сбросится.

■ cpu_resetrequest – Это дополнительный локальный сигнал сброса, который вызывает сброс процессора без влияния на другие компоненты системы Nios II. Процессор заканчивает исполнение всех инструкций в конвейере команд и переходит в состояние сброса. Этот процесс продолжается несколько тактов, сигнал *cpu_resetrequest* активен до появления сигнала ядра процессора *cpu_resettaken*.

Ядро процессора назначает сигнал *cpu_resettaken* в течение 1 цикла, когда сброс завершен, а потом периодически, если сигнал *cpu_resetrequest* остаётся назначенным. Когда процессор находится в состоянии сброса, он периодически обращается по адресу сброса. Он отбрасывает результат обращения и остаётся в состоянии сброса.

Процессор не реагирует на *cpu_resetrequest*, когда находится под управлением отладочного модуля JTAG, то есть когда процессор приостановлен. Процессор реагирует на сигнал *cpu_resetrequest,* если отладочный модуль JTAG перестаёт контролировать, причём моментально, за короткий промежуток времени, когда вы восстанавливаете исполнение.

За дополнительной информацией о добавлении сигналов сброса для процессора Nios II, обратитесь к "*Странице расширенные средства"* в главе *Инсталляция процессора Nios II в SOPC Builder* – главе в *Настольной книге процессора Nios II*.

Ядро процессора Nios II поддерживает сигнал отладки:

■ debugreq – Это дополнительный сигнал, который временно приостанавливает процессор в целях отладки. Когда вы назначаете этот сигнал, процессор приостанавливается так же, как при попадании на точку останова, переход работает по программе, расположенной по адресу останова, и назначает сигнал *debugack*. Назначение сигнала *debugack*, когда процессор уже приостановлен бесполезно.

За дополнительной информацией о векторе останова и о добавлении сигналов отладки для процессора Nios II, обратитесь к "*Странице отладочного модуля JTAG"* в главе *Инсталляция процессора Nios II в SOPC Builder* – главе в *Настольной книге процессора Nios II*.

Контроллеры исключений и прерываний

Процессор Nios II содержит аппаратную обработку исключений, включая аппаратные прерывания. Он также может иметь дополнительный интерфейс с внешним контроллером прерываний (EIC). Интерфейс EIC позволяет вам ускорить обработку прерываний в комплексной системе, добавив собственный контроллер прерываний.

Контроллер исключений

Архитектура Nios II имеет простой, невекторный контроллер исключений для обработки исключений всех типов. Каждое исключение, включая внутренние аппаратные прерывания, вызывают переход процессора по адресу исключений. Обработчик исключений по этому адресу определяет вызов исключения и отправку к соответствующей программе исполнения исключений.

Адрес исключений определяется в SOPC Builder на стадии генерации системы.

Все исключения точные. Точность – это средство, позволяющее процессору правильно исполнить все инструкции, предшествующие ошибочным инструкциям, и не начинать исполнение команд, следующих за ошибочными инструкциями. Точные исключения позволяют процессору вернуться к исполнению программы, после сброса исполнения обработчика исключений.

Интерфейс с внешним контроллером прерываний

EIC обычно используется вместе с набором теневых регистров, чтобы получить высоко производительные аппаратные прерывания. Процессор Nios II подключается к EIC посредством интерфейса EIC. Когда объявлен EIC, внутренний контроллер прерываний не реализуется, а SOPC Builder подключает прерывания к EIC.

EIC выбирает между активными прерываниями и представляет одно прерывание для процессора Nios II с указанием адреса обработчика прерываний и информации о наборе выбранных регистров. Алгоритм выбора прерываний уникален для каждой реализации EIC, и обычно основывается на приоритете прерываний. Процессор Nios II не зависит от выбора уникальной схемы приоритета прерываний в EIC.

Для каждого внешнего прерывания, EIC определяет уровень прерывания. Процессор Nios II использует уровень прерывания в зависимости от службы прерываний.

Некоторые внешние прерывания могут быть сконфигурированы как NMI. NMI не маскируются битом *status.PIE*, и не имеют уровня прерываний.

EIC может быть сконфигурирован программно.

Когда интерфейс EIC и набор теневых регистров реализуются в ядре Nios II, вы должны проследить, чтобы ваша программа была создана в Nios II EDS версии 9.0 или старше. В ранних версиях реализовывалась инструкция *eret*, которая не совместима с набором теневых регистров.

За типовым примером EIC обратитесь к главе *Векторный контроллер прерываний* – главе в *Томе 5: Встроенная периферия* в *Настольной книге Quartus II*. За дополнительной информацией об использовании EIC, обратитесь к "*Процессам исключений"* в *Программой модели* – главе в *Настольной книге процессора Nios II*.

Внутренний контроллер прерываний

Архитектура Nios II поддерживает 32 аппаратных прерывания. Ядро процессора имеет 32 уровне зависимых входа запроса прерываний (IRQ), с *irq0* до *irq31*, предлагая уникальный вход для каждого источника прерываний. Приоритет IRQ определяется программно. Архитектура поддерживает вложенные прерывания.

Программа может разрешить и запретить любые источники прерываний индивидуально через контрольный регистр *ienable*, который содержит бит разрешения прерываний для каждого входа IRQ. Программа может глобально разрешить и запретить прерывания с помощью бита *PIE* в *status* регистре контроля. Аппаратные прерывания генерируются тогда и только тогда, когда выполняются следующие условия:

- Бит *PIE* в *status* регистре установлен в 1.
- Назначен вход запроса прерываний, *irq<n>.*
- Соответствующий бит *n* в регистре *ienable* установлен в 1.

Вектор прерываний собственных инструкций

Ядро процессора Nios II предлагает вектор прерываний собственных инструкций, который упорядочивает увеличивающийся вектор прерываний. Используйте эту собственную инструкцию для уменьшения времени обработки прерываний в программе.

Вектор прерываний собственных инструкций основывается на приоритетном дешифраторе с одним входом для каждого прерывания, подключенным к процессору Nios II. Стоимость вектора прерываний собственных инструкций зависит от количества прерываний, подключенных к процессору Nios II. В худшем случае, это система с 32 прерываниями. В этом случае, вектор прерываний собственных инструкций будет занимать 50 логических элементов (LEs).

Если у вас большое количество подключенных прерываний, добавление вектора прерываний собственных инструкций в вашу систему может уменьшить f_{MAX}.

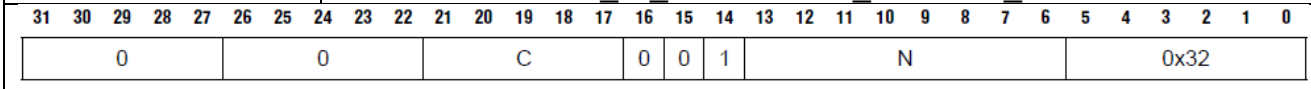
За руководством по добавлению добавление вектора прерываний собственных инструкций в процессор Nios II, обратитесь к главе *Инсталляция процессора Nios II в SOPC Builder* в *Настольной книге процессора Nios II*.

Вектор прерываний собственных инструкций не сочетается с интерфейсом EIC. Для ядра Nios II/f, интерфейс EIC с компонентом векторного контроллера прерываний Altera, приводит к наилучшим характеристикам.

В таблице 2-4 подробно описана реализация вектора прерываний собственных инструкций.

Таблица 2-4. Вектор прерываний собственных инструкций ALT_CI_EXCEPTION_VECTOR_N

Операция	if (ipending == 0) | (estatus.PIE == 0) then rC ← отрицательное значение else rC ← 8 × bit # наименее значимый 1 бит регистра *ipending* (ctl4)
Синтаксис ассемблера	custom ALT_CI_EXCEPTION_VECTOR_N, rC, r0, r0
Пример	custom ALT_CI_EXCEPTION_VECTOR_N, et, r0, r0 blt et, r0, not_irq
Описание	Вектор прерываний собственных инструкций улучшает характеристики вектора прерываний. Эти собственные инструкции определяют наивысший приоритет прерываний, генерируют сдвиг таблицы векторов прерываний и сохраняют этот сдвиг в rC. Инструкции генерируют отрицательный сдвиг, если не было аппаратного прерывания (т.е., исключение вызываемое состоянием программы, например, прерывание при возникновении непредусмотренной ситуации).
Использование	Вектор прерываний собственных инструкций используется только обработчиком исключений.
Исключения	нет
Тип инструкции	R
Поля инструкции	C = индекс регистра операнда rC N = значение ALT_CI_EXCEPTION_VECTOR_N



За разъяснениями в формате справки, обратитесь к к *Руководству пользователя по собственным инструкциям Nios II* – главе в *Настольной книге процессора Nios II*.