



9. Ядро PIO

Общее представление о ядре

Ядро параллельного входа/ выхода (PIO) с интерфейсом Avalon® предлагает интерфейс с распределением в памяти между слейв портом Avalon® с распределением в памяти (Avalon-MM) и I/O общего назначения. Порты I/O подключаются либо к пользовательской логике на чипе, либо к I/O выводам, подключенным к устройствам снаружи FPGA.

Ядро PIO предоставляет простой доступ I/O к пользовательской логике или внешним устройствам в ситуации, когда достаточно подхода "ударные биты" (bit banging). Следующие примеры используют его:

- Контроль светодиодов
- Получение данных от переключателей
- Контроль устройств дисплея
- Конфигурирование и связь с внешними устройствами, такими как стандартные продукты специфического применения (ASSP)

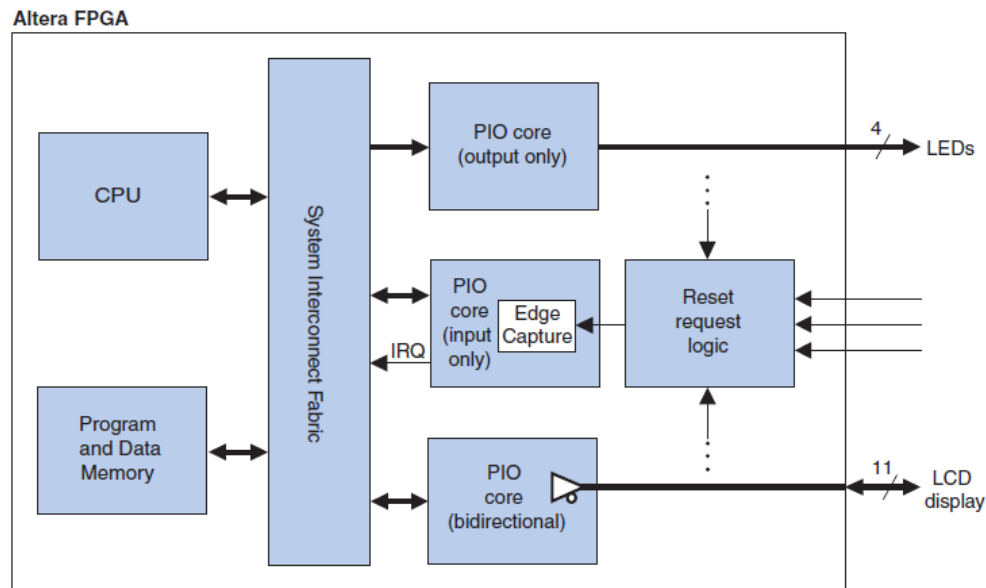
Выход запроса прерывания (IRQ) ядра PIO может вызвать прерывание по входным сигналам. Ядро PIO предназначено для SOPC Builder и легко интегрируется в любую систему, генерируемую SOPC Builder. Эта глава состоит из следующих секций:

- "Функциональное описание"
- "Примеры конфигурации" на странице 9-3
- "Инсталляция ядра PIO в SOPC Builder" на странице 9-4
- "Поддержка чипов" на странице 9-5
- "Программная модель" на странице 9-5

Функциональное описание

Каждое ядро PIO может иметь до 32 I/O портов. Интеллектуальный хост, такой как микропроцессор, контролирует порты PIO, читая и записывая Avalon-MM интерфейс с распределением в регистрах. Под контролем хоста, ядро PIO принимает данные со своих входов и посылает данные на свои выходы. Когда порты PIO напрямую подключены к I/O выводам, хост может перевести выводы в третье состояние, записав в контрольные регистры в ядре PIO. На рис. 9-1 показан пример системы на основе процессора, которая использует несколько ядер PIO для управления светодиодами, захвата фронтов от внутри чиповой логики контроля запроса сброса и для контроля над внешним LCD дисплеем.

Figure 9–1. An Example System Using Multiple PIO Cores



После интегрирования в систему SOPC Builder, ядро PIO имеет два видимых для пользователя средства:

- Пространство регистра с распределением в памяти для четырёх регистров: data, direction, interruptmask и edgecapture
- от 1 до 32 I/O портов

Порты I/O могут быть подключены к логике внутри FPGA или к выводам чипа, которые подключены к внешним устройствам. Регистры представляют собой интерфейс с I/O портами посредством интерфейса Avalon-MM. Посмотрите описание регистров в табл. 9-2 на стр. 9-6.

Вход и выход данных

Порты I/O ядра PIO могут быть подключены к логике внутри и снаружи чипа. Ядро может быть сконфигурировано только как вход, только как выход или как вход и выход. Если ядро использует для контроля двунаправленные I/O выводы на чипе, то ядро обеспечивает двунаправленный режим с контролем третьего состояния.

Аппаратная логика разделяет регистры для чтения и записи данных. Чтение регистра данных возвращает значение предустановленное на входных портах (если он есть). Запись данных влияет на значение, передаваемое в выходной порт (если он есть). Эти порты независимы; чтение регистра данных не может вернуть значение только что записанных данных.

Захват фронта

Ядро PIO может быть сконфигурировано для захвата фронтов на своих входных портах. Оно может захватывать переход 0 – 1, 1 – 0 или оба. Когда на входе детектируется фронт, состояние отображается в регистре edgecapture. Тип детектируемого фронта задаётся на стадии генерации системы и не может быть изменен с помощью регистров.

Генерация IRQ

Ядро PIO может быть сконфигурировано для генерирования IRQ при определённых состояниях входа. Состояния IRQ могут быть следующие:

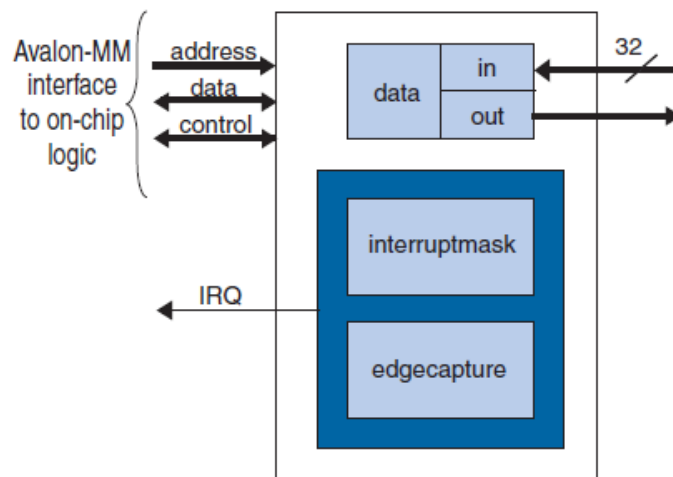
- *Чувствительные к уровню* – аппаратное ядро PIO может детектировать уровень 1. Вентиль NOT (инвертор) может быть установлен снаружи ядра, чтобы создать уровень 0.
- *Чувствительные к фронту* – конфигурация ядра для захвата фронтов может определить тип фронта, вызывающий IRQ.

Прерывания могут быть индивидуально маскируемые для каждого входного порта. Маска прерывания определяет, какой порт может генерировать прерывания.

Примеры конфигураций

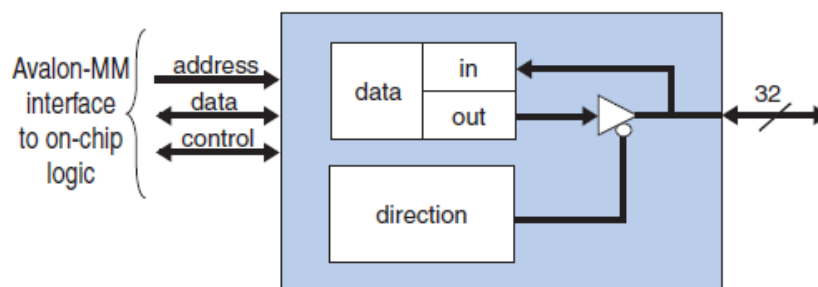
На рис. 9-2 показана блок-схема ядра PIO, сконфигурированного под входные и выходные порты, в качестве примера поддержки IRQ.

Figure 9–2. PIO Core with Input Ports, Output Ports, and IRQ Support



На рис. 9-3 показана блок-схема ядра PIO, сконфигурированного в двунаправленном режиме без поддержки IRQ.

Figure 9–3. PIO Core with Bidirectional Ports



Интерфейс Avalon-MM

Интерфейс Avalon-MM ядра PIO состоит из одного слейв порта Avalon-MM. Слейв порт предназначен для основных передач чтения и записи по Avalon-MM. Слейв порт Avalon-MM оснащён IRQ выходом, так чтобы ядро смогло вызвать прерывания.

Инсталляция ядра PIO в SOPC Builder

Используйте интерфейс MegaWizard™ для ядра PIO в SOPC Builder, чтобы сконфигурировать ядро. В следующих секциях описываются доступные опции.

Основные настройки

На странице **Basic Settings** задаются значения ширины, направления и сброса I/O портов.

Ширина

Ширина I/O портов может быть установлена целым числом от 1 до 32.

Направление

Вы можете установить направление порта одной из опций, показанных в таб. 9-1.

Табл. 9-1. Настройки направления

| Настройки | Описание |
|--|--|
| Двухнаправленные (тристабильные) порты | В этом режиме каждый бит PIO обращается к одному выводу чипа для отправки и получения данных. Направление каждого вывода индивидуально. Чтобы сделать тристабильным FPGA I/O вывод, установите направление – вход. |
| Только входные порты | В этом режиме PIO могут только принимать входные данные. |
| Только выходные порты | В этом режиме PIO могут только передавать данные на выход. |
| Входные и выходные порты | В этом режиме входные и выходные шины портов независимы, однонаправленные шины шириной n бит. |

Значение сброса выходного порта

Вы можете задать значение сброса выходного порта. Величина реального значения зависит от ширины порта.

Выходной регистр

Опция **Enable individual bit set/clear output register** позволяет вам устанавливать или сбрасывать отдельные биты выходного порта. Когда эта опция включена, реализуются два дополнительных регистра - outset и outclear. Вы можете использовать эти регистры для задания состояния выходного бита – установлен или сброшен.

Входные опции

Страница **Input Options** позволяет вам задать настройки захвата фронта и генерации IRQ. Страница **Input Options** не доступна, когда выбрана настройка **Output ports only** на странице **Basic Settings**.

Регистр захвата фронта

Включите **Synchronously capture**, чтобы добавить в ядро регистр захвата фронта – edgcapture. Регистр захвата фронта позволяет ядру детектировать и генерировать дополнительное прерывание, при появлении на входном порте фронта заданного типа. Пользователь в дальнейшем может задать следующие свойства:

-
- Выбрать тип детектируемого фронта:
 - Нарастающий фронт
 - Спадающий фронт
 - Любой фронт
 - Включить **Enable bit-clearing for edge capture register**, чтобы сбросить конкретный бит в регистре захвата фронта. Чтобы сбросить бит, запишите 1 в регистр фронта захвата.

Прерывания

Включите **Generate IRQ**, чтобы назначить выход IRQ при возникновении определённого события на входных портах. Пользователь должен задать причину IRQ события:

- Уровень – ядро генерирует IRQ, когда на определённом входе 1, а прерывания разрешены для этого входа в регистре interruptmask.
- Фронт - ядро генерирует IRQ, когда определённый бит в регистре захвата фронта – 1, а прерывания разрешены для этого входа в регистре interruptmask.

Когда **Generate IRQ** выключено, регистра interruptmask не существует.

Симуляция

Страница **Simulation** позволяет вам задать значения на входном порте во время симуляции. Включите **Hardware PIO inputs in test bench**, чтобы установить на входных портах PIO определённое значение для тестового стенда (testbench), и задайте значение в поле **Drive inputs to**.

Поддержка чипов

Ядро PIO поддерживается всеми семействами чипов Altera®.

Программная модель

В этой секции описывается программная модель для ядра PIO, включая карту регистров и программные конструкции для доступа к аппаратной части. Для использующих процессор Nios® II, Altera предлагает заголовочный файл системной библиотеки HAL, в котором определены регистры ядра PIO. Ядро PIO не относится к категории общих моделей устройств, поддерживаемых HAL, поэтому к нему невозможно получить доступ через HAL API или стандартную библиотеку ANSI Си.

Nios II Embedded Design Suite (EDS) предлагает несколько примеров проектов, которые демонстрируют использование ядра PIO. В частности, пример **count_binary.c** использует ядро PIO для управления светодиодами и детектирования нажатия на кнопку, используя прерывания по фронту PIO.

Программные файлы

Ядро PIO сопровождается одним программным файлом - **altera_avalon_pio_regs.h**. Этот файл определяет карту регистров ядра, предлагая символьные константы для доступа к аппаратному уровню устройства. Функции в этом файле используются в драйвере устройства.

Карта регистра

Мастер периферия на Avalon-MM, такая как процессор, контролирует и обменивается с ядром PIO посредством четырёх 32-битных регистров, см. табл. 9-2. В таблице допускается, что I/O порты ядра PIO могут быть сконфигурированы шириной до n бит.

Табл. 9-2. Карта регистра для ядра PIO

| Офсет | Имя регистра | | R/W | (n-1) | ... | 2 | 1 | 0 |
|-------|----------------------|--------|--------|---|-----|---|---|---|
| 0 | data | чтение | чтение | Текущее значение данных на входах PIO. | | | | |
| | | запись | запись | Новые значения поступают на выходы PIO. | | | | |
| 1 | direction (1) | | R/W | Направление контролируется для каждого I/O порта индивидуально. Значение 0 устанавливает направление – вход, 1 устанавливает направление – выход. | | | | |
| 2 | interruptmask (1) | | R/W | Разрешение/запрещение IRQ для каждого входного порта. Установите бит в 1 для разрешения прерываний от соответствующего порта. | | | | |
| 3 | edgecapture (1), (2) | | R/W | Детектирование фронта для каждого входного порта. | | | | |
| 4 | outset | | запись | Определяет, какой бит будет установлен в выходном порте. | | | | |
| 5 | outclear | | запись | Определяет, какой бит будет сброшен в выходном порте. | | | | |

Примечания к табл. 9-2:

- (1) Этот регистр может не существовать, он зависит от аппаратной конфигурации. Если регистра нет, чтение регистра возвращает неопределённое значение, а запись в регистр не имеет эффекта.
- (2) Запись любого значения в edgecapture сбрасывает все биты в нуль.

Регистр data

Чтение из регистра data возвращает текущее значение на входных портах. Если аппаратное ядро PIO сконфигурировано в режиме только выход, чтение данных возвращает неопределённое значение.

Запись в регистр данных сохраняет значение в регистре, которое поступает на выходные порты. Если аппаратное ядро PIO сконфигурировано в режиме только вход, запись данных не имеет эффекта. Если аппаратное ядро PIO сконфигурировано в двунаправленном режиме, регистрированное значение появляется на выходном порту только, когда соответствующий бит в регистре direction установится в 1 (выход).

Регистр direction

Регистр direction контролирует направление данных для каждого порта PIO, при условии, что порт двунаправленный. Когда бит n в регистре direction установлен в 1, порт n подаёт на выход значение в соответствующем биту регистре данных.

Регистр direction существует только тогда, когда аппаратная часть ядра PIO сконфигурирована в режиме двунаправленный порт. Режим (вход, выход или двунаправленный) задаётся на стадии конфигурации системы и не может быть изменён во время прогона. В режимах только выход или только вход, регистр direction не существует. В этом случае, направление чтения возвращает неопределённое значение, а записи не имеет эффекта.

После сброса, все биты регистра direction устанавливаются в нуль, таким образом, все двунаправленные I/O порты конфигурируются как входы. Если эти PIO порты подключены к выводам чипа, то выводы удерживаются в высоко импедансном состоянии. В двунаправленном режиме, для изменения направления PIO порта, необходимо перепрограммировать регистр direction.

Регистр interruptmask

Установка в 1 бита в регистре interruptmask разрешает прерывания для соответствующего входного порта PIO. Поведение прерывания зависит от аппаратной конфигурации ядра PIO. Посмотрите "Режим прерывания" на стр. 9-7.

Регистр interruptmask существует только тогда, когда аппаратная часть сконфигурирована для генерирования IRQ. Если ядро не генерирует IRQ, чтение регистра interruptmask возвращает неопределённое значение, а запись не имеет эффекта.

После сброса, все биты регистра interruptmask устанавливаются в нуль, таким образом, прерывания запрещены для всех портов PIO.

Регистр edgecapture

Бит *n* регистра edgecapture установлен в 1, когда вход порта *n* детектирует фронт. Мастер периферия на Avalon-MM читает регистр edgecapture чтобы определить, появился ли фронт на любом входном порту PIO. Если опция **Enable bit-clearing for edge capture register** выключена, запись любого значения в регистр edgecapture сбрасывает в нём все биты. С другой стороны, запись 1 в конкретный бит регистра, сбрасывает только этот бит.

Тип детектируемых фронтов фиксируется в устройстве на стадии генерации системы. Регистр edgecapture существует только тогда, когда аппаратная часть сконфигурирована для детектирования фронтов. Если ядро не сконфигурировано для захвата фронтов, чтение из регистра edgecapture возвращает неопределённое значение, а запись не имеет эффекта.

Регистры outset и outclear

Вы можете использовать регистры outset и outclear для установки и сброса конкретных битов в выходном порте. Например, для установки 6 бита в выходном порте, запишите 0x40 в регистр outset. Запись 0x08 в регистр outclear сбросит 3 бит в выходном порте.

Эти регистры существуют только тогда, когда включена опция **Enable individual bit set/clear output register**.

Режим прерывания

Ядро PIO выводит один IRQ сигнал, который может быть подключен к любой мастер периферии системы. Мастер может читать регистр data или регистр edgecapture, чтобы определить порт – источник прерывания.

Когда устройство сконфигурировано под прерывания по уровню, IRQ возникает, когда соответствующие биты регистров data и interruptmask устанавливаются в 1. Когда устройство сконфигурировано под прерывания по фронту, IRQ возникает, когда соответствующие биты регистров edgecapture и interruptmask устанавливаются в 1. IRQ остаётся вызванным, пока известно о не сброшенных соответствующих битах в регистре interruptmask или не записаны соответствующие биты в регистр edgecapture.

Оглавление

| | |
|---|-----|
| 9. Ядро PIO..... | 9-1 |
| Общее представление о ядре | 9-1 |
| Функциональное описание..... | 9-1 |
| Вход и выход данных | 9-2 |
| Захват фронта | 9-2 |
| Генерация IRQ..... | 9-3 |
| Примеры конфигураций | 9-3 |
| Интерфейс Avalon-MM | 9-3 |
| Инсталляция ядра PIO в SOPC Builder | 9-4 |
| Основные настройки | 9-4 |
| Ширина | 9-4 |
| Направление | 9-4 |
| Значение сброса выходного порта | 9-4 |
| Выходной регистр | 9-4 |
| Входные опции | 9-4 |
| Регистр захвата фронта | 9-4 |
| Прерывания..... | 9-5 |
| Симуляция | 9-5 |
| Поддержка чипов..... | 9-5 |
| Программная модель | 9-5 |
| Программные файлы | 9-5 |
| Карта регистра..... | 9-6 |
| Регистр data | 9-6 |
| Регистр direction | 9-6 |
| Регистр interruptmask..... | 9-7 |
| Регистр edgecapture..... | 9-7 |
| Регистры outset и outclear | 9-7 |
| Режим прерывания | 9-7 |