

Кэш память

Архитектура Nios II поддерживает кэш память для мастер порта инструкций (кэш инструкций) и для мастер порта данных (кэш данных). Кэш память размещается внутри чипа в виде интегрированной части ядра процессора Nios II. Кэш память может улучшить характеристики времени доступа для процессорной системы Nios II, которая использует медленную внешнюю память SDRAM для хранения программы и данных.

Кэш инструкций и данных является постоянным после точки запуска, но в этом случае программа пропускает кэш данных, так чтобы доступ периферии не возвращал кэшированные данные. Управление кэшем и когерентность кэша обрабатывается программно. Инструкции Nios II содержат набор инструкций для управления кэшем.

Конфигурируемые настройки кэш памяти

Кэш памяти опциональны. Потребность в улучшении характеристик памяти (и, соответственно, потребность в кэш памяти) зависит от приложений. Многие приложениям требуется наименьшее возможное ядро процессора, поэтому размер важнее характеристик.

Ядро процессора Nios II может иметь одну, две или вовсе не иметь кэш памяти. Кроме того, для ядер, имеющих кэш данных и/или инструкций, размер кэш памяти устанавливается пользователем. Использование кэш памяти не зависит от функционирования программы, но влияет на скорость выбора инструкций и скорость записи/чтения данных.

Эффективное использование кэш памяти

Эффективность использования кэш памяти для улучшения характеристик зависит от следующих условий:

- Регулярная память расположена снаружи чипа, а время доступа к ней больше, чем к внутри чиповой памяти.
- Большинство критичных по характеристикам циклов инструкций меньше, чем кэш инструкций.
- Большинство блоков критичных по характеристикам данных, чем кэш данных.

Оптимальная конфигурация кэша зависит от приложений, хотя вы можете создать решение, которое будет эффективным для большого количества приложений. Например, если процессорная система Nios II содержит только быструю память на чипе (т.е. она никогда не обращается к медленной внешней памяти), кэш инструкций и данных вряд ли увеличит характеристики системы. Другой пример, если критичный программный цикл имеет размер 2 КБайта, а размер кэша инструкций – 1 Кбайт, то кэш инструкций не сможет увеличить скорость исполнения. Кэш инструкций в этой ситуации способен ухудшить характеристики.

Если приложениям всегда требуются определённые данные или секции кода, то это веский довод улучшения характеристики кэш памяти, средство сдвоенной памяти может предоставить большое количество решений. Обратитесь на страницу 2-15 за подробным описанием "Сдвоенной памяти".

Методы пропуска кэша

Архитектура Nios II предлагает следующие способы пропуска кэша данных:

- Инструкции загрузки и хранения I/O
- Пропуск 31-битом кэша

Инструкции загрузки и хранения I/O

Инструкции загрузки и хранения I/O, называемые **ldio** и **stio**, пропускают кэш данных и перемещают данные по Avalon-MM на специальный адрес.

Пропуск 31-битом кэша

Метод пропуска 31-битом кэша по мастер порту данных использует 31 бит адреса в качестве признака, показывающего, что процессор должен перемещать данные в/из кэша, или пропускать их. Это удобно для программы, которой нужно держать в кэше определённые адреса и пропускать другие. Программа может пропустить адреса в качестве параметров между функциями, без надобности определять в будущем информацию о том были ли кэшированы адресные данные или нет.

Чтобы узнать, какие ядра реализуют какие методы пропуска кэша, обратитесь к *Подробностям реализации ядер* – главе в *Настольной книге процессора Nios II*.

Сдвоенная память

Сдвоенная память предлагает гарантированный доступ с малой задержкой к памяти для критичных по характеристикам приложений. Применительно к кэш памяти, сдвоенная память обладает следующими преимуществами:

- Характеристики аналогичные кэш памяти
- Программа может гарантировать, что критичный по характеристикам код или данные будут размещены в сдвоенной памяти.
- Отсутствуют издержки кэширования в реальном времени, такие как загрузка, инвалидация или смещение памяти.

Физически сдвоенный порт памяти – это отдельные мастер порты процессорного ядра Nios II, аналогично мастер портам инструкций или данных. Ядро Nios II может иметь нуль, одну или несколько сдвоенных памяти. Архитектура Nios II поддерживает сдвоенную память для доступа к инструкциям и данным. Каждый порт сдвоенной памяти напрямую подключен к одной памяти с гарантированной низкой постоянной задержкой. Память является внешней для ядра Nios II и обычно располагается на чипе.

Доступ к сдвоенной памяти

Сдвоенная память занимает обычное адресное пространство, такое же, как и другие устройства памяти, подключенные через систему внутренних соединений. Величина адреса для сдвоенной памяти (если требуется) определяется временем генерации системы.

Программный доступ к сдвоенной памяти использует обычные инструкции записи и сохранения. С точки зрения программы, доступ к сдвоенной памяти не отличается от доступа к другой памяти.

Эффективное использование сдвоенной памяти

Система может использовать сдвоенную память, чтобы достигать максимальных характеристик при доступе к специфическим секциям кода или данных. Например, приложения с интенсивными прерываниями могут размещать код обработчика исключительных операций в сдвоенной памяти, чтобы уменьшить задержку прерываний. Аналогично, цифровые сигнальные процессоры (DSP) с интенсивными вычислительными приложениями могут размещать буферы данных в сдвоенной памяти для наибольшего ускорения доступа к данным. Если потребности в памяти для приложений меньше, по отношению к возможности размещения её на чипе, можно использовать сдвоенную память отдельно для кода и данных. Большие приложения должны селективно выбирать, что нужно включать в сдвоенную память для выявления компромисса со стоимостью устройства.

Адресная карта

Адресная карта для памяти и периферии в процессорной системе Nios II зависит от конкретного проекта. Вы определяете адресную карту на стадии генерации системы.

Есть три адреса, которые являются частью процессора и заслуживают отдельного упоминания:

- Адрес сброса
- Адрес исключений
- Адрес обработчика останова

Программисты имеют доступ к памяти и периферии, используя макросы и драйверы. Поэтому гибкая адресная карта не оказывает влияния на разработчиков приложений.

Диспетчер памяти

Опциональный Nios II MMU предлагает следующие средства и функции:

- Отображение адресов от виртуального к физическому
- Защита памяти
- 32-битные виртуальные и физические адреса, отображаемые как 4 ГБайта виртуального адресного пространства и более чем 4 ГБайта физической памяти
- Размер страницы и фрейма 4 КБайта
- Нижние 512 Мбайт физического адресного пространства доступны для прямого доступа
 - Аппаратные буферы быстрого преобразования адреса (TLBs), доступ к переходам адреса
 - Отдельные TLBs для доступа к инструкциям и данным
 - Права для чтения, записи и исполнения контролируются на каждой странице
 - Поведение по умолчанию контролируется на каждой странице
 - TLBs работающие как *n*-way ассоциативный кэш с множественным доступом для программных страниц таблиц
 - TLB размер и ассоциативность конфигурируется на стадии генерации системы
 - Формат страниц таблиц (или эквивалента структуры данных) определяется системной программой
 - Политика перемещения для элементов TLB определяется системной программой
 - Политика записи для элементов TLB определяется системной программой

За подробной информацией о реализации MMU, обратитесь к *Программной модели* – главе в *Настольной книге процессора Nios II*.

Вы можете опционально включить MMU, когда устанавливаете процессор Nios II в вашу аппаратную систему Nios II. Когда это сделано, MMU всегда разрешён, а кэш данных и инструкций являются виртуально проиндексированным, физически помеченным кэшем. Доступны несколько параметров, позволяющих вам оптимизировать MMU в соответствии с системными требованиями.

За подробной информацией настраиваемых пользователем параметров Nios II MMU, в главе *Инсталляция процессора Nios II в SOPC Builder* – главе в *Настольной книге процессора Nios II*.

Nios II MMU является опциональным и взаимно независимым с Nios II MPU. Система Nios II может иметь либо MMU, либо MPU, но не может иметь одновременно MMU и MPU в одном процессорном ядре Nios II.

Элемент защиты памяти

Опциональный Nios II MPU предлагает следующие средства и функции:

- Защиту памяти
- До 32 регионов инструкций и 32 регионов данных
- Переменные размеры регионов инструкций и данных
- Величина памяти региона определяется по размеру или верхней границе адреса

-
- Права доступа к чтению и записи для регионов данных
 - Права доступа к исполнению для регионов инструкций
 - Перекрывающиеся регионы

За подробной информацией о реализации MPU, обратитесь к *Программной модели* – главе в *Настольной книге процессора Nios II*.

Вы можете опционально включить MPU, когда устанавливаете процессор Nios II в вашу аппаратную систему Nios II. Когда это сделано, MPU всегда разрешён. Доступны несколько параметров, позволяющих вам оптимизировать MPU в соответствии с системными требованиями.

За подробной информацией настраиваемых пользователем параметров Nios II MPU, в главе *Инсталляция процессора Nios II в SOPC Builder* – главе в *Настольной книге процессора Nios II*.

Nios II MPU является опциональным и взаимно независимым с Nios II MMU. Система Nios II может иметь либо MMU, либо MPU, но не может иметь одновременно MMU и MPU в одном процессорном ядре Nios II.