

Окончание, начало в №№ 2–4'2002

# Микроконтроллер для встроенного применения – NIOS. Конфигурация шины и периферии

Иосиф Каршенбойм

ik@lmail.loniis.ru

## 7. Параметры выравнивания адреса

### 7.1. Краткий обзор

Когда периферийное устройство с узкой шиной (например 16-битные данные) подключается к шинам мастера Avalon с широкими данными (например 32-битные данные), имеет место проблема выравнивания.

Эта проблема возникает для всех систем микропроцессоров, а не только для Avalon.

В частности, возникает вопрос: как мастер будет «видеть» старшие биты данных в цикле чтения от узкого периферийного устройства? Если не рассматривать детали, то есть два «больших» ответа на эту проблему:

- 1) Старшие биты заполняются нулями.

Когда хост производит чтение от узкого периферийного устройства, полезные данные появляются в младших разрядах регистра, а старшие разряды читаются как ноль.

- 2) Шины с «умной логикой».

Логика шины выполняет многократные последовательные операции чтения из узкого устройства «за спиной мастера» (то есть в то время, как для мастера установлен сигнал запроса ожидания) и собирает в единственный результат полной разрядности несколько индивидуальных результатов от чтения с меньшей разрядностью.

(Оба случая 1 и 2 имеют соответствующие способы операции записи, такие, как описаны ниже.)

Подход 1 называется выравниванием адреса «native». Подход 2 называется выравниванием адреса «dynamic» (динамическим выравниванием адреса).

Выравнивание адреса определено на основании «save-by-slave» в ptf-файле системы Avalon. Раздел MODULE/SYSTEM\_BUILDER\_INFO для каждого slave-устройства содержит назначение «Address\_Alignment», как описано в разделе 4.2.3.

Это назначение определяет, как логика шины Avalon обращается к устройству.

### 7.2. Назначения выравнивания адреса для включенных компонентов

Для любого компонента (как в Avalon в программе SOPC Builder software, так и определяемого пользователем) в программе SOPC Builder software может быть определена его собственная установка Address\_Alignment.

Некоторые из включенных в Avalon устройств — это периферийные устройства с управляемыми регистрами (например, UART или таймер).

Эти устройства определены как устройства с выравниванием адреса типа «native». Другие включенные в Avalon устройства — память разных типов (например ОЗУ/ПЗУ на кристалле и несколько определенных стандартных микросхем памяти).

Этим устройствам определяют выравнивание адреса типа «dynamic». Выравнивание «dynamic» применяется для любого устройства, которое используется как память для хранения кодов программы или данных.

Когда мастер читает или пишет команды или данные в устройство, он видит полный 32-битный (например) результат. Так, например, даже в системе, где 8-битная память связана с 32-битным мастером по выравниванию адреса типа «dynamic», мастер будет «видеть» устройство как сверхъестественно медленное 32-битное устройство (медленное, потому что каждый результат доступа мастера состоит из четырех последовательных обращений по четырем адресам). Логика шины «скрывает» узость устройства от программного обеспечения, выполняющегося на мастере. Программное обеспечение мастера значительно упрощается, потому что можно, не задумываясь, предполагать, что вся память имеет полную разрядность, и не нужны никакие специальные предосторожности, чтобы справиться с более узким устройством.

Выравнивание адреса типа «dynamic» позволяет, например, откомпилированному C-коду выполнять даже из узких внешних устройств типа Flash-памяти.

Всякий раз, когда программа выполняет 32-битную операцию доступа к данным, логика шины возвращает соответствующие четыре байта данных — независимо от того, как байты физически сохранены во внешнем устройстве.

Обратите внимание, что выравнивание адреса типа «dynamic» не всегда подходит для управления регистрами периферийных устройств. Операция записи, например, в регистр данных UART TX-Data не должна повлечь за собой последовательную операцию записи в регистр UART STATUS.

Следующие разделы описывают операцию записи и операцию чтения для параметров настройки Address\_Alignment «native» и «dynamic» как для 16-битного, так и для 32-битного мастера Avalon.

В этих примерах мастер — это Nios CPU, соответствующей разрядности «Data\_Width», но использо-

вание Nios в этих примерах никоим образом не ограничивается.

### 7.3. Выравнивание типа «Native» для узких периферийных устройств (V1.1, Nios-32)

#### 7.3.1. Периферийные устройства с разрядностью от 1 до 8 бит.

Представим гипотетическое узкое 8-битовое периферийное устройство, связанное с ядром Nios-32 CPU. Периферийное устройство для примера имеет, скажем, 5 внутренних регистров 8-битовой ширины с такими именами:

**Пример 1.** 8-битовое периферийное устройство с 5 регистрами.

Внутренний адрес периферийного устройства	8-битное имя регистра
0	aa
1	bb
2	cc
3	dd
4	ee

(Показано периферийное устройство на кристалле.)

Предположим, что это периферийное устройство было добавлено к Nios-системе с назначением SYSTEM\_BUILDER\_INFO/Address\_Alignment «native» для диапазона адресов данной периферии в некотором базовом адресе «BASE». Когда программа, выполняемая в Nios, читает данные от этого периферийного устройства (выполняет команду LD по адресу, в котором располагается это периферийное устройство), Nios всегда «получает» 32-битное значение в целевом регистре LD. Результат чтения от этого периферийного устройства после выравнивания «native»:

После LD из этого адреса	Программное обеспечение «видит» это 32-битное значение у получателя
BASE + 0x00	0x00.00.00.aa
BASE + 0x04	0x00.00.00.bb
BASE + 0x08	0x00.00.00.cc
BASE + 0x0C	0x00.00.00.dd
BASE + 0x10	0x00.00.00.ee

«Истинное» значение данных при чтении от любого native 8-битного (или более узкого) периферийного устройства будет появляться в младших значащих битах регистра назначения. Биты более старшего порядка, чем у периферии, не определены, если устройство находится на трехстабильной шине данных, или равны нулю, как показано в примере.

#### 7.3.2. Периферийные устройства с разрядностью от 9 до 16 бит

Представим гипотетическое 16-битное периферийное устройство, связанное с ядром Nios-32 CPU. Периферийное устройство, для примера, имеет 5 внутренних регистров разрядностью 16 бит с такими именами:

**Пример 2.** 16-битное периферийное устройство с 5 регистрами.

Внутренний адрес периферийного устройства	16-битное имя регистра
0	aaaa
1	bbbb
2	cccc
3	dddd
4	eeee

Вот как программное обеспечение «видит» содержание добавленного к Nios-системе устройства при выравнивании «native»:

После LD из этого адреса	Программное обеспечение «видит» это 32-битное значение у получателя
BASE + 0x00	0x00.00.aaaa
BASE + 0x04	0x00.00.bbbb
BASE + 0x08	0x00.00.cccc
BASE + 0x0C	0x00.00.dddd
BASE + 0x10	0x00.00.eeee

«Истинное» значение данных при чтении от любого native 16-битного (или более узкого, до 9 бит) периферийного устройства будет появляться в младших значащих битах регистра назначения. Биты более старшего порядка, чем у периферии, не определены, если устройство находится на трехстабильной шине данных, или равны нулю, как показано в примере.

#### 7.3.3. Периферийные устройства с разрядностью от 17 до 31 бит

Данные при чтении от периферийных устройств с выравниванием типа «native» и с разрядностью от 17 и до 31 бита (включительно), будут появляться в младших значащих битах регистра назначения.

Биты более старшего порядка, чем у периферии, будут заполнены нулями до 32 бит (если устройство не находится на трехстабильной шине данных, когда эти биты не определены).

### 7.4. Динамическое выравнивание для Узких Периферийных устройств (V1.1, Nios-32)

#### 7.4.1. Периферийные устройства с разрядностью от 1 до 8 бит

Рассмотрим 8-битное периферийное устройство, показанное в примере 1 с регистраторами a—e. Предположим, что это то же самое периферийное устройство, связанное с ядром Nios-32 и имеющее опцию выравнивания «dynamic». Опция «dynamic» заставляет программу SOPC Builder software добавлять специализированную логику к генерируемой Nios-системе (в модуле шины Avalon). Эта специализированная логика «останавливает» CPU всякий раз, когда он читает из периферийного устройства. Специализированная логика исполнит четыре операции чтения из четырех последовательно расположенных адресов внутри периферийного устройства, в то время как CPU находится в состоянии ожидания. Специализированная логика возвратит полное 32-битное значение данных, собранное от четырех последовательно расположенных байт в устройстве. Таким образом, программное обеспечение Nios «видит» это 8-битное периферийное устройство при динамически изменяемой разрядности как:

После LD из этого адреса	Программное обеспечение «видит» это 32-битное значение у получателя
BASE + 0x00	0xdd.cc.bb.aa
BASE + 0x04	0xUU.UU.UU.ee («UU» означает «неопределенный»)

Специализированная логика динамического переключения разрядности эффективно «конвертирует» периферийное устройство и представляет его для CPU так, как будто это 32-битное устройство.

Динамическая установка размеров шины прежде всего предназначено для узких устройств памяти, и это не рекомендуется для периферийных устройств на основе регистра.

Динамическая установка размеров шины может приводить к непредвиденным последствиям, когда она используется с периферийными устройствами, содержащими регистры. Например, нет никакой причины для того, чтобы делать динамическую установку ширины шины, если требуется только лишь прочитать один регистр «aa» в периферийном устройстве, как в примере, приведенном выше. Операция — ld по адресу BASE+0 приведет к динамической установке размеров шины, а это, в свою очередь, к выполнению операций чтения из регистров bb, cc, и dd. Некоторые периферийные устройства осуществляют свои внутренние функции при чтении их регистров. Когда используется динамическая установка размеров шины, такие периферийные устройства не могут работать, как им положено.

Логика динамической установки размеров шины также имеет значение для многократных последовательных операций записи в выбранное устройство. 8-битная операция записи (ST8x) затрагивает только один адрес и никакой «дополнительный» доступ в устройство не выполняется. Более широкие операции записи (ST16x или STx) выполняют так много операций записи, сколько требуется, чтобы передать данные в устройство. Таким образом, операция 32-битной записи в BASE+0 в приведенном выше примере записала бы данные по первым четырем адресам в выбранном устройстве.

Периферийные устройства с разрядностью более узкой, чем 8 бит, дополняются нулем до полного байта, затем обрабатываются логикой динамической установки размеров шины как описано выше (только если устройство не находится на трехстабильной шине данных, когда эти биты данных не определены).

#### 7.4.1.1. Шина, динамически устанавливающая размеры при выборке команд

Nios CPU может выполнять команды от 8-битного устройства памяти, которое подключено к системе с опцией динамической установки размеров шины.

Для 8-битных устройств шина всегда выбирает 32-битное значение, даже если в этом цикле выборки команды фактически используются только 16 бит! Поэтому не должен вызывать удивление тот факт, что программы, выполняющиеся из 8-битных устройств памяти, имеют тенденцию быть неэффективными.

#### 7.4.2. Периферийные устройства с разрядностью от 9 до 16 бит

Шина, динамически устанавливающая размеры для 16-битных устройств подобна шине с 8-битными устройствами, за исключением того, что в течение операции чтения (ld) производится два обращения (вместо четырех) к адресам, расположенным в выбранном устройстве. Рассмотрим периферийное устройство, показанное в примере 2, с 16-битными регистрами aaaa—eeee. Когда это периферийное устройство включено в систему Nios-32 с опцией выравнивания «dynamic», программное обеспечение «видит» это так:

После LD из этого адреса	Программное обеспечение «видит» это 32-битное значение у получателя
BASE + 0x00	0xbbbb.aaaa
BASE + 0x04	0xdddd.cccc
BASE + 0x08	0xUUUU.eeee («UU» означает «неопределенный»)

Периферия более узкая, чем 16 бит (но более широкая, чем 9 бит) заполняется нулями до половины слова (16 бит), затем обрабатывается логикой, динамически устанавливающей размеры, так как описано выше (только если устройство не находится на трехстабильной шине данных, когда эти биты данных не определены).

Логика, динамически устанавливающая размеры, пройдет через 8-и и 16-битовые операции записи (ST8x и ST16X) к соответствующему байту или половине слова для выбранного периферийного устройства. 32-битовые операции записи (STx) будут вызывать две последовательных операции записи к последовательным расположенным адресам внутри устройства.

Nios CPU может выполнять команды из 16-битного устройства памяти, которое подключено к системе с опцией динамической установки ширины.

Поведение логики шины с опцией динамической установки размеров различно для выборки данных (ld) и операций выборки команды. Все операции выборки данных передают 32 бита (2 полуслова) на CPU и требуют двух последовательных доступов к байтной памяти. С другой стороны, операции выборки команды, передают только одно полуслово в CPU. Логика шины с динамической установкой может «сообщить» через бит состояния «ifetch», что производит сейчас CPU: выборку команды или выборку данных. Логика установления динамических размеров шины выполнит только одну (а не две) операции чтения от 16-битного устройства в течение доступа выборки команды.

#### 7.4.3. Периферийные устройства с разрядностью от 17 до 31 бит

Опция Dynamic Bus Sizing предназначена прежде всего для устройств памяти. Устройства памяти обычно имеет разрядность 8, 16 и 32-бита. «Забавные» микросхемы памяти

с разрядностью, не кратной байту, настолько необычны, что спокойно могут игнорироваться. Однако поведение для таких устройств с шиной, динамически устанавливающей размеры, четкое. Чтение данных от выровненных словом периферийных устройств с разрядностью от 17 и до 31 бита (включительно) заполнено нулями к 32 битам (только если устройство не находится на трехстабильной шине данных, когда эти биты данных не определены).

#### 7.5. Шина, динамически устанавливающая размеры узких периферийных устройств (V1.1, Nios-16)

##### 7.5.1. Периферийные устройства с разрядностью от 1 до 8 бит

Рассмотрим то же самое 8-битное периферийное устройство, показанное в примере 1, с регистраторами a-e. Предположим, что это периферийное устройство связано с ядром Nios-16 с опцией выравнивания «dynamic». Динамическая устанавливающая размеры шина на Nios-16 работает точно так же, как динамическая устанавливающая размеры шина на Nios-32, за исключением того, что каждый доступ для чтения возвращает 16 (вместо 32) бит. Таким образом, программное обеспечение для Nios-16 «видит» это 8-битное периферийное устройство с динамически изменяющимся размером как:

Специализированная логика установления размеров динамической шины эффективно «конвертирует» периферийное устройство и подключает его к CPU так, как будто 16-битное устройство.

Логика шины с также играет свою роль при выполнении операций записи в выбранное устройство. Операции записи разрядностью 8 бит (ST8x) затрагивают только один адрес и никакой «дополнительный» доступ к устройству не выполняется. Чтобы передать в устройство дан-

После LD из этого адреса	Программное обеспечение «видит» это 32-битное значение у получателя
BASE + 0x00	0xbb.aa
BASE + 0x02	0xdd.cc
BASE + 0x04	0xUU.ee («UU» означает «неопределенный»)

ные разрядностью в половину слова, требуются более широкие операции записи. Операции

(STX) выполняют эти две операции записи так, как требуется. Таким образом, операция 16-битовой записи по адресу BASE+0 в примере (см. выше) записала бы данные по первым двум адресам в выбранном устройстве.

Периферийные устройства более узкие, чем 8 бит, заполняются нулями до полного байта и затем обрабатываются логикой динамической установки размеров шины, как описано выше (только если устройство не находится на трехстабильной шине данных, когда эти биты данных не определены).

## Выводы

Встроенный процессор Nios обеспечивает мощное и гибкое решение для разработчиков встроенных систем. Комбинация встроенного процессора с перестраиваемой конфигурацией периферийных устройств на кристалле, а также аппаратных и программных инструментальных средств развития обеспечивает законченное решение для разработок system-on-a-programmable-chip.

Однако имеются и недостатки. Шина с единственным мастером не позволяет создавать дополнительных каналов передачи данных в основную память и интегрировать их в единую систему. Невозможность же интеграции каналов DMA понижает производительность всего устройства.

## Литература

1. И. Каршенбойм, Н. Семенов. Микропрограммные автоматы на базе специализированных ИС // Chip News. 2000. № 7.
2. EDN от 07.20.2000. Processors drive (or dive) into programmable-logic devices. By Markus Levy.
3. И. Кривченко. Системы на кристалле: общее представление и тенденции развития // Компоненты и технологии. 2001. № 6.
4. Altera™. Nios Soft Core Embedded Processor. Data sheet. June 2000. Ver. 1.
5. Altera™. ARM-Based Embedded Processor Device Overview. Data sheet. February 2001. Ver. 1.2.
6. Altera™. Nios Tutorial.