



11. Ethernet and the NicheStack TCP/IP Stack - Nios II Edition

NII52013-11.0.0

11. Изернет и TCP/IP стек NicheStack - Версия под Nios II

TCP/IP стек NicheStack - Версия под Nios II является упрощённой реализацией пакета TCP/IP. Основным направлением реализации TCP/IP стека NicheStack является уменьшение используемых ресурсов при реализации полноценного TCP/IP стека. TCP/IP стек NicheStack разработан для использования во встраиваемых системах с малой памятью, выделяемой под код, что сделало возможным его использование в процессорных системах Nios II.

Altera предлагает TCP/IP стек NicheStack в качестве пакета программ, который вы сможете добавить к вашему пакету поддержки платы (BSP). Этот пакет доступен в Nios II SBT. TCP/IP стек NicheStack состоит из следующих средств:

- Протокол интернета (IP), включая пакетную передачу данных посредством нескольких сетевых протоколов;
- Протокол управляющих сообщений в интернете (ICMP) для поддержки и отладки сети;
- Протокол передачи дейтаграмм пользователя (UDP)
- Протокол управления передачей (TCP) с отслеживанием перегрузок, оценивание периода кругового обращения (RTT), и быстрого возобновления и повтора передачи;
- Протокол динамической конфигурации хоста (DHCP);
- Протокол разрешения адресов (ARP) для изернета;
- Стандартное гнездо (Socket) под интерфейс прикладного программирования (API).

В этой главе обсуждаются подробности использования NicheStack TCP/IP стека только для процессора Nios II. Эта глава состоит из следующих секций:

- "Предварительные условия для понимания устройства NicheStack TCP/IP стека" на стр. 11-2;
- "Введение в NicheStack TCP/IP стек - версия под Nios II" на стр. 11-2;
- "Другие TCP/IP стеки для процессора Nios II" на стр. 11-3;
- "Использование NicheStack TCP/IP стека - версия под Nios II" на стр. 11-3;
- "Конфигурирование NicheStack TCP/IP стека в программе Nios II" на стр. 11-9;
- "Информация для дальнейшей работы" на стр. 11-10;
- "Информация об ограничениях" на стр. 11-10.

Section III. Advanced Programming Topics.

11. Ethernet and the NicheStack TCP/IP Stack - Nios II Edition.

11. Изернет и TCP/IP стек NicheStack - Версия под Nios II. Перевод: Егоров А.В., 2012 г.

Предварительные условия для понимания устройства NicheStack TCP/IP стека

Чтобы вникнуть в информацию, излагаемую в этой главе, вы должны быть знакомы со следующими темами:

- Гнёзда (Sockets). Существует несколько книг на тему программирования с гнёздами. Приведём две хороших: [Программирование сетей UNIX](#) (Richard Stevens) и [Межсетевой обмен с помощью TCP/IP том 3](#) (Douglas Comer).
- Nios II EDS. Обратитесь к главе "[Введение](#)" в Настольной книге программиста Nios II за дополнительной информацией о Nios II EDS.
- MicroC/OS-II RTOS. Чтобы узнать подробнее об MicroC/OS-II, обратитесь к главе "[Операционная система реального времени MicroC/OS-II](#)" в Настольной книге программиста Nios II или к "[Использование MicroC/OS-II RTOS в процессорах NIOS II. Учебное руководство](#)".

Введение в NicheStack TCP/IP стек - версия под Nios II

Altera предлагает Nios II реализацию NicheStack TCP/IP стека, включая исходный код, в Nios II EDS. NicheStack TCP/IP стек предоставляет вам немедленный доступ к стеку Изернет соединения для процессора Nios II.

Реализация Altera NicheStack TCP/IP стека включает в себя API упаковщик, предоставляющий стандартное, хорошо задокументированное гнездо API.

NicheStack TCP/IP стек использует многопоточную среду MicroC/OS-II RTOS. Поэтому, для использования NicheStack TCP/IP стека с Nios II EDS, вы должны адаптировать ваш C++ проект под MicroC/OS-II RTOS. Процессорная система Nios II должна также иметь Изернет интерфейс или контроллер доступа к среде (MAC). Предлагаемый Altera NicheStack TCP/IP стек имеет драйвер, поддерживающий следующие две модели MAC:

- Устройство SMSC lan91c111,
- Altera® Triple Speed Ethernet MegaCore® функция (утроенная скорость Изернета).

Nios II EDS содержит аппаратную реализацию обоих MAC. Драйвер NicheStack TCP/IP стека иницирует прерывания, таким образом, вы сможете отслеживать прерывания от подключенного Изернет компонента.

Реализация Altera NicheStack TCP/IP стека основана на уровне аппаратной абстракции (HAL) обобщённой модели устройства Изернет. Для обобщённой модели устройства вы сможете написать новый драйвер, который будет поддерживать любой выбранный Ethernet MAC, и поддерживать совместимость HAL и гнёзд API для доступа к аппаратной части устройства.

За подробной информацией о написании драйвера Изернет устройства, обратитесь к главе "[Разработка драйверов устройств для слоя аппаратной абстракции \(HAL\)](#)" в Настольной книге программиста Nios II.

Section III. Advanced Programming Topics.

11. Ethernet and the NicheStack TCP/IP Stack - Nios II Edition.

11. Изернет и TCP/IP стек NicheStack - Версия под Nios II. Перевод: Егоров А.В., 2012 г.

Файлы и директории NicheStack TCP/IP стека

Вам не потребуется редактирование исходного кода NicheStack TCP/IP стека для использования этого стека в C/C++ программе под Nios II. Несмотря на это, Altera предоставляет вам исходный код для ознакомления. По умолчанию, эти файлы инсталлируются в директорию вместе с Nios II EDS *<Nios II EDS install path>/components/ altera_iniche/ UCOSII*. Ради краткости, в этой главе будем ссылаться на эту директорию как на *<iniche path>*.

Оригинальный код используется по возможности в директории *<iniche path>/src/downloads*. Подобная организация позволяет обновлять версию NicheStack TCP/IP стека. Директория *<iniche path>/src/downloads/packages* содержит оригинальный исходный код и документацию на NicheStack TCP/IP стек; директория *<iniche path>/src/downloads/30src* содержит специальный код для Nios II версии NicheStack TCP/IP стека, включая исходный код для поддержки MicroC/OS-II.

Справочное руководство доступно на странице [Literature: Nios II Processor](#) на сайте Altera под **Other Related Documentation**.

Реализация Altera NicheStack TCP/IP стека основана на версии 3.1 протокола стека, с необходимой упаковкой кода для интегрирования его в HAL.

Лицензирование

NicheStack TCP/IP стек - это TCP/IP протокол стека, созданный InterNiche Technologies, Inc. Вы сможете лицензировать NicheStack TCP/IP стек от Altera, перейдя по ссылке на [сайт Altera](#).

Вы сможете лицензировать другие протоколы стеков напрямую в InterNiche. Для этого, вам необходимо узнать подробности в InterNiche Technologies, Inc. (<http://www.iniche.com/>)

Другие TCP/IP стеки для процессора Nios II

Другие производители также предоставляют поддержку Изернет для процессора Nios II. В большей степени, другие производители RTOS зачастую предлагают Изернет модули под собственные оболочки RTOS.

Чтобы посмотреть последние данные о доступных продуктах других производителей, посетите страницу [Embedded Software](#) на сайте Altera.

Использование NicheStack TCP/IP стека - версия под Nios II

В этой секции описывается, как включить NicheStack TCP/IP стек в программу под Nios II.

Основным интерфейсом для NicheStack TCP/IP стека является интерфейс стандартного гнезда (socket). Также вы можете использовать следующие функции для инициализации стека и драйверов:

- alt_iniche_init()
- netmain()

Вы также можете использовать глобальную переменную **iniche_net_ready** в процессе инициализации.

Section III. Advanced Programming Topics.

11. Ethernet and the NicheStack TCP/IP Stack - Nios II Edition.

11. Изернет и TCP/IP стек NicheStack - Версия под Nios II. Перевод: Егоров А.В., 2012 г.

Вам необходимо использовать следующие простые функции, которые вызывает код системы HAL для получения MAC адреса и IP адреса:

- `get_mac_addr()`
- `get_ip_addr()`

Требования к системе Nios II

Для использования NicheStack TCP/IP стека, ваша система Nios II должна соответствовать следующим требованиям:

- Аппаратная часть системы должна иметь Изернет интерфейс с разрешёнными прерываниями.
- BSP должен быть основан на MicroC/OS-II.
- MicroC/OS-II RTOS должен быть сконфигурирован следующим образом:
 - TimeManagement / OSTimeTickHook должен быть разрешён.
 - Maximum Number of Tasks (максимальное количество задач) должен быть 4 и выше.
- Системный тактовый сигнал должен иметь значение, соответствующее аппаратному таймеру.

Задачи для NicheStack TCP/IP стека

NicheStack TCP/IP стек в своей стандартной для Nios II конфигурации выполняет две основные задачи. Каждая из этих задач потребляет ресурс потока MicroC/OS-II, параллельно с определённой памятью под стек потока. В дополнение к задачам, созданным вашей программой, непрерывно запущены следующие задачи:

- **The NicheStack main task**, `tk_netmain()` (главная задача NicheStack) - После инициализации, эта задача находится в режиме ожидания, пока новый пакет не доступен для обработки. Пакеты принимаются процедурой обработки прерываний (ISR). Когда ISR принимает пакет, она помещает его в очередь приёма и вызывает главную задачу.
- **The NicheStack tick task**, `tk_nettick()` (задача NicheStack по системному таймеру) - Эта задача периодически вызывается для мониторинга состояний превышения времени ожидания.

Эти задачи запускаются после успешного окончания процесса инициализации в функции `netmain()`, как это описано в разделе "`netmain()`".

Вы можете изменить приоритет задач и размер стека, используя директивы `#define` в файле конфигурации **ippport.h**. Вы можете создать дополнительный системные задачи, разрешив другие опции в NicheStack TCP/IP стеке путём редактирования **ippport.h** файла.

Инициализация стека

Перед тем, как проинициализировать стек, запустите планировщик MicroC/OS-II, вызвав `OSStart()` из `main()`. Выполните инициализацию стека, задав наивысший приоритет задачи, чтобы убедиться в том, что ваш код не пытается продолжить инициализацию, пока RTOS запускается, а I/O драйверы уже доступны.

Для инициализации стека, вызывайте функции `alt_niche_init()` и `netmain()`. Глобальная переменная `niche_net_ready` устанавливается в истину, когда инициализация стека завершена.

Section III. Advanced Programming Topics.

11. Ethernet and the NicheStack TCP/IP Stack - Nios II Edition.

11. Изернет и TCP/IP стек NicheStack - Версия под Nios II. Перевод: Егоров А.В., 2012 г.

Проследите за тем, чтобы ваш код не использовал интерфейс гнёзд (socket) прежде, чем `iniche_net_ready` не установится в истину. Например, вызовите `alt_iniche_init()` и `netmain()` в планировщике с наивысшим приоритетом, и подождите `iniche_net_ready`, прежде чем запускать другие задачи (как показано в примере 11-1).

alt_iniche_init()

Функция `alt_iniche_init()` инициализирует стек для использования в операционной системе MicroC/OS-II. Прототип функции следующий:

`void alt_iniche_init(void)`

Функция `alt_iniche_init()` ничего не возвращает и не имеет параметров.

netmain()

Функция `netmain()` ответственна за инициализацию и запуск задач NicheStack. Прототип функции следующий:

`void netmain(void)`

Функция `netmain()` ничего не возвращает и не имеет параметров.

iniche_net_ready

Когда NicheStack стек завершает инициализацию, он передаёт глобальной переменной `iniche_net_ready` ненулевое значение.

Не вызывайте никакие NicheStack API функций (кроме функций инициализации), пока `iniche_net_ready` не будет истина.

В примере 11-1 показано, как использовать `iniche_net_ready` для ожидания момента окончания инициализации сетевого стека.

Example 11-1. Instantiating the NicheStack TCP/IP Stack

```
void SSSInitialTask(void *task_data)
{
    INT8U error_code;

    alt_iniche_init();
    netmain();

    while (!iniche_net_ready)
        TK_SLEEP(1);

    /* Now that the stack is running, perform the application
       initialization steps */

    .
    .
    .
}
```

Макрос `TK_SLEEP()` является частью операционной системы (OS) NicheStack TCP/IP стека на уровне портов.

Section III. Advanced Programming Topics.

11. Ethernet and the NicheStack TCP/IP Stack - Nios II Edition.

11. Изернет и TCP/IP стек NicheStack - Версия под Nios II. Перевод: Егоров А.В., 2012 г.

get_mac_addr()* и *get_ip_addr()

Системный код NicheStack TCP/IP стека вызывает функции `get_mac_addr()` и `get_ip_addr()` во время инициализации устройства. Эти функции необходимы для системного кода, чтобы задать адреса MAC и IP для сетевого интерфейса, которые вы можете выбрать с помощью BSP настройки `altera_iniche.iniche_default_if`. Поскольку вы пишете эти функции сами, ваша система приобретает гибкость при сохранении MAC адреса и IP адреса в порядке размещения, это лучше, чем иметь фиксированный адрес, внесённый на аппаратном уровне в драйвере устройства. Например, некоторые системы могут сохранять MAC адрес во флеш памяти, тогда как другие могут хранить MAC адрес во внутри чиповой памяти.

Обе функции имеют в качестве параметров структуры устройства, используемые внутренними функциями NicheStack TCP/IP стека. Однако вам не нужно знать подробное описание этих структур. Вам необходимо знать только, как заполнить MAC и IP адреса.

Прототип `get_mac_addr()` следующий:

```
int get_mac_addr(NET net, unsigned char mac_addr[6]);
```

Вы должны использовать функцию `get_mac_addr()` для задания MAC адреса с помощью аргумента `mac_addr`. Оставьте аргумент `net` незаполненным. Прототип для функции `get_mac_addr()` находится в заголовочном файле `<iniche path>/inc/alt_iniche_dev.h`. Структура `NET` определена в `<iniche path>/src/downloads/30src/h/net.h` файле.

В примере 11-2 показана реализация функции `get_mac_addr()`. Этот пример приведён только в целях демонстрации, в нём MAC адрес сохраняется под адресом `CUSTOM_MAC_ADDR`. Это не вызывает ошибки в этом примере. В реальной программе, если это вызывает ошибку, то функция `get_mac_addr()` возвращает -1.

Example 11-2. An Implementation of *get_mac_addr()*

```
#include <alt_iniche_dev.h>
#include "includes.h"
#include "ipport.h"
#include "tcpport.h"
#include <io.h>
int get_mac_addr(NET net, unsigned char mac_addr[6])
{
    int ret_code = -1;

    /* Read the 6-byte MAC address from wherever it is stored */
    mac_addr[0] = IORD_8DIRECT(CUSTOM_MAC_ADDR, 4);
    mac_addr[1] = IORD_8DIRECT(CUSTOM_MAC_ADDR, 5);
    mac_addr[2] = IORD_8DIRECT(CUSTOM_MAC_ADDR, 6);
    mac_addr[3] = IORD_8DIRECT(CUSTOM_MAC_ADDR, 7);
    mac_addr[4] = IORD_8DIRECT(CUSTOM_MAC_ADDR, 8);
    mac_addr[5] = IORD_8DIRECT(CUSTOM_MAC_ADDR, 9);
    ret_code = ERR_OK;

    return ret_code;
}
```

Section III. Advanced Programming Topics.

11. Ethernet and the NicheStack TCP/IP Stack - Nios II Edition.

11. Изернет и TCP/IP стек NicheStack - Версия под Nios II. Перевод: Егоров А.В., 2012 г.

Вы должны использовать функцию `get_ip_addr()` для задания IP адреса стека протокола. Ваша программа может задать либо статичный адрес, либо запросить DHCP для поиска IP адреса. Прототип функции `get_ip_addr()` следующий:

```
int get_ip_addr(alt_iniche_dev* p_dev,
                ip_addr*      ipaddr,
                ip_addr*      netmask,
                ip_addr*      gw,
                int*          use_dhcp);
```

Функция `get_ip_addr()` задаёт и возвращает следующие параметры:

`IP4_ADDR(&ipaddr, IPADDR0,IPADDR1,IPADDR2,IPADDR3);`

`IP4_ADDR(&gw, GWADDR0,GWADDR1,GWADDR2,GWADDR3);`

`IP4_ADDR(&netmask, MSKADDR0,MSKADDR1,MSKADDR2,MSKADDR3);`

Для фиктивных переменных `IP_ADDR0-3`, замещается выражение для байтов 0 - 3 в IP адресе. Для `GWADDR0-3`, замещаются байты в адресе шлюза (gateway). Для `MSKADDR0-3`, замещаются байты маски сети. Например, следующее выражение задаёт `ip_addr` IP адресу 137.57.136.2:

`IP4_ADDR (ip_addr, 137, 57, 136, 2);`

Для разрешения DHCP, задайте строку:

`*use_dhcp = 1;`

NicheStack TCP/IP попытается получить IP адрес от сервера. Если сервер не предоставит IP адрес в течение 30 секунд, то время ожидания выходит и стек использует настройки по умолчанию, заданные при вызове функции `IP4_ADDR()`.

Для задания статичного IP адреса, задайте строку:

`*use_dhcp = 0;`

Прототип для функции `get_ip_addr()` находится в заголовочном файле `<iniche path>/inc/ alt_iniche_dev.h`.

В примере 11-3 показана реализация `get_ip_addr()` и представлен список необходимых файлов включения.

Пример 11-3 не проверяется на наличие ошибок. В реальной программе, вам вернётся -1 (ошибка).

`INICHE_DEFAULT_IF`, определено в **system.h**, идентифицирует сетевой интерфейс, который вы определяете на стадии генерирования системы. Вы можете проконтролировать `INICHE_DEFAULT_IF` с помощью BSP настройки **iniche_default_if**.

`DHCP_CLIENT`, также определено в **system.h**, задаёт возможность использования приложения DHCP client для получения IP адреса. Вы можете установить или сбросить эту настройку с помощью настройки **altera_iniche.dhcp_client**.

Вызов интерфейса гнезда

После того, как вы проинициализировали ваше Изернет устройство, используйте API гнезд (sockets API) в соответствии с вашей программой для доступа к IP стеку.

Section III. Advanced Programming Topics.

11. Ethernet and the NicheStack TCP/IP Stack - Nios II Edition.

11. Изернет и TCP/IP стек NicheStack - Версия под Nios II. Перевод: Егоров А.В., 2012 г.

Для создания новой задачи, которая обращается к IP стеку, используя API гнёзд, вы должны использовать функцию TK_NEWTASK(). Функция TK_NEWTASK() является частью уровня портов операционной системы (OS) NicheStack TCP/IP стека. TK_NEWTASK() вызывает функцию MicroC/OS-II OSTaskCreate() для создания потока и выполняет некоторые другие действия, относящиеся к NicheStack TCP/IP стеку. Прототип TK_NEWTASK() следующий:

```
int TK_NEWTASK(struct inet_task_info* nettask);
```

Example 11-3. An Implementation of get_ip_addr()

```
#include <alt_iniche_dev.h>
#include "includes.h"
#include "ipport.h"
#include "tcpport.h"
int get_ip_addr(alt_iniche_dev* p_dev,
               ip_addr* ipaddr,
               ip_addr* netmask,
               ip_addr* gw,
               int* use_dhcp)
{
    int ret_code = -1;
    /*
     * The name here is the device name defined in system.h
     */
    if (!strcmp(p_dev->name, "/dev/" INICHE_DEFAULT_IF))
    {
        /* The following is the default IP address if DHCP
         fails, or the static IP address if DHCP_CLIENT is
         undefined. */
        IP4_ADDR(&ipaddr, 10, 1, 1, 3);
        /* Assign the Default Gateway Address */
        IP4_ADDR(&gw, 10, 1, 1, 254);
        /* Assign the Netmask */
        IP4_ADDR(&netmask, 255, 255, 255, 0);

#ifdef DHCP_CLIENT
        *use_dhcp = 1;
#else
        *use_dhcp = 0;
#endif /* DHCP_CLIENT */

        ret_code = ERR_OK;
    }
    return ret_code;
}
```

Прототип определён в *<iniche path>/src/downloads/30src/nios2/osport.h*. Вы можете включить в проект этот заголовочный файл следующим образом:

```
#include "osport.h"
```

Вы можете посмотреть другие детали реализации уровня портов OS в файле **osport.c** в директории компонентов NicheStack TCP/IP стека, *<iniche path>/src/downloads/30src/nios2/*.

За дополнительной информацией о том, как использовать функцию TK_NEWTASK() в программе, обратитесь к учебному пособию "[Использование NicheStack TCP/IP стека - версия под Nios II](#)".

Section III. Advanced Programming Topics.

11. Ethernet and the NicheStack TCP/IP Stack - Nios II Edition.

11. Изернет и TCP/IP стек NicheStack - Версия под Nios II. Перевод: Егоров А.В., 2012 г.

Конфигурирование NicheStack TCP/IP стека в программе Nios II

NicheStack TCP/IP стек имеет некоторые настройки, которые вы можете сконфигурировать с помощью директивы **#define** в файле **ipport.h**. Nios II EDS позволяет вам сконфигурировать основные настройки (т.е. изменить директивы **#define** в файле **system.h**) без редактирования исходного кода. Большинство часто используемых опций доступны через настройки BSP, они идентифицируются через префикс **altera_iniche**.

За дополнительной информацией о настройках BSP для NicheStack, обратитесь к главе "[Nios II Software Build Tools Reference](#)" в Настольной книге программиста Nios II.

Некоторые менее часто используемые настройки не доступны через настройки BSP. Если вам потребуется изменить эти настройки, вы должны отредактировать **ipport.h** файл вручную.

Вы сможете найти **ipport.h** файл в директории **debug/system_description** вашего BSP проекта.

В следующих секциях описываются свойства, которые вы можете сконфигурировать в Nios II SBT. В обоих процессах разработки предлагаются значения по умолчанию для каждого свойства. Эти значения являются хорошей отправной точкой, и вы сможете завершить тонкую настройку значений в соответствие с потребностями вашей системы позднее.

Основные настройки NicheStack TCP/IP стека

Протоколы ARP, UDP и IP всегда разрешены. В табл. 11-1 показаны настройки протокола.

Табл. 11-1. Настройки протокола

Настройка	Описание
TCP	Разрешает и запрещает TCP.

В табл. 11-2 показаны глобальные настройки, влияющие на общую работу TCP/IP стека.

Табл. 11-2. Глобальные настройки

Настройка	Описание
Use DHCP to automatically assign IP address (Использование DHCP для автоматического задания IP адреса)	Если эта настройка включена, компонент использует DHCP для получения IP адреса.
Enable statistics (Разрешить статистику)	Если эта настройка включена, стек размещает счётчики принятых пакетов, ошибок и т.д. Счётчики определены в структуре mib в некоторых заголовочных файлах директории <iniche_path>/src/downloads/30src/h . За подробным описанием структуры mib обратитесь к документации на NicheStack.
MAC interface (MAC интерфейс)	Если IP стек имеет более одного сетевого интерфейса, этот параметр отображает используемый интерфейс. Обратитесь к главе "Информация об ограничениях" на стр. 11-10

Section III. Advanced Programming Topics.

11. Ethernet and the NicheStack TCP/IP Stack - Nios II Edition.

11. Изернет и TCP/IP стек NicheStack - Версия под Nios II. Перевод: Егоров А.В., 2012 г.

Настройки IP

В табл. 11-3 показаны настройки IP.

Табл. 11-3. Настройки IP

Настройка	Описание
Forward IP packets (Ускорение IP пакетов)	Если имеется более одного сетевого интерфейса, то эта настройка включена, тогда IP стек принимает пакеты одного интерфейса, не адресованные ему, и ускоряет вывод пакетов другого интерфейса. Обратитесь к главе "Информация об ограничениях" на стр. 11-10.
Reassemble IP packet fragments (Пересборка фрагментов IP пакетов)	Если эта опция включена, NicheStack TCP/IP стек пересобирает фрагменты IP пакетов в полный IP пакет. Иначе фрагменты IP пакетов отбраковываются. Этот раздел описан в книге Программирование сетей UNIX (Richard Stevens).

Настройки TCP

В табл. 11-4 показана настройка TCP zero copy (TCP нулевое копирование), которая доступна только, если включена настройка TCP.

Табл. 11-4. Настройка TCP

Настройка	Описание
Use TCP zero copy (Использовать TCP нулевое копирование)	Эта настройка разрешает NicheStack zero copy TCP API. Эта настройка позволяет вам устранить копирование из буфера в буфер при использовании NicheStack TCP/IP стека. За подробной информацией обратитесь к руководству NicheStack. Вы должны изменить ваш исходный код, для использования преимуществ zero copy API.

Информация для дальнейшей работы

За информацией для дальнейшей работы по реализации Altera NicheStack обратитесь к учебному пособию "[Использование NicheStack TCP/IP стека - версия под Nios II](#)". В учебном пособии представлены углублённые знания о NicheStack TCP/IP стеке, и проиллюстрировано его использование в сетевом приложении.

За подробной информацией о NicheStack обратитесь к руководству по NicheStack TCP/IP стека, доступному на странице [Literature: Nios II Processor](#) на сайте Altera под **Other Related Documentation**.

Информация об ограничениях

Несмотря на то, что код NicheStack содержит внутренние средства для поддержки нескольких сетевых интерфейсов, эти средства не тестировались в версии под Nios II. Обратитесь к руководству NicheStack TCP/IP стека и исходному коду для получения информации о поддержке нескольких сетевых интерфейсов.

Section III. Advanced Programming Topics.

11. Ethernet and the NicheStack TCP/IP Stack - Nios II Edition.

11. Изернет и TCP/IP стек NicheStack - Версия под Nios II. Перевод: Егоров А.В., 2012 г.

11. Изернет и TCP/IP стек NicheStack - Версия под Nios II	11-1
Предварительные условия для понимания устройства NicheStack TCP/IP стека	11-2
Введение в NicheStack TCP/IP стек - версия под Nios II.....	11-2
Файлы и директории NicheStack TCP/IP стека	11-3
Лицензирование	11-3
Другие TCP/IP стеки для процессора Nios II	11-3
Использование NicheStack TCP/IP стека - версия под Nios II	11-3
Требования к системе Nios II.....	11-4
Задачи для NicheStack TCP/IP стека	11-4
Инициализация стека.....	11-4
alt_iniche_init()	11-5
netmain()	11-5
iniche_net_ready	11-5
get_mac_addr() и get_ip_addr()	11-6
Вызов интерфейса гнезда	11-7
Конфигурирование NicheStack TCP/IP стека в программе Nios II.....	11-9
Основные настройки NicheStack TCP/IP стека.....	11-9
Настройки IP	11-10
Настройки TCP	11-10
Информация для дальнейшей работы	11-10
Информация об ограничениях	11-10