



6. Developing Programs Using the Hardware Abstraction Layer

NII52004-10.0.0

6. Разработка программ с использованием слоя аппаратной абстракции

Введение

В этой главе обсуждается, как разрабатывать программы для процессора Nios[®] II с использованием слоя аппаратной абстракции (HAL) Altera[®]. Эта глава состоит из следующих секций:

- "Структура проекта Nios II" на странице 6-2
- "Файл описания системы - system.h" на странице 6-4
- "Ширина данных и определение типов в HAL" на странице 6-5
- "Интерфейс в стиле UNIX" на странице 6-5
- "Файловая система" на странице 6-6
- "Использование устройств с символьным режимом" на странице 6-8
- "Использование файловой подсистемы" на странице 6-15
- "Использование устройств таймеров" на странице 6-16
- "Использование флеш устройств" на странице 6-19
- "Использование DMA устройств" на странице 6-25
- "Использование контроллеров прерываний" на странице 6-30
- "Уменьшение размера кода" на странице 6-30
- "Последовательность загрузки и точка входа" на странице 6-37
- "Использование памяти" на странице 6-39
- "Работа с исходными файлами HAL" на странице 6-44

Программный интерфейс приложения (API) для систем HAL быстро осваивается программистами, впервые знакомящимися с процессором Nios II. Программы, созданные на основе HAL, используют стандартные функции библиотеки ANSI Си и оперативные средства, а также доступ к аппаратным ресурсам через общие модели устройств HAL API. HAL API в значительной мере приспособило множество известных функций стандартной библиотеки ANSI Си, несмотря на то, что стандартная библиотека ANSI Си отделена от HAL. Близкая интеграция стандартной библиотеки ANSI Си и HAL делает возможным разработку полезных программ, которые никогда не вызывают напрямую функции HAL. Например, вы можете управлять устройствами с символьным режимом и файлами, используя I/O функции стандартной библиотеки ANSI Си, такие как printf() и scanf().

Этот документ не описывает стандартную библиотеку ANSI Си. Необходимое описание содержится в книге "Язык программирования Си, вторая редакция" Brian Kernighan и Dennis M. Ritchie (Prentice-Hall).

Пути разработки под Nios II

Пакет разработки встраиваемых систем на Nios II (EDS) предлагает два различных пути создания программ под Nios II. Вы можете использовать инструмент разработки программ под Nios II (SBT), или работать в интегрированной среде разработки Nios II (IDE). Эти два подхода по-своему используют HAL.

В большинстве случаев, вы можете создавать новые проекты, используя либо Nios II SBT на Eclipse™, либо командную строку SBT. IDE поддерживает следующие ситуации:

- работу с уже существующими в Nios II IDE программными проектами,
- создание новых проектов для компилятора Nios II C2H
- отладку в консоли FS2.

Настройки HAL BSP

Каждый пакет поддержки платы Nios II (BSP) содержит настройки, определяющие характеристики BSP. Например, HAL BSP содержит настройки, которые идентифицируют аппаратные компоненты, такие как stdout, ассоциированные со стандартными устройствами. Определение и управление настройками BSP – важная часть создания проекта Nios II. Вы можете управлять настройками BSP через редактор Nios II BSP, через опции командной строки или через Tcl скрипты.

За подробной информацией о контролировании настроек BSP, обратитесь к следующим документам:

- О Nios II SBT на Eclipse, к главе "[Начало работы с графическим интерфейсом пользователя](#)" в настольной книге программиста Nios II.
- О командной строке Nios II SBT, к главе "[Инструмент разработки программ Nios II](#)" в настольной книге программиста Nios II.

За подробным описанием настроек BSP, обратитесь к главе "[Инструмент разработки программ Nios II](#)" в настольной книге программиста Nios II.

Большинство настроек HAL отражаются в файле **system.h**, который представляет собой справочное пособие по вашему BSP. За информацией о файле **system.h**, обратитесь к секции "Файл описания системы - system.h" на странице 6-4.

Ни в коем случае не редактируйте файл **system.h**. Nios II EDS предоставляет вам инструменты для управления настройками системы.