5. Overview of the Hardware Abstraction Layer

NII52003-10.0.0

5. Общее представление о слое аппаратной абстракции

Введение

В этой главе вводится понятие слоя аппаратной абстракции (HAL) для процессора Nios® II. Эта глава состоит из следующих секций:

- "Начало работы" на странице 5-1
- "HAL архитектура" на странице 5-2
- "Поддерживаемая периферия" на странице 5-4

HAL – это облегчённая версия рабочей среды, предоставляющая простой драйвер интерфейса устройства под программы, для подключения к основному устройству. Программный интерфейс HAL приложения (API) интегрирован в стандартную библиотеку ANSI C. HAL API позволяет вам иметь доступ к устройствам и файлам, используя хорошо знакомые функции C библиотеки, такие как printf(), fopen(), fwrite() и т.д.

Сервис HAL в качестве набора драйверов устройства для процессорной системы Nios II предоставляет интерфейс, совместимый с периферией вашей системы. Тесная интеграция между SOPC Builder и инструментом разработки программы Nios II автоматизирует конструкцию элемента HAL для вашего устройства. После генерирования SOPC Builder аппаратной системы, инструмент создания программы Nios II (SBT) может генерировать собственный пакет поддержки платы (BSP), чтобы сделать аппаратную конфигурацию. Изменения в аппаратной конфигурации автоматически распространяются на конфигурацию драйвера HAL устройства, запрещая изменения в основном устройстве, чтобы не создавать ошибок.

Абстракция драйвера HAL устройства даёт ясное представление о разнице между программой приложения и программой драйвера устройства. Этот абстрактный драйвер стимулирует многократно используемый код приложения, который устойчив к изменениям в основном устройстве. Дополнительно, стандарт HAL создаёт его для прямого написания драйверов под новую аппаратную периферию, которая совместима с существующими драйверами периферии.

Начало работы

Простейший путь начать работу - это использовать HAL для создания программного проекта. В процессе создания нового проекта, вы также создаёте HAL BSP. Вам не нужно создавать или копировать HAL файлы, и не нужно редактировать какие-либо исходные файлы HAL. Nios II SBT сгенерирует HAL BSP для вас.

Чтобы поупражняться в создании простого программного проекта Nios II HAL, обратитесь к секции "Начало работы" в главе "Начало работы с графической оболочкой" в настольной книге программиста Nios II.

В командной строке Nios II SBT вы можете создать пример BSP, основанный на HAL, используя один скрипт **create-this-bsp** из Nios II Embedded Design Suite.

Вы должны приложить HAL к определённой системе SOPC Builder. Система SOPC Builder – это ядро процессора Nios II со встроенной периферией и памятью (всё это генерируется системой SOPC Builder). Если у вас нет собственной системы SOPC Builder, вы можете приложить ваш проект к примеру аппаратной системы, предлагаемой Altera. Обычно вы начинаете разработку вашего первого проекта, применительно к плате разработчика Altera®, а затем, переносите проект на собственную плату. Вы сможете в дальнейшем запросто изменить систему SOPC Builder.

Информация о создании нового проекта с помощью Nios II SBT содержится в главе "Начало работы с графической оболочкой" или в главе "Начало работы с из командной строки" в настольной книге программиста Nios II.

HAL архитектура

В этой секции описываются фундаментальные элементы HAL архитектуры.

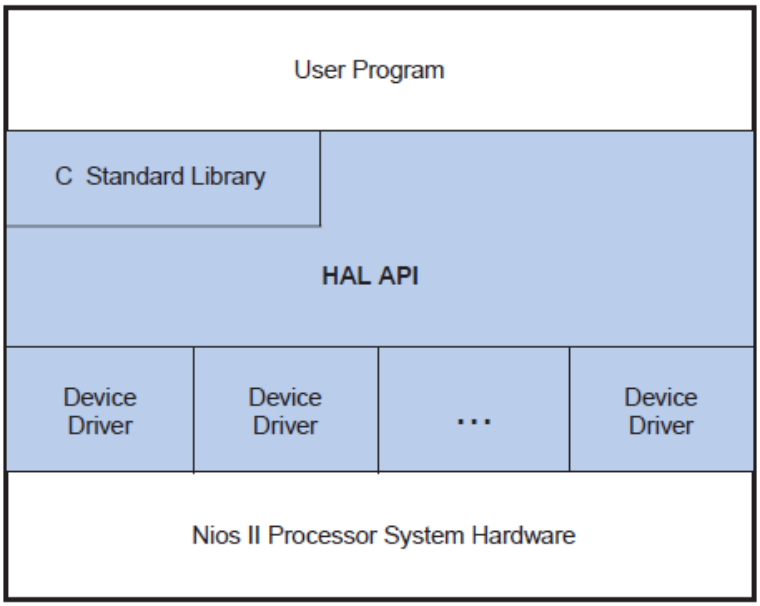
Сервисы

HAL предоставляет следующие сервисы:

- Интеграция со стандартной библиотекой newlib ANSI C – предоставление общеизвестных функций стандартной библиотеки.
- Драйверы устройств – предоставление доступа к каждому устройству в системе.
- HAL API – предоставление последовательного стандартного интерфейса с HAL сервисами, такими как доступ к устройствам, обработка прерываний и сигнальные средства.
- Инициализация системы – выполняет задачи инициализации для процессора и управления работой программы перед блоком main().
- Инициализация устройства – обрабатывает и инициализирует каждое устройство в системе перед запуском main().

На рис. 5-1 показаны слои HAL системы от аппаратного уровня до пользовательской программы.

Figure 5–1. The Layers of a HAL-Based System



Прикладная версия драйверов

Разработчики приложений несут ответственность за написание процедуры блока main() системы в числе других процедур. Приложения взаимодействуют с ресурсами системы либо через стандартную библиотеку Си, либо через HAL API. Разработчики драйверов устройств несут ответственность за то, чтобы сделать ресурсы устройства доступными для разработчиков приложений. Драйверы устройств напрямую связаны с аппаратными средствами через макрос низкоуровневого аппаратного доступа.

Подробнее о HAL содержится в следующих главах:

- "Разработка программ с использованием слой аппаратной абстракции" – главе в настольной книге программиста Nios II описывается, как использовать преимущества HAL при написании программ, не учитывая основные аппаратные средства.
- "Разработка драйверов устройств для слоя аппаратной абстракции" – главе в настольной книге программиста Nios II описывается, как напрямую связаться с аппаратными средствами и сделать аппаратные ресурсы видимыми в HAL API.

Общие модели устройств

HAL предлагает общие модели устройств для классов периферии, найденной во встроенной системе, таких как таймеры, чипы Ethernet MAC/PHY и I/O периферия, передающая знаковые данные. Общие модели устройств являются показателями производительности HAL. Они позволяют вам писать программы, используя совместимость API, не учитывая основные аппаратные средства.

Классы модели устройства

HAL предлагает модели следующих классов устройств:

- Устройства с символьным режимом – аппаратная периферия, которая посылает и / или принимает последовательные символы, например UART.
- Устройства таймеры - аппаратная периферия, которая подсчитывает тактовые импульсы и может генерировать периодический запрос прерывания.
- Файловая подсистема – механизм доступа к файлам в физическом устройстве. В зависимости от внутренней реализации, драйвер файловой подсистемы должен иметь прямой доступ к основным аппаратным средствам или использовать отдельный драйвер устройства. Например, вы можете написать драйвер файловой подсистемы, который получает доступ к флеш-памяти, используя HAL API для чипов флеш-памяти.
- Устройства Ethernet – устройства, которые предоставляют доступ к Ethernet подключению для сетевого стека, такие как предлагаемые Altera NicheStack® TCP/IP Stack - Nios II Edition. Вам необходим сетевой стек для использования ethernet устройства.
- Устройства прямого доступа к памяти (DMA) – периферия, которая выполняет передачу большого массива данных от источника до приёмника данных. Источник и приёмник может быть памятью или другим устройством, таким как Ethernet подключение.
- Устройства флеш-памяти – энергонезависимые устройства памяти, использующие специальный программный протокол для сохранения данных.

Преимущества разработчиков приложений

HAL определяет набор функций, который вы используете для инициализации и доступа к каждому классу устройства. API является совместимой и независимой от аппаратной реализации устройства. Например, для доступа к устройствам с символьным режимом и к файловой подсистеме вы можете использовать функции стандартной библиотеки Си, такие как printf() и fopen(). Для разработчиков приложений, вам не потребуется писать низкоуровневые процедуры для адаптации основных связей с аппаратной частью этих классов периферии.

Преимущества разработчиков драйверов устройств

Каждая модель устройства определяет набор функций драйверов, необходимый для манипуляции над собственным классом устройства. Если вы пишете драйверы для новой периферии, вам нужно только предоставить этот набор функций драйверов. В результате, ваша задача разработки драйвера предопределена и хорошо задокументирована. Дополнительно, вы можете использовать существующие HAL функции и приложения для доступа к устройству, которые сохраняют работу по разработке программы. HAL вызывает функции драйвера для доступа к аппаратным средствам. Разработчики приложений вызывают ANSI C или HAL API для доступа к аппаратным средствам, а не вызывают напрямую вашу процедуру драйвера. Поэтому использование вашего драйвера должно быть задокументировано как часть HAL API.

Стандартная библиотека Си – Newlib

HAL интегрировал стандартную библиотеку ANSI Си в свои оперативные средства. HAL использует newlib в качестве реализации стандартной библиотеки Си с открытым исходным кодом. Newlib - это библиотека Си для использования во встроенных системах, идеально подходящая для использования с HAL и процессором Nios II. Лицензия newlib не требует от вас выпуска вашего исходного кода или платы за проекты, основанные на библиотеке newlib.

Стандартная библиотека ANSI Си уже задокументирована. Наверное главное справочное пособие – это "Язык программирования Си" B. Kernighan и D. Ritchie, опубликованное Prentice Hall и доступное на 20 языках. Redhat также предоставляет он-лайн документацию на newlib по адресу: http://sources.redhat.com/newlib.

Поддерживаемые аппаратные средства

В этой секции описывается Nios II HAL поддержка аппаратных средств Nios II.

Поддержка ядра процессора Nios II

Nios II HAL поддерживает все доступные реализации ядра Nios II.

Поддержка периферии

Altera предлагает обширную периферию для использования с процессорными системами Nios II. Большинство периферии Altera имеют HAL драйверы устройств, которые предоставляют вам доступ к аппаратным средствам через HAL API. Следующая периферия Altera предоставляет полную поддержку HAL:

- Устройства с символьным режимом
  - Ядро UART
  - Ядро JTAG UART
  - Контроллер дисплея LCD 16207

- Устройства флеш-памяти
  - Общий флеш интерфейс совместимости флеш чипов
  - Контроллер последовательной конфигурации чипа (EPCS) Altera
- Файловая подсистема
  - Файловая подсистема Altera основанная на хосте
  - Файловая подсистема Altera zip только для чтения
- Устройства таймеры
  - Ядро таймера
- Устройства DMA
  - Ядро контроллера DMA
  - Ядро контроллера DMA расброс-сборка
- Устройства Ethernet
  - Функция тройной скорости Ethernet MegaCore®
  - Контроллер LAN91C111 Ethernet MAC/PHY

Компонентам LAN91C111 и тройная скорость Ethernet потребуются оперативные средства MicroC/OS-II.

За дополнительной информацией обратитесь к главе "Стек NicheStack TCP/IP - Nios II Edition" в настольной книге разработчика программ Nios II. Здесь отсутствует периферия, поставляемая сторонними разработчиками. За списком другой периферии для процессора Nios II обратитесь к странице Embedded Software на веб-сайте Altera.

Вся периферия (и Altera и сторонних разработчиков) должна иметь заголовочный файл, который определяет низкоуровневый интерфейс с аппаратной частью. Поэтому вся периферия в некоторой степени поддерживает HAL. Однако некоторая периферия может не иметь драйверов устройств. Если драйверы не доступны, используйте только описания в заголовочных файлах для доступа к устройству. Не используйте безымянные постоянные, такие как жёстко запрограммированный адрес, для доступа к периферии.

Определённая периферия неизбежно будет иметь специальные аппаратные средства, использование которых не описано в API широкого применения. HAL обрабатывает специфические требования устройств через функцию стиля UNIX ioctl(). Поскольку аппаратные средства зависят от периферии, опция ioctl() документирована в описании для каждой периферии.

Некоторая периферия предоставляет специальную функцию accessor (средство доступа), которой нет в общих моделях устройств HAL. Например, Altera поставляет ядро широкого применения параллельного I/O (PIO) для использования с процессорной системой Nios II. Периферия PIO не может компоноваться в каком-нибудь классе общих моделей устройств HAL, она даёт только заголовочный файл и несколько специальных функций accessor.

За подробной информацией о соответствующей программной поддержке периферии, обратитесь к описанию периферии. За подробной информацией о предлагаемой Altera периферии, обратитесь к руководству пользователя по IP встроенной периферии.

Поддержка MPU

HAL не имеет явной поддержки устройства опциональный элемент защиты памяти (MPU). Однако он поддерживает систему расширенной обработки исключений, которая обрабатывает исключения Nios II MPU.

За подробной информацией об обработке MPU и прочих расширенных исключений, обратитесь к главе "Обработка исключений" в настольной книге программиста Nios II. За подробной информацией о реализации устройства MPU обратитесь к главе "Программная модель" в настольной книге процессора Nios II.

Поддержка MMU

HAL не поддерживает устройство опциональный элемент менеджера памяти (MMU). Чтобы использовать MMU, вам необходимо реализовать полнофункциональную операционную систему.

За подробной информацией о Nios II MMU, обратитесь к главе "Программная модель" в настольной книге процессора Nios II.

Оглавление