Последовательность загрузки и точка входа

Обычно точка входа вашей программы – это функция main(). Есть другая точка входа - alt_main() – которую вы можете использовать для усиления контроля над последовательностью загрузки. Разница между входом через main() и входом через alt_main() - это отличие между ведущей (hosted) прикладной системой и автономной (free-standing) прикладной системой.

Ведущая версия автономных приложений

Стандарт ANSI Си определяет ведущими приложениями только те, которые вызывают main() для начала исполнения. С запуском main() ведущее приложение предполагает наличие среды прогона и то, что все системные сервисы инициализированы и готовы к использованию. Это правило справедливо для среды HAL. Если вы начинающий программист Nios II, ведущая среда HAL помогает вам быстро освоиться, поскольку вам учитывать, какие устройства существуют в системе или как инициализировать каждое. HAL инициализирует всю систему.

Стандарт ANSI Си также предлагает вам другую точку входа, которая препятствует автоматической инициализации и допускает программисту Nios II явно инициализировать только востребованные аппаратные средства. Функция alt_main() предлагает автономную среду, предоставляя вам полный контроль над инициализацией системы. Автономная среда является зоной ответственности программиста, она определяет инициализацию только используемых в программе системных средств. Например, вызов printf() не будет корректно работать в автономной среде, пока alt_main() не инсталлирует драйвер устройства с символьным режимом и не перенаправит stdout на устройство.

Использование автономной среды увеличивает сложность в написании программ Nios II, поскольку вы берёте ответственность за инициализацию системы. Если ваш главный интерес – уменьшение размера кода, используйте советы, описанные в секции "Уменьшение размера кода" на странице 6-30. Проще уменьшить размер кода HAL BSP, используя настройки BSP, чем используя автономный режим.

Nios II EDS предлагает примеры ведущей (hosted) и автономной (free-standing) программ.

Последовательность загрузки программ на основе HAL

HAL предлагает код инициализации системы в библиотеки Си программ этапа исполнения (**crt0.S**). Этот код исполняет следующую последовательность загрузки:

- Выключает кэш инструкций и данных
- Конфигурирует указатель стека
- Конфигурирует глобальный указатель регистра
- Инициализирует блок, начинающийся с символа (BSS) региона до нулей, используя специальные символы компилятора __bss_start и __bss_end. Это указатели начала и конца BSS региона.
- Если в системе нет загрузчика, копирует в RAM только секцию компилятора, которая запускается из адреса в RAM, например .rwdata, .rodata и .exceptions. Обратитесь к секции "Регистр глобального указателя" на странице 6-42.
- Вызывает alt_main().

HAL предлагает реализацию по умолчанию функции alt_main(), выполняющую следующие шаги:

- Вызов функции alt_irq_init(), размещённой в **alt_sys_init.c.** Функция alt_irq_init() **инициализирует аппаратный контроллер прерываний**. Процесс разработки Nios II создаёт **alt_sys_init.c** для каждого HAL BSP.
- Вызов функции ALT_OS_INIT() для выполнения специфической инициализации операционной системы. Для систем, не включающих в семя планировщик операционной системы (OS), этот макрос не оказывает эффекта.
- Если вы используете HAL с операционной системой, инициализируйте семафор (одно из средств синхронизации в многопоточных процессах, используемое для управления доступом к разделяемыми ресурсами) alt_fd_list_lock, который контролирует доступ к файловым системам HAL.
- Разрешение прерываний.
- Вызов функции alt_sys_init(), также находящейся в **alt_sys_init.c.** Функция alt_sys_init() инициализирует все драйверы устройств и пакет программ в системе.
- Перенаправление стандартных Си I/O каналов (stdin, stdout и stderr) для их использования в соответствующих устройствах.
- Вызов конструкций C++, используя функцию _do_ctors().
- Регистрация деструкторов C++, вызываемых при выключении системы.
- Вызов main().
- Вызов exit(), предоставление кода возврата из main() в качестве входного аргумента для exit().

Файл **alt_main.c**, инсталлируемый вместе с Nios II EDS, предлагает реализацию по умолчанию. SBT копирует **alt_main.c** в директорию с вашим BSP.

Настройка последовательности загрузки

Вы можете создать собственную последовательность загрузки, просто определив в вашем Nios II проекте функцию alt_main(). Она предоставляет вам полный контроля над последовательностью загрузки и позволяет вам выборочно разрешать сервисы HAL. Если ваше приложение использует точку входа alt_main(), вы можете скопировать в него реализацию по умолчанию в качестве отправной точки и приступить к настройке под ваши нужды.

Функция alt_main() вызывает функцию main(). После возврата main(), функция alt_main() по умолчанию зацикливается. С другой стороны, ваша собственная функция alt_main() может завершиться вызовом exit(). Не используйте оператор **return** (возврат).

Прототипом для alt_main() является:

void alt_main (void)

HAL собирает среду, включающую в себя механизмы для подмены кода HAL BSP по умолчанию. Этим допускается подмена загрузчика, то есть драйверов устройств по умолчанию и прочего системного кода, вашей собственной реализацией.

Файл **alt_sys_init.c** является генерируемым файлом, который вы не должны модифицировать. Однако Nios II SBT позволяет вам контролировать генерируемое содержимое файла **alt_sys_init.c**. Чтобы задать последовательность инициализации в **alt_sys_init.c**, вы манипулируете свойствами auto_initialize и alt_sys_init_priority в каждом драйвере, используя Tcl команду set_sw_property.

За подробной информацией о генерируемых файлах и как контролировать содержимое **alt_sys_init.c**, обратитесь к главе "Инструмент разработки программы Nios II" в настольной книге программиста Nios II. За подробной информацией о файле **alt_sys_init.c**, обратитесь к главе "Разработка драйверов устройств для слоя аппаратной абстракции" в настольной книге программиста Nios II. За подробной информацией о Tcl команде set_sw_property, обратитесь к главе "Справка по инструменту разработки программы Nios II" в настольной книге программиста Nios II.