Создание собственного драйвера устройства для HAL

В этой секции описывается, как подготовить соответствующие файлы для интегрирования драйвера вашего устройства в HAL. В секции "Интегрирование драйвера устройства в HAL" на стр. 7-17 описано правильное расположение файлов.

Заголовочные файлы и alt_sys_init.c

Сердцем HAL является авто генерируемый исходный файл **alt_sys_init.c**. Этот файл содержит исходный код, который использует HAL для инициализации драйверов устройств для всех поддерживаемых устройств в системе. Особенно в этом файле определяется функция alt_sys_init(), которая вызывается перед main() для инициализации пакета программ драйверов устройств, чтобы сделать их доступными для программы.

Когда вы создаёте драйвер или пакет программ, вы задаёте в Tcl скрипте, как вы хотите, чтобы функция alt_sys_init() вызывала ваши макросы INSTANCE и INIT. Обратитесь к секции "Разрешение программы инициализации" на стр. 7-24 за подробностями. В примере 7-4 показан отрывок из **alt_sys_init.c** файла.

В оставшейся части этой секции подразумевается, что вы знакомы с механизмом инициализации alt_sys_init() HAL.

Инструменты сборки программы (SBT) создают **alt_sys_init.c,** основанный на заголовочных файлах, ассоциированных с каждым драйвером устройства и пакетом программ. Для драйвера устройства, заголовочный файл должен определять макросы <*component name*>*_INSTANCE* и <*component name*>*_INIT*.

Аналогично драйверу устройства, пакет программ предлагает макрос INSTANCE, который однократно вызывает alt_sys_init(). Заголовочный файл пакета программ может дополнительно предлагать макрос INIT.

Example 7–4. Excerpt from an alt_sys_init.c File Performing Driver Initialization

```
#include "system.h"
#include "sys/alt_sys_init.h"

/*
* device headers
*/
#include "altera_avalon_timer.h"
#include "altera_avalon_uart.h"

/*
* Allocate the device storage
*/
ALTERA_AVALON_UART_INSTANCE( UART1, uart1 );
ALTERA_AVALON_TIMER_INSTANCE( SYSCLK, sysclk );

/*
* Initialize the devices
*/
void alt_sys_init( void )
{
    ALTERA_AVALON_UART_INIT( UART1, uart1 );
    ALTERA_AVALON_TIMER_INIT( SYSCLK, sysclk );
}
```

Например, **altera_avalon_jtag_uart.h** должен определять макрос ALTERA_AVALON_JTAG_UART_INSTANCE и ALTERA_AVALON_JTAG_UART_INIT. Назначение этих макросов следующее:

- Макрос *_INSTANCE выполняет любую необходимую локализацию статической памяти. Для драйверов, *_INSTANCE вызывается один раз для каждого устройства, так чтобы память могла быть инициализирована по базовым адресам каждого устройства. Для пакета программ, *_INSTANCE вызывается однократно.
- Макрос *_INIT выполняет инициализацию драйвера устройства или пакета программ во время прогона.

В случае с драйвером устройства, оба макроса дают два входных аргумента:

- Первый аргумент, name, это имя элемента устройства прописными буквами.
- Второй аргумент, dev, это версия имени устройства строчными буквами. Аргумент, dev, это имя, даваемое компоненту в SOPC Builder на стадии генерации системы.

Вы можете использовать эти входные параметры для выделения специфической информации об устройстве из файла **system.h**.

Имя заголовочного файла должно быть следующим:

- Драйвер устройства: <*hardware component class*>**.h**. Например, если ваш драйвер предназначен для компонента **altera_avalon_uart**, имя файла должно быть **altera_avalon_uart.h**.
- Пакет программ: <*package name*>**.h**. Например, если вы создаёте пакет программ с помощью следующей команды:

create_sw_package my_sw_package

заголовочный файл называется **my_sw_package.h**.

За полными примерами обратитесь к любому поставляемому Altera драйверу устройств, например к драйверу JTAG UART в <*Altera installation*>**/ip/sopc_builder_ip/ altera_avalon_jtag_uart.**

Чтобы оптимизировать время сборки проекта, не включайте заголовки периферии в **system.h**. Их нужно включать в **alt_sys_init.c**.

Исходный код драйвера устройства

В дополнении к заголовочному файлу, драйверу компонента, для объединения с BSP, может понадобиться совместимый исходный код. Этот исходный код специфический для каждого компонента, он может состоять из одного или нескольких Си файлов (или файлов на языке Ассемблера).