Scan Engineering Telecom SPb

Загрузчик MBL

Руководство пользователя

Версия 1.0

Код документа: UG-FPGA-00-MBL
Дата сборки: 28 августа 2015 г.
Листов в документе: 45

http://www.setdsp.ru

Содержание

Перечень рисунков

Перечень процедур

Перечень сокращений и условных обозначений

ANSI	American National Standards Institute	5
BRAM	Block Random Access Memory	7–10, 13, 21, 33
BSP	Board Support Package	3, 4, 8, 16–18, 37
DDR	Double Data Rate	6–10, 42, 43
ELF	Executable and Linkable Format	3, 8, 13, 33, 34, 38
FPGA	Field-Programmable Gate Array	3, 7–10, 12, 13, 18, 21, 22, 24, 29–33, 35–39, 42, 43
IP	Intellectual Property	7
JTAG	Joint Test Action Group	2–4, 12, 21, 22, 24, 30, 35, 38–40
MBL	MicroBlaze Boot Loader	2–4, 6–10, 12, 13, 18–24, 27, 32, 33, 35, 37
PCIe	PCI Express	9
PCI	Peripheral Component Interconnect	5
SDK	Software Development Kit	3, 4, 8, 10, 12–24, 33–44
SDRAM	Synchronous Dynamic Random Access Memory	6–10, 21, 33, 42, 43
SPI	Serial Peripheral Interface	2, 4, 10, 33, 37, 38, 41, 43
SREC	Motorola S-record format	2–4, 8–10, 21, 33–38, 41–43
USB	Universal Serial Bus	21
VITA	VME International Trade Association	5
VPX	System specification ANSI/VITA 46	10
ЗАО	Закрытое Акционерное Общество	7, 8, 10, 12, 24, 33

Общие сведения

Данный документ описывает программный проект, который носит название «MicroBlaze Boot Loader» (сокращенно MBL). Проект реализует исполняемый микропроцессором MicroBlaze загрузчик, который обеспечивает процесс инициализации подключенной к микропроцессору памяти типа SDRAM DDR содержимым двоичного образа приложения пользователя, сохраняемого в энергонезависимой памяти типа флеш, и последующий запуск на исполнение микропроцессором MicroBlaze этого приложения пользователя.

1 Введение

Использование микропроцессорного IP-ядра MicroBlaze в качестве основы микропроцессорной системы, реализуемой в аппаратной платформе для FPGA модулей производства ЗАО «Скан Инжиниринг Телеком», предполагает последующую разработку для этой системы приложения пользователя. В результате разработки приложения пользователя получается исполняемый двоичный образ, содержащий набор инструкций, предназначенных для исполнения микропроцессорным ядром MicroBlaze.

При работе с микропроцессорной системой на базе MicroBlaze, содержимое любого исполняемого двоичного образа необходимо размещать в областях памяти, в которых разрешенно исполнение процессором его инструкций. Как правило, это блоки энергозависимой памяти BRAM или SDRAM DDR.

Конечный объём любого двоичного образа определяется объёмом исходного кода приложения пользователя, написанного на языках «Си» или «ASM», компилируемого в двоичный образ. У каждого пользователя объём получаемого двоичного образа будет разный и для размещения его содержимого в виде исполняемых инструкций и данных необходимо использовать разные по объёму области памяти. В FPGA микросхемах объёмы блоков памяти BRAM всегда ограничен и их выделяют для размещения содержимого двоичного образа в объёмах не более десятков или сотен Кбайт. В противовес BRAM, объёмы блоков SDRAM DDR измеряются сотнями Мбайтов или единицами Гбайтов и в них можно размещать большое по объёму содержимое двоичных образов.

В функционально законченных решениях пользователя, построенных на базе FPGA модулей, помимо размещения содержимого двоичного образа в энергозависимых блоках BRAM или SDRAM DDR необходимо хранить сам двоичный образ приложения пользователя в энергонезависимой памяти, чтобы обеспечить его загрузку в энергозависимые блоки памяти после подачи питания на FPGA модуль. Физическое размещение энергонезависимой памяти для хранения двоичного образа может быть непосредственно на той же печатной плате FPGA модуля, где находится микросхема FPGA, а может быть на внешнем, по отношению к печатной плате, носителе. Как правило, размещаемый на печатной плате FPGA модуля банк энергонезависимой памяти представляет собой параллельную или последовательную по типу доступа флеш-память.

Чтобы обеспечить исполнение ядром MicroBlaze инструкций из любого двоичного образа,сохраненного в энергозависимых блоках памяти, после подачи питания FPGA микросхеме необходимо иметь систему, позволяющую переносить двоичный образ из энергонезависимых блоков памяти в энергозависимые, и обеспечить передачу управления микропроцессорным ядром инструкций и состава этого образа.

Устройство системы конфигурирования FPGA микросхемы Xilinx после подачи на нее питания предполагает автоматическую загрузку конфигурации в виде аппаратной платформы в FPGA микросхему из подключенной к ней флеш-памяти. В терминологии Xilinx такая флеш-память называется «Platform Flash». Данная загрузка предполагает возможность инициализации содержимого внутренних блоков памяти BRAM, что позволяет сохранять в «Platform Flash» двоичный образ с набором исполняемых инструкций для ядра MicroBlaze. По окончанию автоматической загрузки и инициализации блока BRAM содержимым сохраняемого в «Platform Flash» двоичного образа, управление ядром MicroBlaze передается инструкциям из этого блока BRAM.

Инициализировать содержимое блоков памяти SDRAM DDR автоматической системой конфигурации FPGA микросхем Xilinx не представляется возможным, так как эта система просто не умеет этого делать.

Если у пользователя, при разработке своего приложения для микропроцессора MicroBlaze, получается двоичный образ, объём которого превышает возможности предоставленных в распоряжение пользователя для размещения содержимого исполняемого образа блоков BRAM, то у пользователя возникает необходимость использовать для размещения содержимого образа блоки памяти большего размера. В качестве таких блоков выступают блоки памяти SDRAM DDR. Чтобы осуществить размещение исполняемого образа в SDRAM DDR после подачи питания на FPGA микросхему необходимо решение, которое предоставляет такую возможность. Состоит такое решение из двух частей: аппаратной и программной.

Основой аппаратной части является инфраструктура IP-ядер «Микропроцессорная система на MicroBlaze», описанная в документе [1]. Эта платформа построенна на базе микропроцессорного ядра MicroBlaze, содержит в себе контроллеры, обеспечивающие доступ к флеш-памяти и блокам памяти BRAM и SDRAM DDR.

Основой для программной части служит исполняемый MicroBlaze образ загрузчика MBL, размещаемый во внутреннем блоке BRAM. Данный образ загружается автоматически в блок BRAM после подачи питания на FPGA микросхему из «Platform Flash» и берет на себя управление работой микропроцессорного ядра MicroBlaze.

Когда управление микропроцессорным ядром MicroBlaze переходит загрузчику MBL, загрузчик осуществляет инициализацию блока памяти SDRAM DDR содержимым образа, сохраняемого во флеш-памяти, и передает управление микропроцессорным ядром MicroBlaze инструкциям этого образа, размещенным в SDRAM DDR в процессе инициализации.

2 Описание проекта

Проект загрузчика MBL реализуется в среде разработки Xilinx SDK. Исходный код проекта написан на языке «Си». Получаемый в результате компиляции проекта двоичный образ предназначен для исполнения микропроцессорным ядром MicroBlaze.

Микропроцессорное ядро MicroBlaze включается в состав микропроцессорной системы, назначение составных частей которой описаны в документе [1].

Проект загрузчика MBL является демонстрационным проектом и может входить в состав различных демонстрационных пакетов BSP, предназначенных для различных FPGA модулей производства ЗАО «Скан Инжиниринг Телеком». Проект MBL поставляется с текстами программ в открытом исходном виде.

Разработчики компании ЗАО «Скан Инжиниринг Телеком» рекомендуют использовать проект MBL в составе решений пользователя, в которых существует необходимость работы приложений пользователя, написанных для ядра MicroBlaze, в блоках памяти SDRAM DDR.

2.1 Основа для реализации

В основе реализации загрузчика MBL лежит работа с форматом хранения двоичных данных в текстовом виде, используя кодировку ASCII. Формат носит название «Motorola S-record» (сокращенно SREC).

Пользователь должен преобразовать двоичный образ своего приложения, получаемый в процессе компиляции собственного программного проекта, в файл формата SREC. В среде разработки Xilinx SDK существует утилита, которая преобразует полученный в процессе компиляции двоичный образ, сохраняемый в формате ELF, в текстовый файл формата SREC, сохраняемый в кодировке ASCII. Полученный файл в формате SREC должен быть записан пользователем в энергозависимую флеш-память и впоследствии, при старте по питанию FPGA микросхемы, забирается из флеш-памяти и обрабатывается загрузчиком MBL.

2.2 Область размещения и условия запуска

Описанная в документе [1] микропроцессорная система содержит в себе блок памяти BRAM, подключенный непосредственно к портам «IL» и «DL» микропроцессорного ядра MicroBlaze. В адресном пространстве ядра MicroBlaze этот блок памяти занимает место с нулевого адреса (0x00000000), его объём 32 Кбайта для FPGA Virtex-6 или 128 Кбайт для Virtex-7.

В этом блоке памяти, с адреса 0x00000000, располагается таблица векторов прерываний, в которой сохраняются значения адресов памяти, по которым передается управление микропроцессорным ядром MicroBlaze при возникновении соответствующих прерываний. Объём памяти, отведенный для размещения таблицы векторов прерываний, составляет 80 байт. В соответствии с архитектурой микропроцессорного ядра MicroBlaze, сразу за таблицей векторов прерываний, начиная с адреса 0x00000050 для микропроцессорной системы, описанной в документе [1], может располагаться любой исполняемый двоичный образ.

Двоичный образа загрузчика MBL размещается в данном блоке памяти BRAM. Размещение образа начинается с адреса 0x00000050.

В процессе компиляции проекта загрузчика MBL средой Xilinx SDK формируется двоичный образ загрузчика, в котором содержаться сегменты с исполняемыми инструкциями, данными и необходимыми для определения точек запуска на ядре MicroBlaze значениями таблицы векторов прерываний.

Получаемый в процессе компиляции двоичный образ загрузчика MBL сохраняется вместе с микропроцессорной системой, описанной в документе [1], в «Platform Flash» и при подаче питания на FPGA микросхему автоматически загружается в блок памяти BRAM.

После загрузки двоичного образа загрузчика MBL в блок памяти BRAM, управление микропроцессорным ядром MicroBlaze передается инструкциям, адреса расположения которых определены в таблице векторов прерываний и загрузчик MBL преступает к своей работе.

2.3 Этапы работы

Процесс загрузки и инициализации в блок памяти SDRAM DDR сохраненного во флеш-памяти двоичного образа приложения пользователя в формате SREC и передача инструкциям из состава этого образа управления ядром MicroBlaze осуществляется в следующей последовательности:

- получив управление микропроцессорным ядром MicroBlaze, после инициализации FPGA микросхемы, при подаче на неё питания, загрузчик MBL начинает процесс анализа состояния контроллера SDRAM DDR на предмет его готовности к работе;
- когда контроллер SDRAM DDR готов к работе, загрузчик MBL обращается к контроллеру флеш-памяти и осуществляет перенос исполняемого образа приложения пользователя, сохраненного ранее в формате SREC, во временный буфер памяти, расположенный в блоке памяти SDRAM DDR;
- после переноса образа в формате SREC во временный буфер, загрузчик MBL запускает процесс разбора содержимого этого буфера. Из записей формата SREC извлекаются поля с данными. Эти поля содержат информацию об адресе в адресном пространстве микропроцессора MicroBlaze и данные, которые необходимо поместить по этому адресу. Загрузчик обрабатывает все записи с полями данных из буфера и переносит данные в необходимое адресное пространство;
- по окончании процесса разбора содержимого буфера с SREC записями, загрузчик MBL запрещает обработку всех прерываний микропроцессорному ядру MicroBlaze и меняет содержимое векторов прерываний ядра MicroBlaze на значения, необходимые для запуска двоичного образа приложения пользователя, перенесенного в соответствующее адресное пространство и разрешает все прерывания. Разрешив все прерывания, загрузчик MBL осуществляет переход по адресу, сохраняемому ячейках памяти, закрепленными за вектором прерывания «Reset» (адрес вектора 0x00000000). Переход по адресу, сохраненному в векторе «Reset», означает запуск на исполнение инструкций из состава двоичного образа приложения пользователя.

2.4 Специфика реализации и использования

Основой механизма запуска на исполнение загрузчиком MBL двоичного образа приложения пользователя, сохраненного в формате SREC, является перезапись самим загрузчиком MBL таблицы векторов прерываний микропроцессорного ядра MicroBlaze на значения, которые необходимы для работы двоичного образа приложения пользователя. Необходимые значения векторов прерывания приложения пользователя, как и сам его двоичный образ, хранятся в виде записей формата SREC во флеш-памяти.

Фактически, использование такого механизма запуска двоичного образа приложения пользователя с перезаписью значений таблицы векторов прерываний означает только одно: что повторная работа загрузчика MBL после операции аппаратного сброса ядра MicroBlaze становится невозможна без полной перезагрузки аппаратной платформы в FPGA микросхему.

Реальная эксплуатация аппаратных платформ FPGA микросхем предполагает однократную загрузку аппаратной платформы из «Platform Flash» после подачи питания FPGA микросхеме. Но в работе сложных систем на базе FPGA модулей могут использоваться решения, когда необходимо реагировать на распространение сигнала глобального аппаратного сброса по всей системе в любой момент времени, например в рамках микропроцессорной системы с шиной PCIe. В такой системе сигнал глобального аппаратного сброса генерируется центральным процессором при операциях полной перезагрузки всей системы без выключения ее питания.

После фиксации подобного глобального аппаратного сброса во всей системе, аппаратная платформа FPGA микросхемы транслирует этот сигнал в сигнал аппаратного сброса микропроцессорному ядру MicroBlaze. В результате аппаратного сброса, ядро MicroBlaze получит управление, т. е. перейдет к исполнению инструкции, адрес которой располагается в векторе «Reset» таблицы векторов прерываний. Но в этот момент времени значение этого вектора «Reset» соответствует адресу точки начала работы двоичного образа приложения пользователя. В результате аппаратного сброса приложение пользователя запускается на исполнение повторно.

Казалось бы, что такой повторный перезапуск двоичного образа приложения пользователя — это хорошо, но существует определенная проблема, которая переносит данную возможность перезапуска приложения пользователя из разряда хороших в разряд плохих.

Дело в том, что получаемый в процессе компиляции приложения пользователя двоичный образ содержит в себе различные сегменты: сегменты с исполняемыми инструкциями и сегменты данных, такие как статические и динамические переменные, буфера памяти. Эти переменные и буфера памяти в процессе работы приложения пользователя заполняются различными значениями. В момент аппаратного сброса ядра MicroBlaze, исполняемое им приложение пользователя прерывает свою работу, происходит приведение в первоначальное состояние всех внутренних конечных автоматов микропроцессора MicroBlaze и контроллеров доступа к блокам памяти типа BRAM и SDRAM DDR. После приведения в исходное состояние внутренних конечных автоматов, повторно запускается исполнение ядром MicroBlaze двоичного образа приложения пользователя. Однако, приведение в исходное состояние, необходимое для начала работы приложения пользователя, содержимое блоков памяти BRAM и SDRAM DDR при этом аппаратном сбросе не происходит. Контроллеры этих блоков памяти просто не умеют этого делать, да и это задача, которую они не могут выполнять. В этих блоках памяти продолжают храниться инструкции и данные переменных приложения пользователя, причем переменные уже модифицированны по отношению к состоянию, которое было у них первоначально, по окончании инициализации загрузчиком MBL содержимого SDRAM DDR из данных записей SREC загружаемого двоичного образа приложения пользователя. Фактически, повторный запуск двоичного образа приложения пользователя происходит в условиях, когда первоначальные значения статических и динамических переменных разных буферов памяти не имеют необходимых при запуске приложения пользователя первоначальных значений. Последующие вы-

полнение приложения пользователя с обращением к некорректным данным неизбежно приводит к некорректной работе самого приложения, что не редко заканчивается фатальными ошибками уровня нехватки памяти для выделения новых динамических буферов «в куче» или ошибками на стеке, возникающих при вызове функций приложения.

Единственным выходом, позволяющим избежать некорректной работы приложения пользователя после операции аппаратного сброса ядра MicroBlaze и контроллеров блоков памяти BRAM и SDRAM DDR — это полностью перегрузить двоичный образ приложения пользователя, сохраненный в формате SREC во флеш-памяти. Но как это сделать? Ведь после первой загрузки двоичного образа приложения пользователя загрузчик MBL изменил значения таблицы векторов прерываний со значений, необходимых для работы самого загрузчика MBL, на значения, необходимые двоичному образу приложения пользователя. Как восстановить значения таблицы векторов прерываний на те, которые необходимы для работы загрузчика MBL?

Ответ на этот вопрос разработчики компании ЗАО «Скан Инжиниринг Телеком» нашли в реализации собственного решения на аппаратном уровне контроллера блока памяти BRAM, который умеет запоминать инициализированное содержимое таблицы векторов прерываний ядра MicroBlaze в блоке памяти BRAM в момент времени, соответствующий первому запуску аппаратной платформы при подаче питания на FPGA микросхему. Данное решение позволяет после фиксации сигнала аппаратного сброса выдать контроллером блока памяти BRAM значения таблицы векторов прерываний, необходимые для корректного запуска самого загрузчика MBL, вместо текущей таблицы векторов прерываний, путем подстановки ранее запомненных значений при обращении к блоку BRAM по адресам памяти, соответствующим месту расположения таблицы векторов прерывания.

В результате такой манипуляции со значениями таблицы векторов прерываний на уровне контроллера блока памяти BRAM появляется возможность осуществлять корректный перезапуск загрузчика MBL после возникновения события аппаратного сброса ядра MicroBlaze. Повторная работа загрузчика MBL приводит к перезагрузке исполняемого образа приложения пользователя из флеш-памяти в блок памяти SDRAM DDR. Такая полная перезагрузка двоичного образа приложения пользователя означает корректное содержание областей памяти со статическими и динамическими переменными, различными буферами памяти, что позволяет приложению пользователя корректно функционировать после аппаратного сброса ядра MicroBlaze.

Обратите внимание, что рассмотренная в данном разделе специфика работы загрузчика MBL касается его корректного функционирования на аппаратной платформе, которая содержит в себе специальный контроллер блока памяти BRAM, способный производить манипуляции со значениями таблицы векторов прерываний микропроцессорного ядра MicroBlaze.

2.5 Структура каталогов проекта

Проект размещается в каталоге «MBL», в котором располагаются каталоги верхнего уровня «projects», «src» и «doc».

Каталог «doc» содержит документацию, поставляемую с проектом загрузчика MBL.

Каталог «projects» содержит подкаталоги с файлами проектов для среды разработки Xilinx SDK. Имена подкаталогов формируются по шаблону «HW_[имя модуля FPGA]». Например для VPX модуля SVP-726 имя подкаталога будет «HW_SVP_726». При работе со средой разработки Xilinx SDK пользователю необходимо выбрать для импорта в свое рабочее пространство соответствующий своему типу модуля FPGA вариант проекта загрузчика MBL.

Каталог «src» содержит в себе файлы с исходными текстами проекта на языке «Си».

Каталог «utils» содержит в себе файлы дополнительных сервисных утилит с исходными текстами проекта на языке «Си». Сервисные утилиты необходимы для обеспечения дополнительного функционала, напрямую не связанного с непосредственной работой загрузчика MBL. Примером такой утилиты служит проект «Spi_Flash_Writer», предназначенный для обеспечения записи двоичного образа приложения пользователя, сохраненного в формате SREC, в SPI флеш-память, или проект «User_App_Demo», предназначенный для демонстрации в качестве приложения пользователя.

В общем виде структура каталогов проекта загрузчика MBL приведена в листинге 2-1.

Листинг 2-1: Общий вид структуры каталогов проекта «MBL»

```
\---MBL
    +---doc
    +---projects
    |   +---HW_SAMC_713
    |   +---HW_SVP_713
    |   \---HW_SVP_726
    +---src
    \---utils
        +---spi_flash_writer
        |   +---projects
        |   |   \---HW_SVP_726
        |   \---src
        \---user_app_demo
            \---src
```

3 Компиляция проекта

Компиляция проекта осуществляется в среде разработки Xilinx SDK.

Изначально в концепцию реализации проекта загрузчика MBL заложена возможность его использования для работы с разными конфигурациями аппаратных платформ, предназначенными для различных FPGA модулей производства ЗАО «Скан Инжиниринг Телеком». Это позволяет использовать проект загрузчика MBL на разных семействах FPGA микросхем Xilinx.

В зависимости от семейства FPGA микросхем, компания Xilinx предлагает разные версии среды разработки Xilinx SDK. Данное руководство охватывает работу со следующими средами разработки:

- Xilinx SDK Release Version: 14.6;
- Xilinx SDK Release Version: 2014.2.

3.1 Управление компиляцией

При осуществлении процесса компиляции проекта загрузчика MBL пользователю предоставляются следующие возможности по управлению компиляцией:

- выбор режима компиляции;
- настройка аппаратной платформы, для которой будет осуществлена компиляция.

3.1.1 Режимы компиляции

Компиляция проекта загрузчика MBL может осуществляется в двух режимах:

- Debug;
- Release.

Режим «Debug» используется на стадии разработки, когда требуется отладка исходного кода проекта загрузчика MBL. Процесс отладки производится с использованием внешнего JTAG устройства. Получаемый в процессе компиляции в этом режиме двоичный образ загрузчика MBL характеризуется следующими характеристиками:

- относительно медленным исполнением микропроцессорным ядром MicroBlaze, так как при компиляции проекта используются флаги компилятора уровня оптимизации −O0 и флаги включения в образ символов отладки −g3;
- большим объёмом в следствии включения в состав образа дополнительного объёма выводимых в консоль текстовых сообщений, используемых с целью более подробного информирования о внутреннем состоянии протекающих процессов в исполняемом образе.

Режим «Release» используется для формирования окончательного рабочего облика двоичного образа загрузчика MBL, сохраняемого вместе с аппаратной платформой для FPGA микросхемы в «Platform Flash». Получаемый в процессе компиляции в этом режиме двоичный образ загрузчика MBL характеризуется следующими характеристиками:

- максимально возможной скоростью исполнения микропроцессорным ядром MicroBlaze, так как при компиляции проекта используются флаги компилятора уровня оптимизации −O3 и флаги отсутствия включения в образ символов отладки;
- минимальным объёмом всвязи с отсутствием в составе образа дополнительного объёма выводимых в консоль текстовых сообщений, используемых с целью более подробного информирования о внутреннем состоянии протекающих процессов в исполняемом образе.

3.1.2 Настройка аппаратной платформы

Для получения исполняемого двоичного образа загрузчика MBL, пригодного для работы с конкретной аппаратной платформой, применяются макросы оператора «#define» препроцессора языка «Си» и настройки символов компиляции проекта в его свойствах в среде разработки Xilinx SDK.

Определение макросов оператора «#define» препроцессора языка «Си» находятся в файле «blconfig.h». Макросы определяют набор поддерживаемых FPGA модулей. Значения этих макросов используются для определения выбора конкретного модуля, для которого будет компилироваться двоичный образ загрузчика MBL. В листинге 3-1 показаны значения этих макросов.

Листинг 3-1: Макросы определения поддерживаемых модулей FPGA

```
...

#define G_MBL_HW_PLATFORM_S6T_SP605_REV_D       1
#define G_MBL_HW_PLATFORM_SVP_713_REV_1         2
#define G_MBL_HW_PLATFORM_SAMC_713_REV_1        3
#define G_MBL_HW_PLATFORM_V7T_VC_709_REV_1      4
#define G_MBL_HW_PLATFORM_SVP_726_REV_1         5

...
```

Выбор конкретного модуля FPGA, для которого компилируется исполняемый двоичный образ загрузчика MBL, осуществляется через опцию компилятора «-D», путем определения в этой опции для символа «G_MBL_HW_PLATFORM» числового значения. Определение осуществляется в настройках свойств проекта в среде разработки Xilinx SDK.

Окно среды разработки Xilinx SDK, в котором осуществляется определение значения символа «G_MBL_HW_PLATFORM» показано на рисунке 3-1. Числовое значение для символа «G_APP_HW_PLATFORM» выбирается из предопределенных значений макросов поддерживаемых модулей FPGA, определенных в файле «blconfig.h».

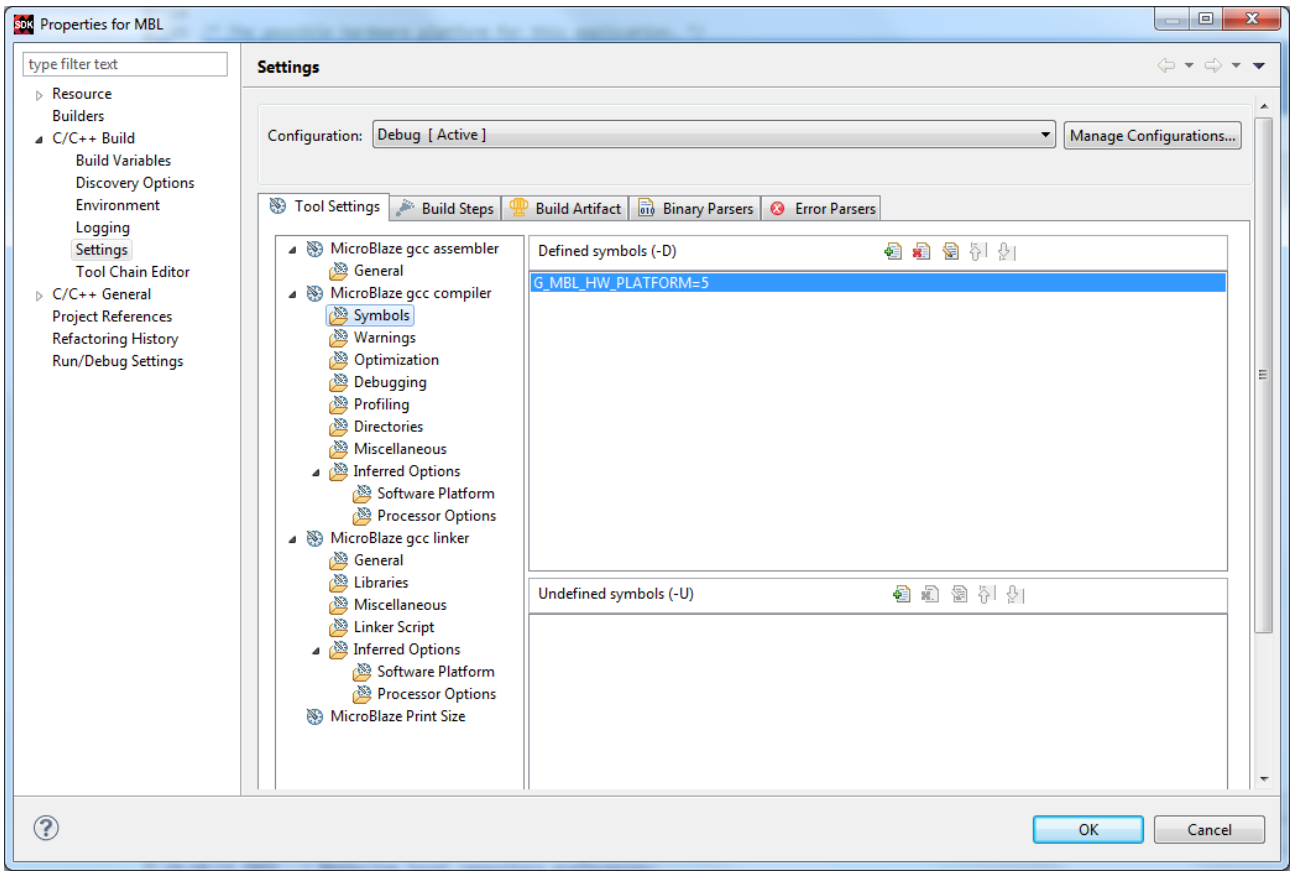

Рисунок 3-1: Уровни организации исходного кода проекта

3.1.3 Настройка области размещения

Получаемый в процессе компиляции проекта двоичный образ загрузчика MBL в формате ELF делится на сегменты, которые необходимо разместить в блоке памяти BRAM, разрешенном для хранения исполняемых инструкций микропроцессорному ядру MicroBlaze. В общем случае, сегменты бинарного образа выглядят так, как показанно на рисунке 3-2.

Управление процессом размещения сегментов двоичного образа загрузчика MBL осуществляется путем изменения настроек в файле «lscript.ld», включаемом в состав проекта.

Рисунок 3-2: Сегменты двоичного образа

3.2 Подготовка к компиляции

3.2.1 Выбор рабочего пространства

Процедура 3-1. Выбор места расположения рабочего пространства в среде Xilinx SDK

1. Запустите среду разработки Xilinx SDK и выбирите место расположения «Workspace». По умолчанию он располагается в папке со следующим путем: «C:\set_spb\dev\[имя модуля]\[имя сборки аппаратной платформы]\SDK\SDK_Export».

   Например, для модуля «SAMC-713», на котором запускается аппаратная платформа с именем сборки «Core 0», путь к «Workspace» выглядит как показанно на рисунке 3-3.

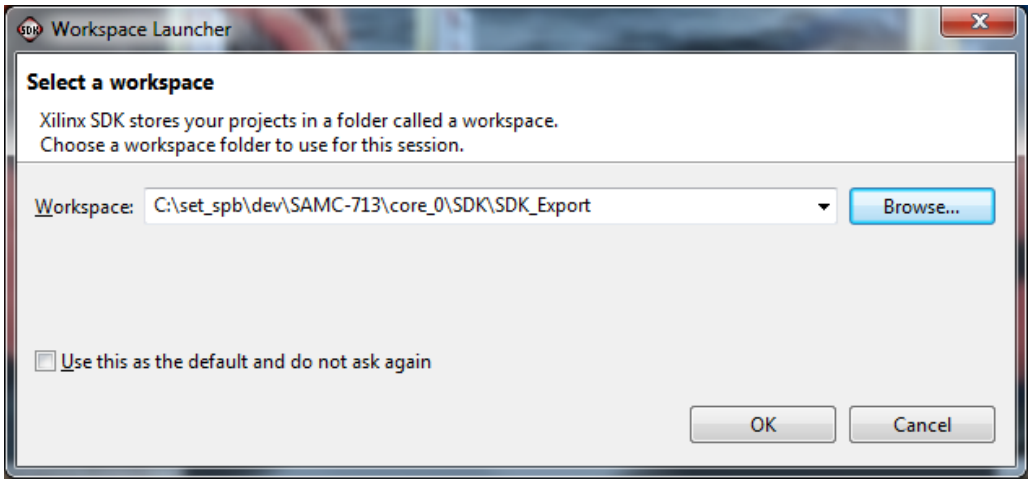

Рисунок 3-3: Выбор места расположения рабочего пространства «Workspace» в Xilinx SDK

3.2.2 Настройка репозитариев

Процедура 3-2. Выбор места расположения репозитариев

1. Выберите пункт меню «Xilinx Tools > Repositories», как показано на рисунке 3-4.

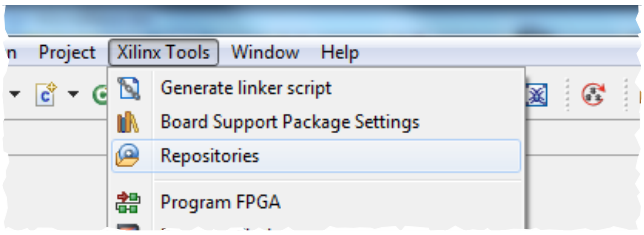

Рисунок 3-4: Выбор пункта меню настройки репозитариев в Xilinx SDK

2. В открывшемся окне, нажав кнопку «New», выберите место расположение репозитария: «C:\set_spb\dev\[имя модуля]]\[имя сборки аппаратной платформы]».

   Например, для модуля «SAMC-713», на котором запускается аппаратная платформа с именем сборки «Core 0», пути к репозитариям выглядят как показано на рисунке 3-5.

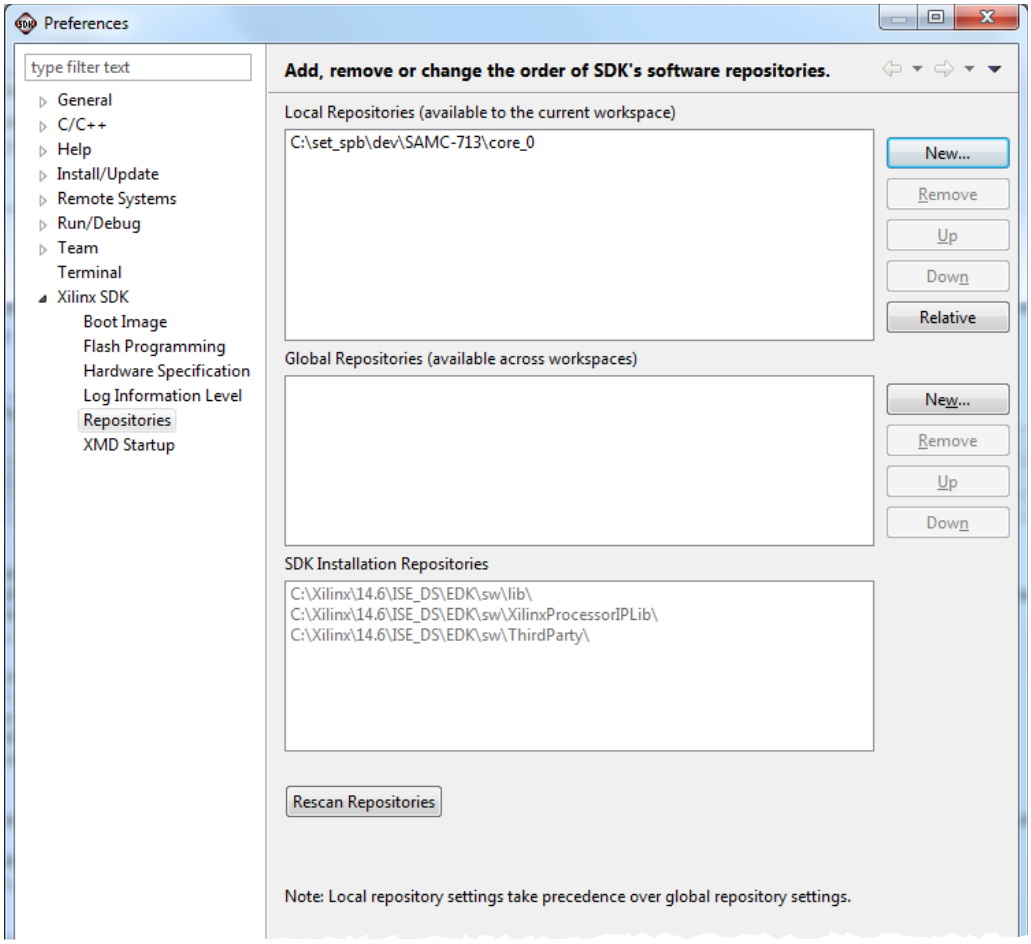

Рисунок 3-5: Настройка местоположения репозитариев в Xilinx SDK

3. Нажмите кнопку «Rescan Repositories» чтобы обновить информацию о репозитариях и выйдите из открытого окна, нажав кнопку «OK».

3.2.3 Создание проекта аппаратной платформы

Процедура 3-3. Создание нового проекта для аппаратной платформы в Xilinx SDK

1. Выбирете пункт меню «File > New > Project».

2. В открывшемся окне выбирете в списке проектов тип «Hardware Platform Specification», как показано на рисунке 3-6 и нажмите кнопку «Next».

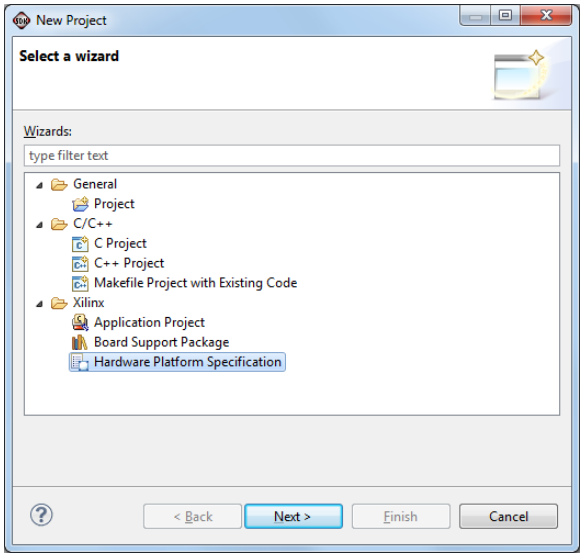

Рисунок 3-6: Создание проекта аппаратной платформы в Xilinx SDK

3. В открывшемся окне, как показано на рисунке 3-7, выбирите место расположения файла спецификации аппаратной платформы, нажав кнопку «Browse». Файл располагается в следующей папке: «C:\set_spb\dev\[имя модуля]\[имя сборки аппаратной платформы]\SDK\SDK_Export\hw».

   Для среды разработки Xilinx SDK Release Version: 14.6 имя файла спецификации аппаратной платформы носит название «system.xml».

   Для среды разработки Xilinx SDK Release Version: 2014.2 имя файла спецификации аппаратной платформы носит название «system.hdf».

   Например, для модуля «SAMC-713», на котором запускается аппаратная платформа с именем сборки «Core 0», путь к файлу спецификации аппаратной платформы выглядит следующим образом: «C:\set_spb\dev\SAMC-713\core_0\SDK\SDK_Export\hw\system.xml».

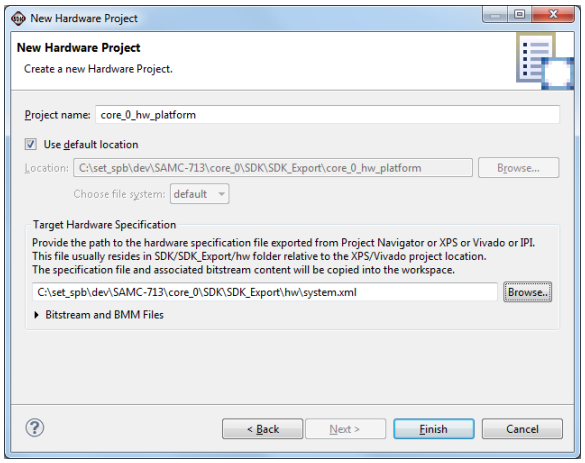

Рисунок 3-7: Создание проекта аппаратной платформы в Xilinx SDK

4. Закройте окно, нажав кнопку «Finish».

3.2.4 Создание проекта BSP поддержки аппаратной платформы

Процедура 3-4. Создание нового проекта BSP поддержки аппаратной платформы в Xilinx SDK

1. Выбирите пункт меню «File > New > Project».
2. В открывшемся окне выбирите в списке проектов тип «Board Support Package», как показано на рисунке 3-8 и нажмите кнопку «Next».

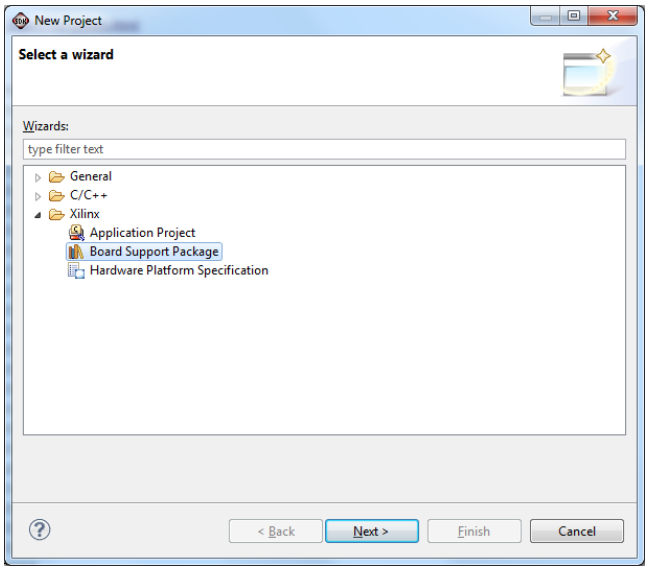

Рисунок 3-8: Создание проекта BSP поддержки аппаратной платформы в Xilinx SDK

3. В открывшемся окне выбирете тип проекта «standelone», как показано на рисунке 3-9.

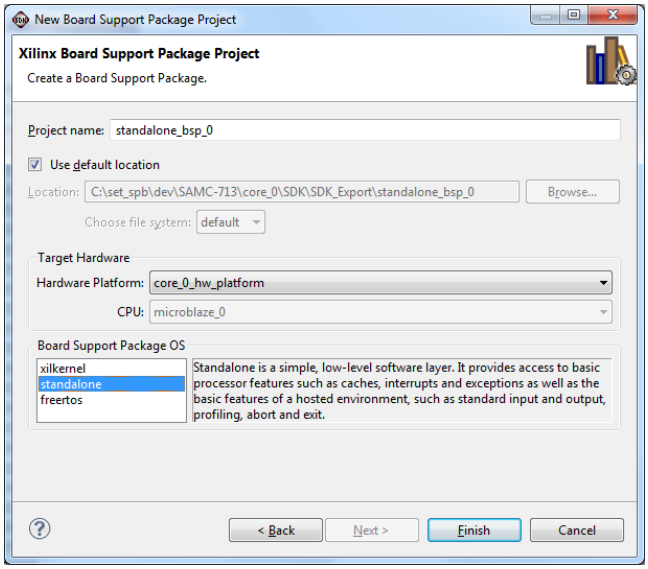

Рисунок 3-9: Создание проекта BSP поддержки аппаратной платформы в Xilinx SDK

4. Закройте окно, нажав кнопку «Finish».
5. Закройте открывшееся окно настроек параметров BSP пакета поддержки аппаратной платформы (рисунок 3-10), нажав кнопку «OK».

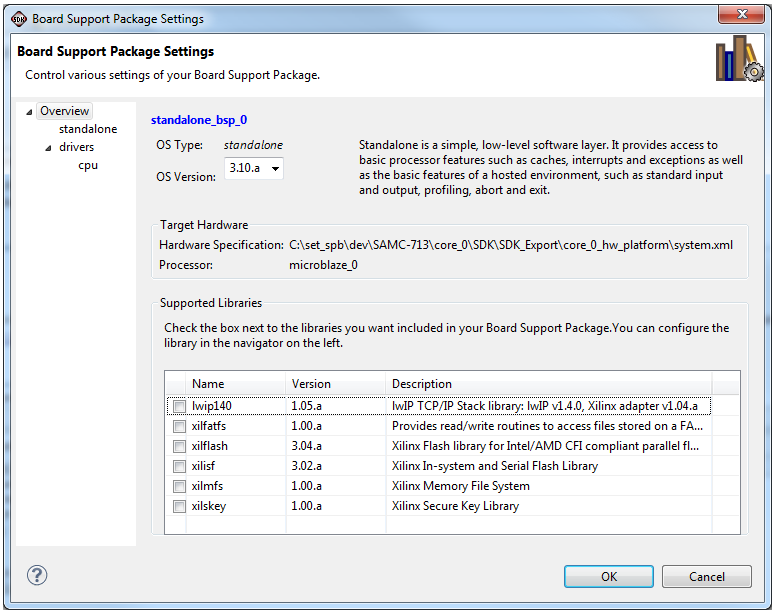

Рисунок 3-10: Создание проекта BSP поддержки аппаратной платформы в Xilinx SDK

3.2.5 Импорт проекта

Процедура 3-5. Импорт проекта загрузчика MBL в рабочие пространство

1. Выбирите пункт меню «File > Import».
2. В открывшемся окне (см. рисунок 3-11) выберите в списке проектов тип «Exising Projects into Workspace» и нажмите кнопку «Next».

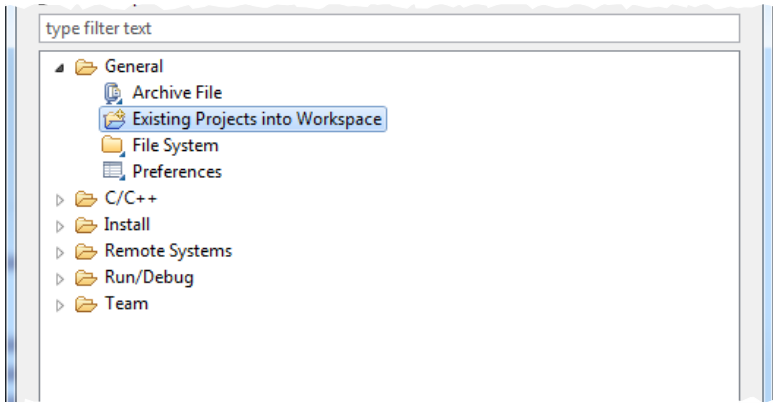

Рисунок 3-11: Импорт проекта в Xilinx SDK

3. В открывшемся окне (рисунок 3-12) в поле «Select root directory», нажав кнопку «Browse…», выбирите место расположения импортируемого проекта «C:\set_spb\dev\[имя_модуля]\MBL\projects».

   Например, для модуля «SAMC-713», на котором запускается аппаратная платформа с именем сборки «Core 0», путь к импортируемому проекту выглядит следующим образом: «C:\set_spb\dev\SAMC-713\MBL\projects».
4. Выбирите в текущем окне (рисунок 3-12), в соответствии с типом используемого модуля FPGA, импортируемый проект, оставив напротив строки с его названием галочку выбора, и нажмите кнопку «Finish».

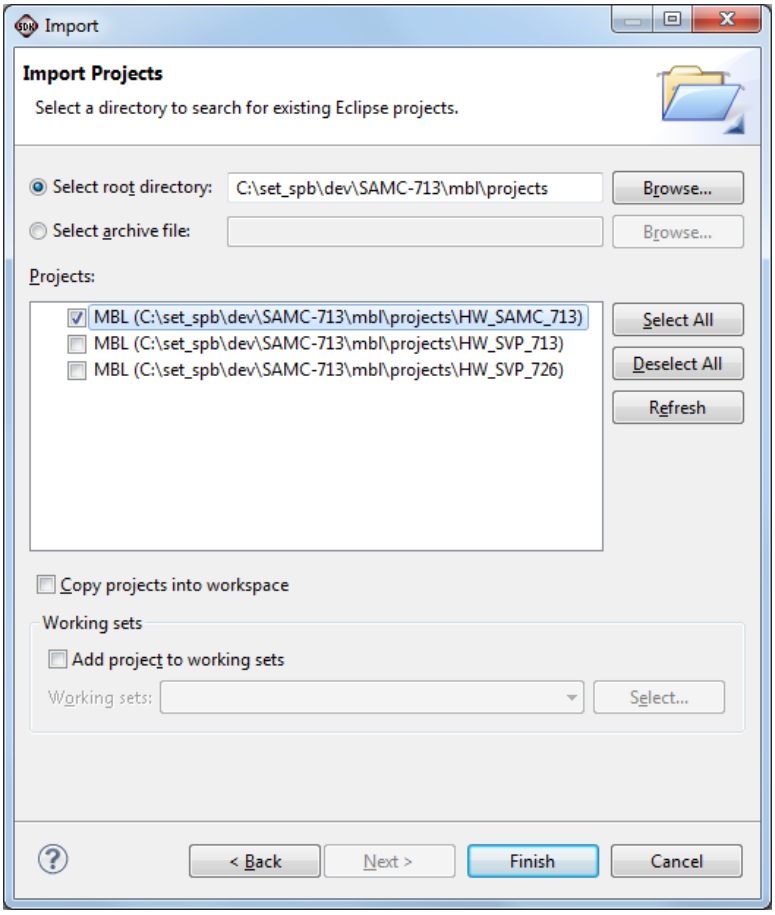

Рисунок 3-12: Выбор проекта для импорта в Xilinx SDK

3.3 Компиляция

Процедура 3-6. Компиляция рабочей версии

1. Выбирите в окне «Project Explorer» проект «MBL» (см. рисунок 3-13), и установите необходимый режим компиляции: «Debug» или «Release».

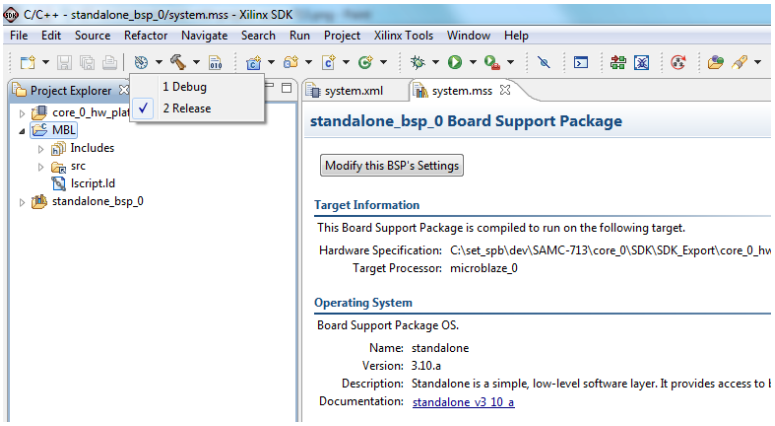

Рисунок 3-13: Сборка проекта в Xilinx SDK

2. Проверьте содержимое файла «lscript.ld» на предмет правильного размещения сегментов двоичного образа загрузчика MBL (рисунок 3-14).

Например, для модуля «SAMC-713», на котором запускается аппаратная платформа с именем сборки «Core 0», для загрузчика MBL в файле «lscript.ld» должно быть:

- в поле «Stack Size» значение «0x800»;
- в поле «Heap Size» значение «0x0»;
- в полях столбца «Memory Region» таблицы «Section to Memory Region Mapping» должно быть значение «microblaze_0_i_bram_ctrl_microblaze_0_d_bram_ctrl».

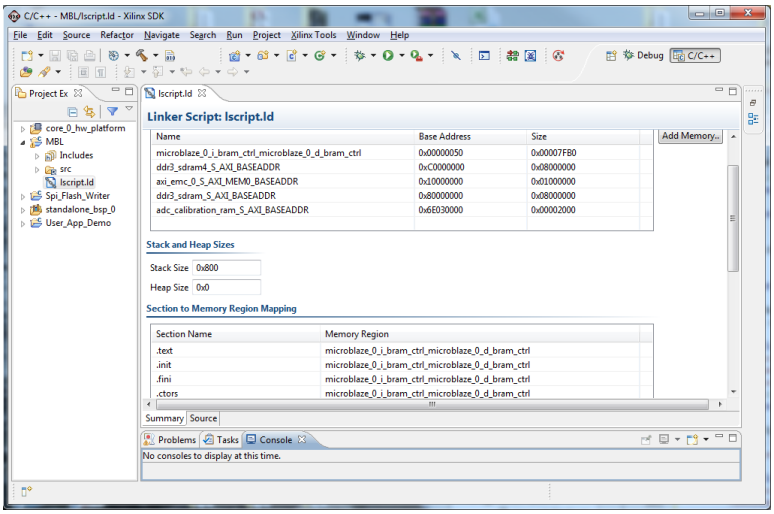

Рисунок 3-14: Содержимое файла «lscript.ld» в Xilinx SDK

3. Запустите процесс компиляции. В случае успешной компиляции проекта, вывод сообщений компилятора и утилиты компоновщика в окно «Console» должен выглядеть подобно тому, что приведен на рисунке 3-15.

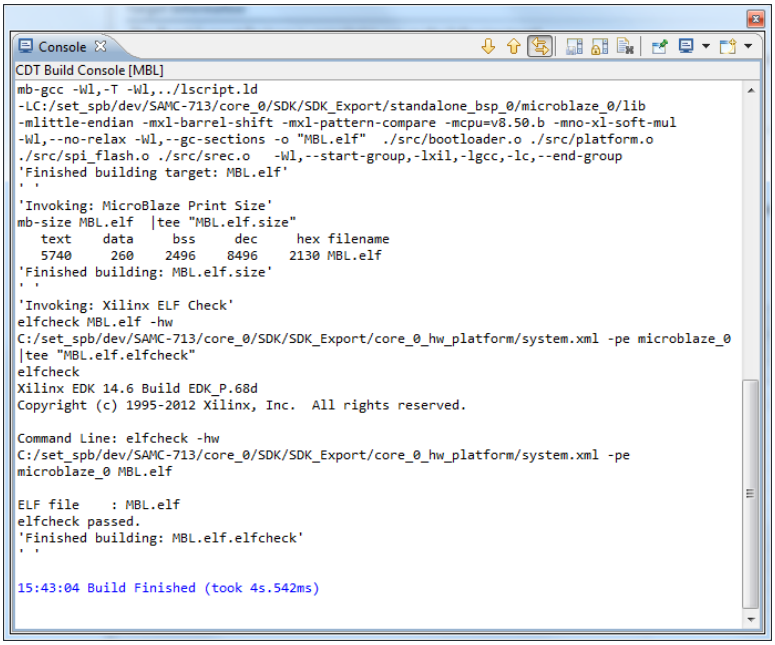

Рисунок 3-15: Протокол сборки проекта в Xilinx SDK

4 Запуск проекта

Прежде чем осуществить процесс запуска на исполнение микропроцессорным ядром MicroBlaze полученного в результате компиляции двоичного образа загрузчика MBL, необходимо подготовить и сохранить во флеш-памяти FPGA модуля двоичный образ приложения пользователя в формате SREC. Выполнение этих операций необходимо для того, чтобы загрузчику MBL было что извлекать из флеш-памяти FPGA модуля для осуществления процесса инициализации SDRAM BRAM содержимым двоичного образа приложения пользователя, с последующим запуском этого приложения.

Процедуры подготовки и сохранения во флеш-памяти FPGA модуля двоичного образа приложения пользователя в формате SREC подробно описанны в приложении Б данного руководства.

Дальнейшее описание процедуры запуска проекта производиться с учетом того, что пользователь уже сохранил необходимый ему образ своего приложения в формате SREC во флеш-памяти FPGA модуля.

Процесс запуска на исполнение микропроцессорным ядром MicroBlaze полученного в результате компиляции проекта двоичного образа загрузчика MBL может быть осуществлен двумя путями:

- непосредственной загрузкой в FPGA микросхему аппаратной платформы и исполняемого двоичного образа загрузчика из среды разработки Xilinx SDK;
- предварительной записи аппаратной платформы и исполняемого двоичного образа в загрузочную флеш-память «Platform Flash» FPGA модуля.

Первый путь, непосредственной загрузки в FPGA микросхему, предполагает, что к FPGA модулю подключен JTAG отладчик фирмы Xilinx, работающий со средой разработки Xilinx SDK. Загрузка осуществляется непосредственно из среды разработки. Загружаемые в FPGA аппаратная платформа и исполняемый двоичный образ способны работать все время пока подается питание на FPGA модуль. После отключения питания и повторной его подачи, загрузку аппаратной платформы и исполняемого двоичного образа нужно будет выполнить повторно. Данный вариант загрузки и работы, как правило, пригоден для этапов разработки и функционального тестирования.

Второй путь, предварительной записи в загрузочную «Platform Flash», используется для подготовки FPGA модуля к постоянной работе. Работа FPGA модуля предполагает периодическую подачу и отключение его питания. При использовании данного пути, непосредственно сам запуск загрузчика MBL осуществляется автоматически, после подачи питания FPGA микросхеме, когда в нее загружается конфигурация из «Platform Flash». Прибегать к использованию данного пути загрузки и запуска на этапе разработки нужно как можно реже, так как ресурс работоспособности флеш-памяти ограничен циклами перезаписи. Для осуществления записи аппаратной платформы и исполняемого двоичного образа в загрузочную «Platform Flash» необходимо воспользоваться JTAG отладчиком, подключаемым к модулю FPGA, средой разработки Xilinx SDK и утилитой Xilinx iMPACT.

Перед началом запуска полученного в процессе компиляции двоичного образа загрузчика MBL подключите JTAG отладчик фирмы Xilinx к FPGA модулю и персональному компьютеру, на котором установлена среда Xilinx SDK.

Так же необходимо подключить к последовательному порту ввода/вывода (консольному порту) FPGA модуля и персональному компьютеру USB-кабель, и запустить на персональном компьютере терминальную программу, чтобы видеть в окне терминальной программы текстовые сообщения загрузчика MBL в процессе его работы. Настройки последовательного порта терминальной программы: 115200-8-N1.

4.1 Через JTAG интерфейс

Процедура 4-1. Запуск загрузчика MBL через JTAG интерфейс

1. После окончания процесса компиляции двоичного образа загрузчика MBL в среде Xilinx SDK выберите пункт меню «Xilinx Tools > Program FPGA», как показано на рисунке 4-1.

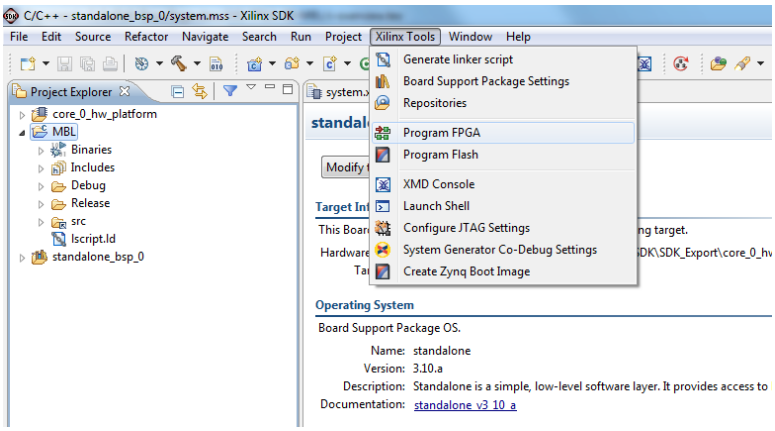

Рисунок 4-1: Выбор режима программирования FPGA через JTAG интерфейс в Xilinx SDK

2. В открывшемся окне (рисунок 4-2) выбирите в таблице «Software Configuration» загружаемый в FPGA микросхему двоичный образ загрузчика MBL.

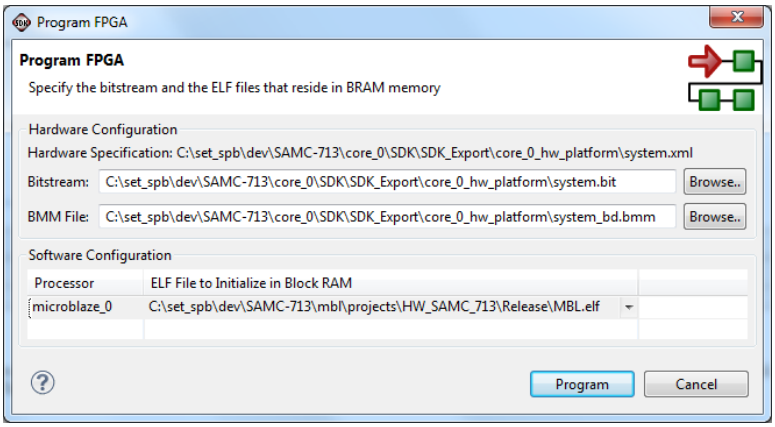

Рисунок 4-2: Выбор загружаемого в FPGA микросхему двоичного образа загрузчика MBL в Xilinx SDK

3. Запустите процесс загрузки в FPGA-микросхему аппаратной платформы и двоичного образа загрузчика MBL, нажав кнопку «Program».

   По окончании процесса загрузки двоичный образ загрузчика MBL автоматически приступит к своей работе, о чем будет свидетельствовать наличие выводимых им текстовых сообщений в запущенную на персональном компьютере терминальную программу. Содержимое окна терминальной программы приведено на рисунке 4-3.

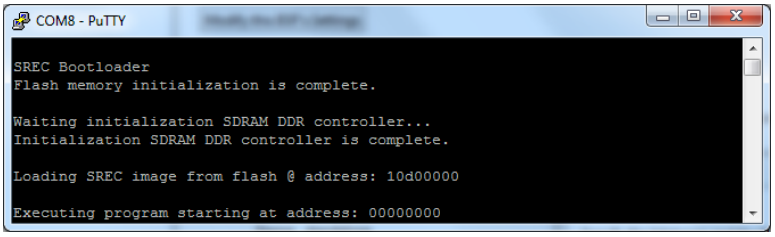

Рисунок 4-3: Терминальный вывод сообщение о работе загрузчика MBL

4.2 Из «Platform Flash»

Процедура 4-2. Запуск загрузчика MBL из «Platform Flash»

1. После окончания процесса компиляции двоичного образа загрузчика MBL в среде Xilinx SDK выполните подготовку объединенного двоичного образа системной платформы и загрузчика MBL, сохраняемого в файле «download.bit». Данный файл формируется автоматически средой Xilinx SDK при выполнении последовательности действий, описанных в процедуре 4-1. Выполните эту последовательность.
2. Осуществите процесс записи объединенного образа из файла «download.bit» в «Platform Flash», выполнив последовательность действий, описанных в приложении A.
3. После успешного окончания процедуры записи объединенного образа из файла «download.bit» в «Platform Flash» результат работы загрузчика MBL можно увидеть в виде текстовых сообщений, отображаемых в окне терминальной программы, приведенных на рисунке 4-3.

Приложение А Запись аппаратной платформы и загрузчика MBL в «Platform Flash»

На всех FPGA модулях производства ЗАО «Скан Инжиниринг Телеком» для осуществления процесса конфигурирования основной доступной для работы пользователю FPGA микросхемы, устанавливается «Platform Flash». В зависимости от версии FPGA модуля, данная флеш-память может быть подключена непосредственно к основной FPGA микросхеме через параллельный интерфейс или через специальную сервисную FPGA микросхему, которая эмулирует работу «Platform Flash» для основной FPGA микросхемы. Примером FPGA модуля с такой эмуляцией работы «Platform Flash» выступает модуль SVP-726.

Использование на ряде FPGA модулей сервисной FPGA микросхемы обеспечивает пользователю гибкость в эксплуатации таких FPGA модулей в части возможности конфигурирования их «Platform Flash» через сеть Ethernet, используя специальную сервисную утилиту производства ЗАО «Скан Инжиниринг Телеком». Более подробно с работой этой утилиты можно ознакомиться в руководстве по эксплуатации конкретного FPGA модуля, на котором устанавливается такая сервисная FPGA.

Для предварительной записи двоичного образа аппаратной платформы и загрузчика MBL в загрузочную «Platform Flash», подключенную непосредственно к основной FPGA микросхеме, необходимо воспользоваться: JTAG отладчиком, средой разработки Xilinx SDK и утилитой Xilinx iMPACT.

Процедура А-1. Запись аппаратной платформы и загрузчика MBL в «Platform Flash»

1. Перед записью в «Platform Flash» можно проверить работоспособность аппаратной платформы и двоичного образа загрузчика MBL, выполнив процедуру 4-1, или пропустить этот шаг, если такая проверка была выполнена ранее.
2. Запустите утилиту Xilinx iMPACT.
3. В открывшемся окне «New iMPACT Project» (рисунок А-1) создайте новый проект, установив селектор выбора вида проекта в положение «create a new project(.ipf)» и нажмите кнопку «Browse…».

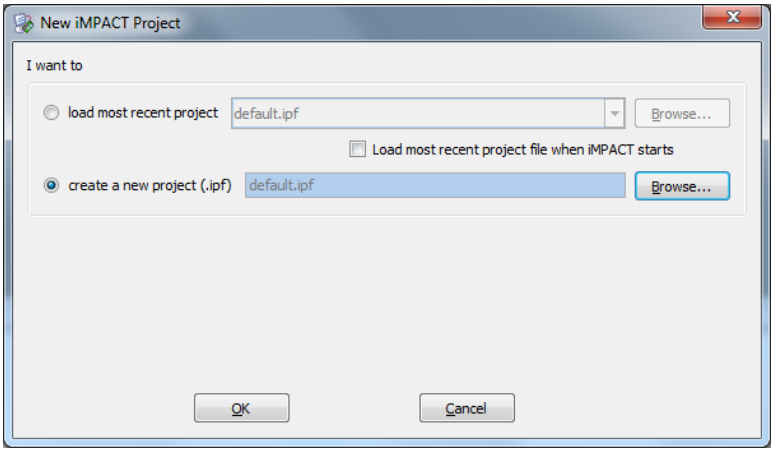

Рисунок А-1: Создание нового проекта в Xilinx iMPACT

4. В открывшемся окне «Create New iMPACT Project File» (рисунок А-2) выберите место размещения для каталога нового проекта и создайте в файловой системе новый каталог с именем «pf_image_for_core_0».

   Например, для модуля «SAMC-713», на котором запускается аппаратная платформа с именем сборки «Core 0», путь к новому каталогу «pf_image_for_core_0» может быть следующим: «C:\set_spb\dev\SAMC-713\pf_image_for_core_0».

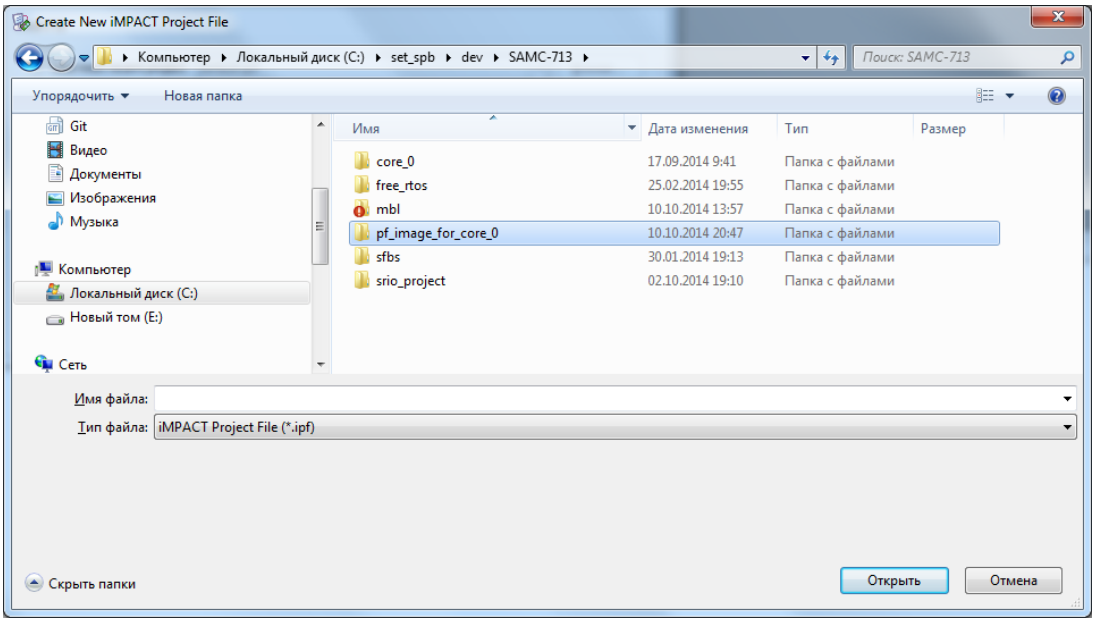

Рисунок A-2: Создание каталога для нового проекта в Xilinx iMPACT

5. Выберите созданный каталог «pf_image_for_core_0» в качестве места сохранения нового проекта iMPACT и введите в поле «Имя файла» новое имя проекта и нажмите кнопку «Сохранить».

   Например, для модуля «SAMC-713», на котором запускается аппаратная платформа с именем сборки «Core 0», имя проекта может быть «samc_713_core_0_pf_image» (рисунок A-3).

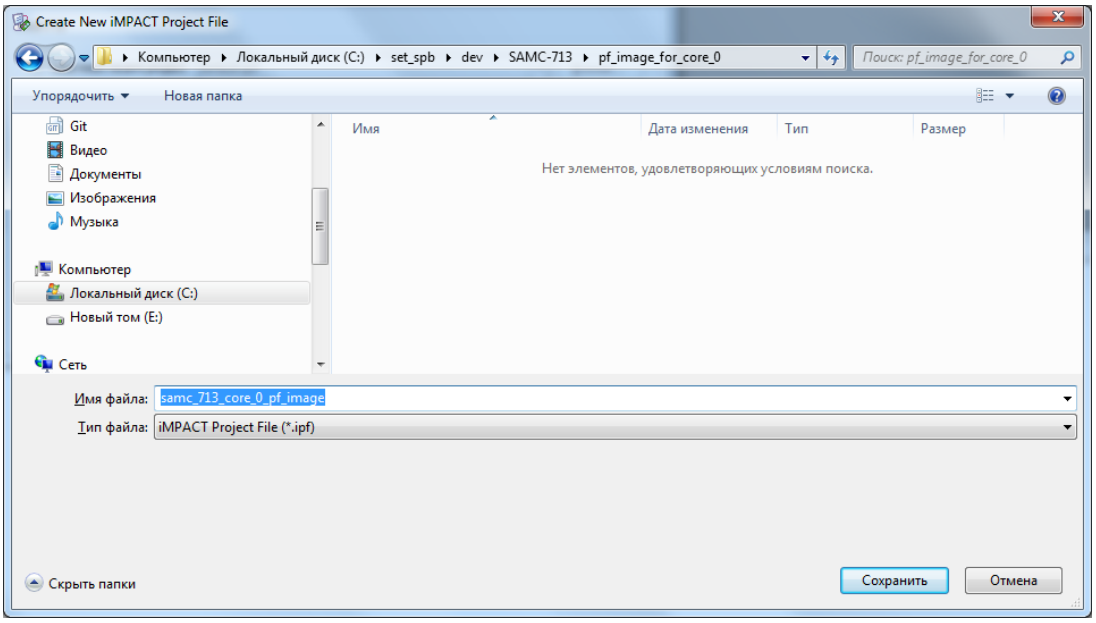

Рисунок A-3: Ввод имени для нового проекта в Xilinx iMPACT

6. В окне создание нового проекта «Create New iMPACT Project File» (рисунок A-1) завершите процесс создания, нажав кнопку «OK».
7. В открывшемся окне «Welcome to iMPACT» (рисунок A-4) выберите тип нового проекта «Prepare a PROM File» и нажмите кнопку «OK».

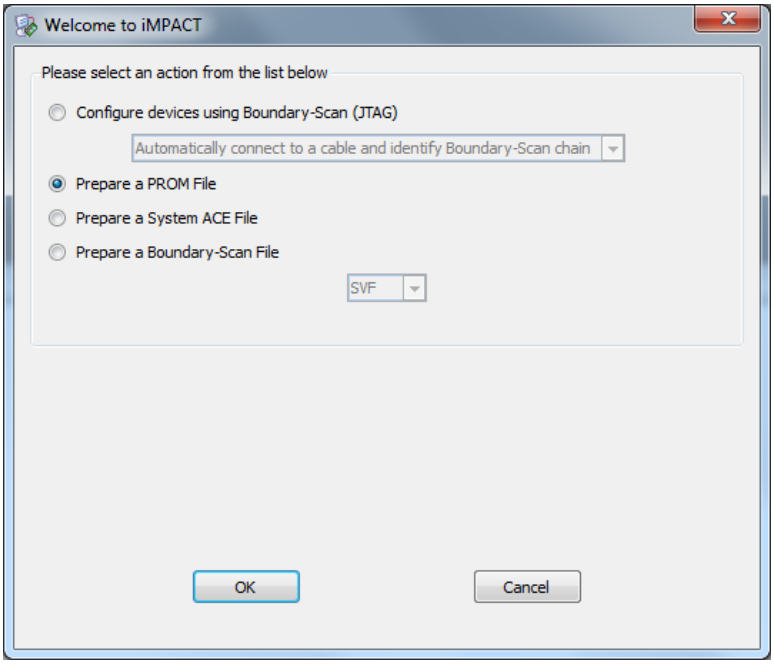

Рисунок A-4: Выбор типа нового проекта в Xilinx iMPACT

8. В открывшемся окне «PROM File Formatter» (рисунок A-5) последовательно выполните настройку следующих параметров: «Storage Device Type», «Target FPGA», «Storage Device (Bytes)», «Output File Name» и нажмите кнопку «OK».

   Например, для модуля «SAMC-713», на котором запускается аппаратная платформа с именем сборки «Core 0», параметры будут следующими:

   - «Storage Device Type» -> «BPI Flash» -> «Configure Single FPGA»;
   - «Target FPGA» -> «Virte6»;
   - «Storage Device (Bytes)» -> «xcf128x»;
   - «Output File Name» -> «samc_713_core_0_img».

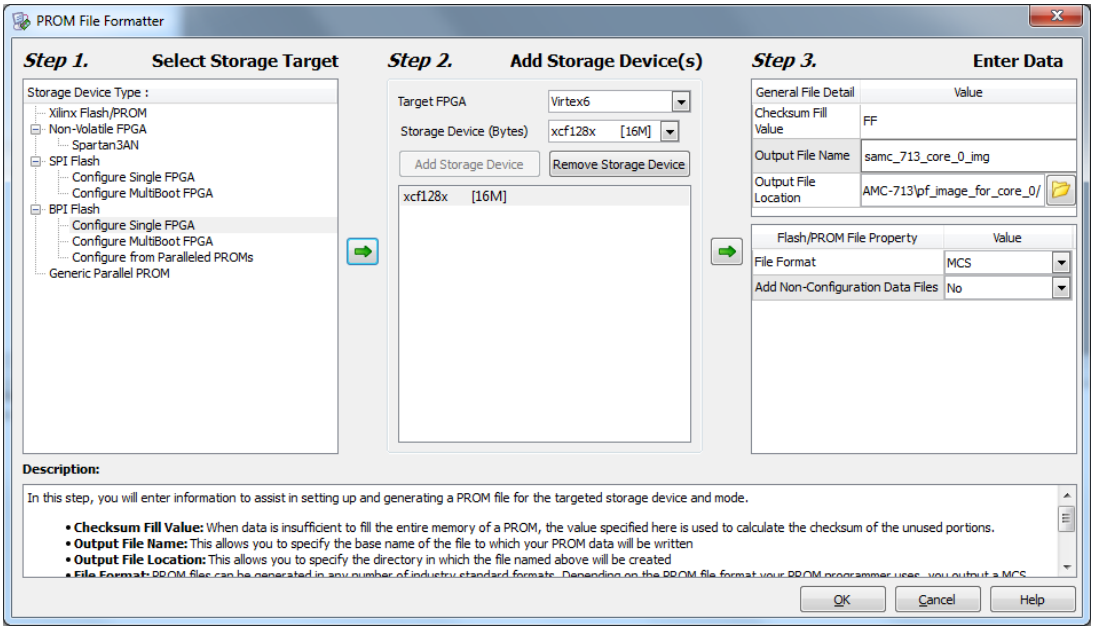

Рисунок A-5: Настройка параметров нового проекта в Xilinx iMPACT

9. В открывшемся окне «Add Device» (рисунок A-6) нажмите кнопку «OK».

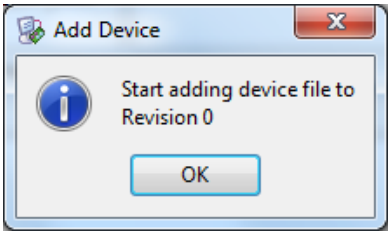

Рисунок A-6: Добавление устройств в новый проект в Xilinx iMPACT

10. В открывшемся окне «Add Device» (рисунок A-7) выберите файл «download.bit», содержащий в себе двоичный образ аппаратной платформы и объединенный с ней двоичный образ загрузчика MBL и нажмите кнопку «Открыть».

    Например, для модуля «SAMC-713», на котором запускается аппаратная платформа с именем сборки «Core 0», место расположения файла «download.bit» будет следующим: «C:\set_spb\dev\SAMC-713\core_0\SDK\SDK_Export\core_0_hw_platform\download.bit».

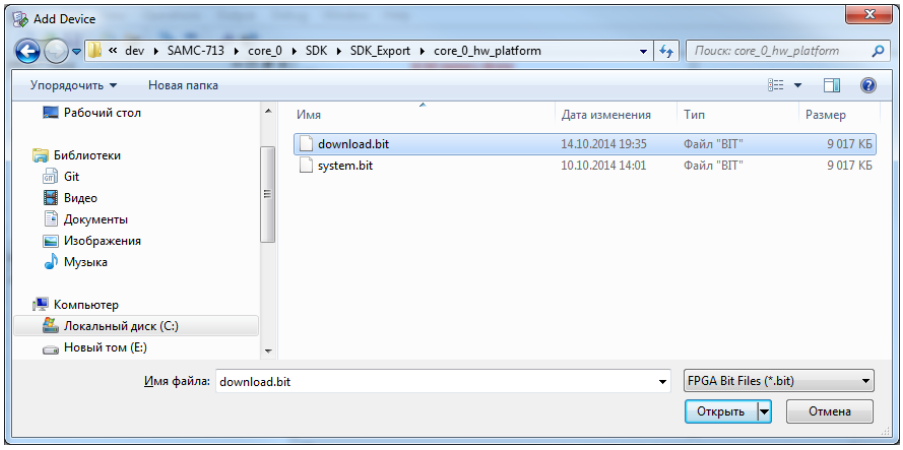

Рисунок A-7: Добавление двоичного образа устройств «download.bit» в новый проект в Xilinx iMPACT

11. В открывшемся окне «Add Device» (рисунок A-8) на предложение добавить новое устройство ответить нет, нажав кнопку «No».

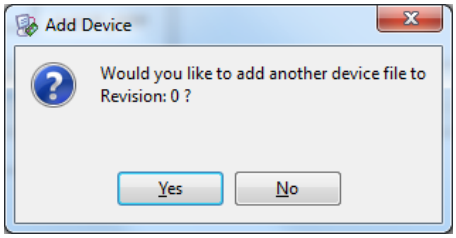

Рисунок A-8: Добавление еще одного устройств в новый проект в Xilinx iMPACT

12. В открывшемся окне «Add Device» (рисунок A-9) нажмите кнопку «OK».

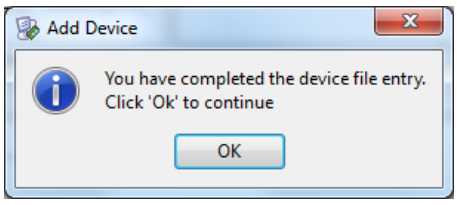

Рисунок A-9: Завершение добавления устройств в новый проект в Xilinx iMPACT

13. В открывшемся окне «MultiBoot BPI Revision and Data File Assignment» (рисунок A-10) нажмите кнопку «OK».

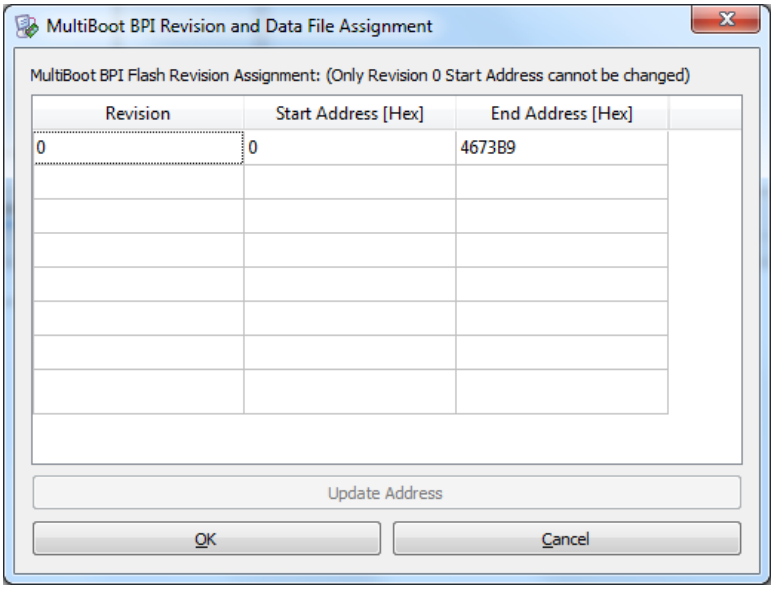

Рисунок A-10: Конфигурирование множественной загрузки в новом проекте Xilinx iMPACT

14. Общий вид состояния утилиты Xilinx iMPACT по окончании работы мастера конфигурации нового проекта приведен на рисунке рисунок A-11.

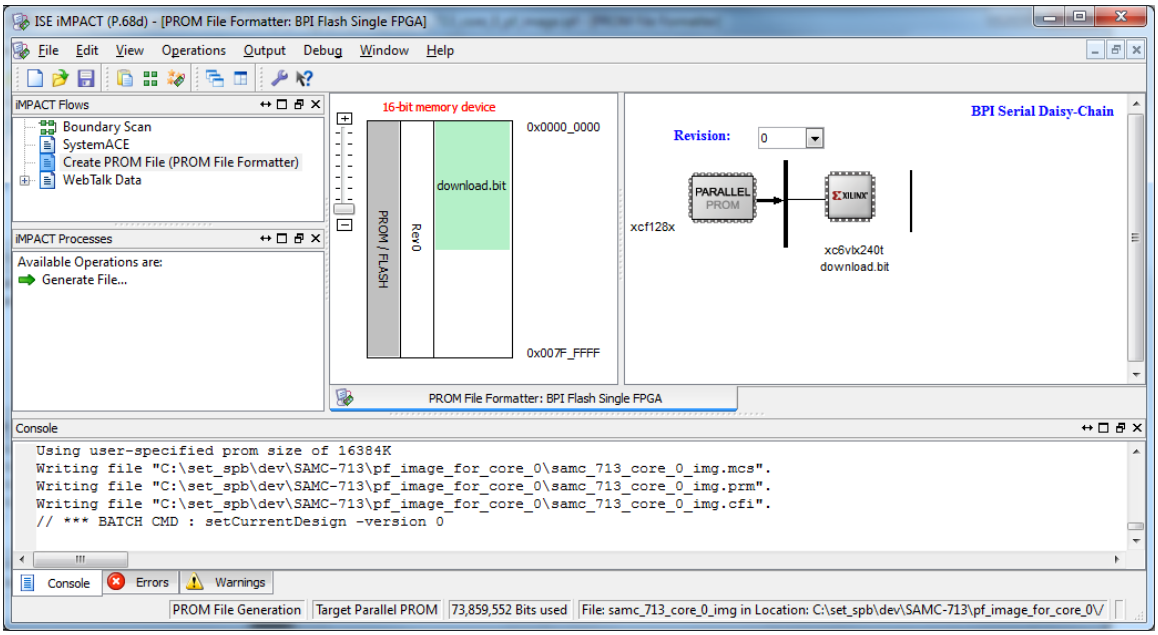

Рисунок A-11: Конфигурирование множественной загрузки в новом проекте Xilinx iMPACT

15. Создайте содержимое файла PROM созданного нового проекта, которое будет записывается в «Platform Flash», выбрав пункт меню «Operations > Generate File…» (рисунок A-12).

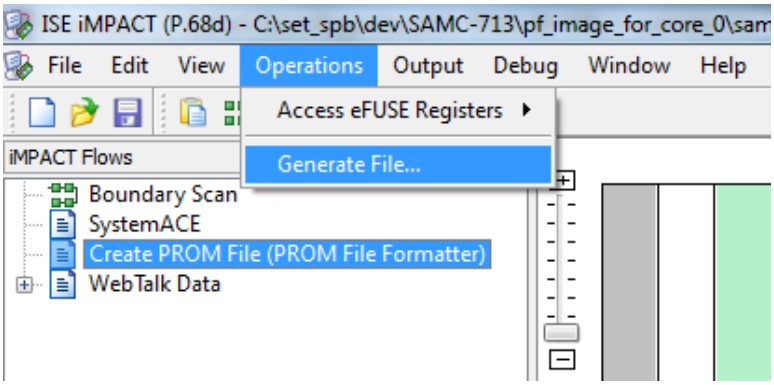

Рисунок A-12: Создание содержимого файла PROM в новом проекте Xilinx iMPACT

16. Сделайте двойной щелчок на «Boundary Scan», как показано на рисунке A-13

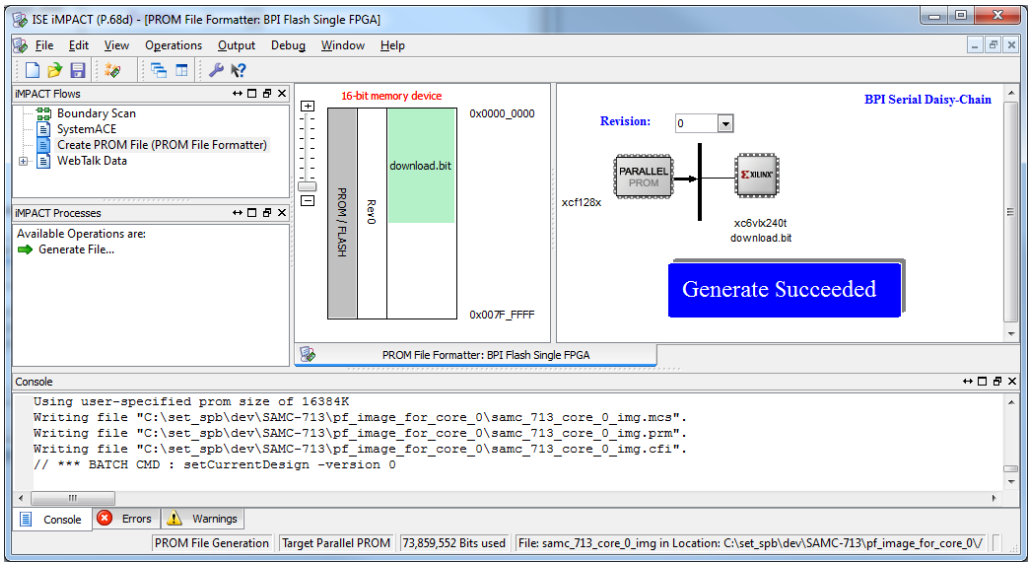

Рисунок A-13: Создание области сканирования устройств в Xilinx iMPACT

17. В открывшемся окне (рисунок A-14), в пространстве вкладки «Boundary Scan» нажатием правой кнопки мыши вызовите всплывающее меню и выбирете пункт «Initialize Chain».

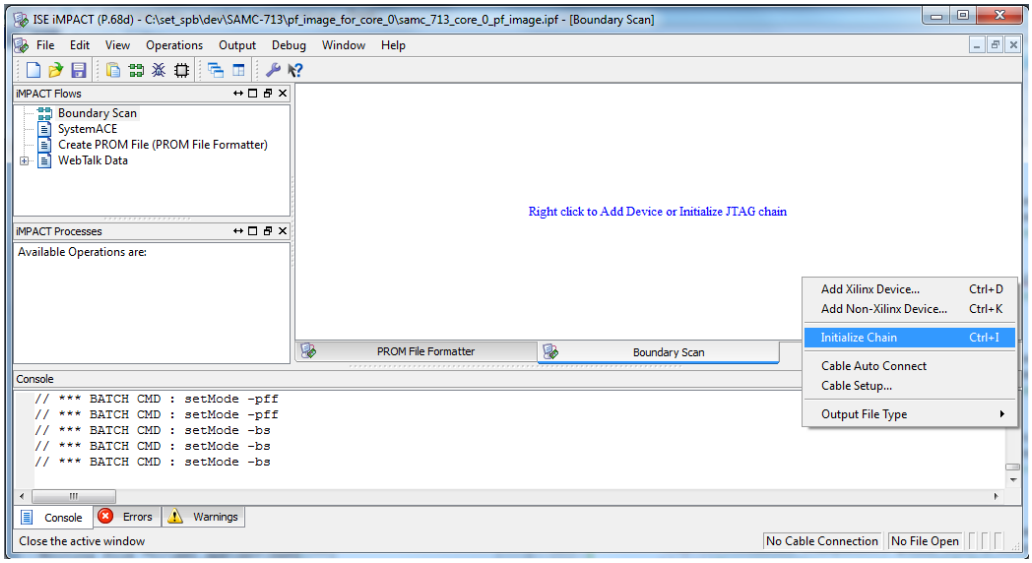

Рисунок A-14: Поиск активного FPGA устройства в Xilinx iMPACT

18. В открывшемся окне «Device Programming Properties - Device 1 Programming Properties» (рисунок A-15) нажмите кнопку «OK».

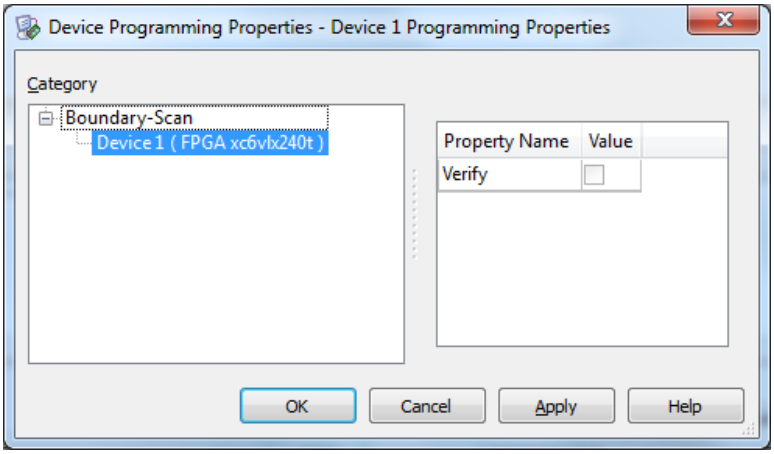

Рисунок A-15: Свойства программируемого FPGA устройства в Xilinx iMPACT

19. В открывшемся окне (рисунок A-16) наведите указатель мыши на изображение идентифицированного через JTAG интерфейс FPGA устройства, нажатием правой кнопки мыши вызовите всплывающее меню и выбирете пункт меню «Add SPI/BPI Flash…».

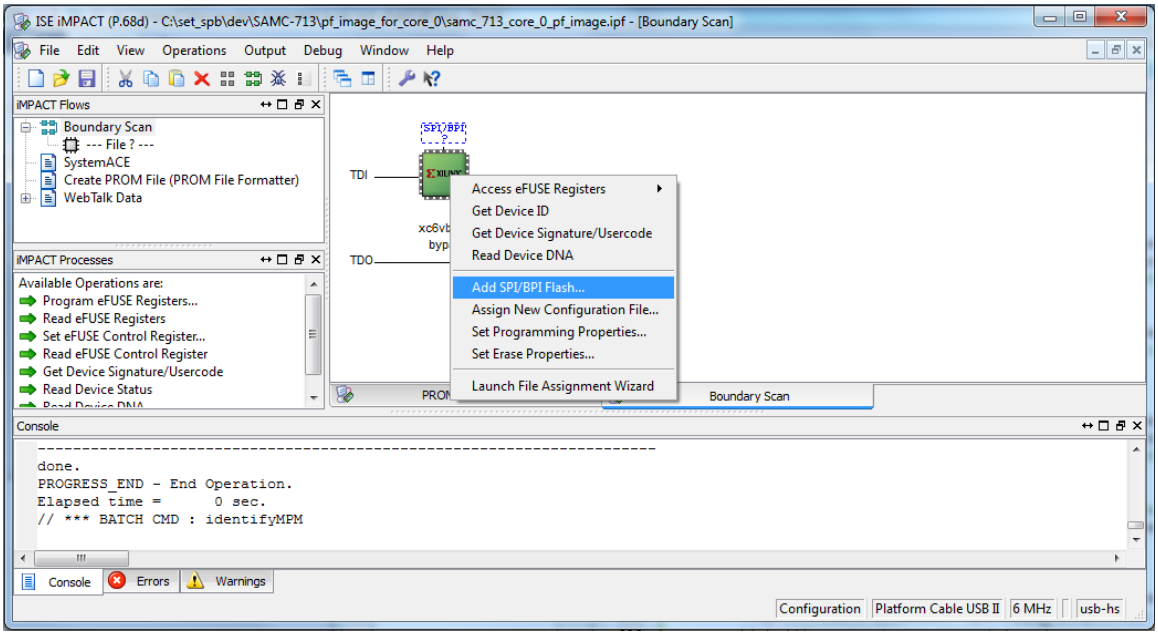

Рисунок A-16: Свойства программируемого FPGA устройства в Xilinx iMPACT

20. В открывшемся окне «Add PROM File» (рисунок A-17) выберите созданный ранее PROM файл и нажмите кнопку «Открыть».

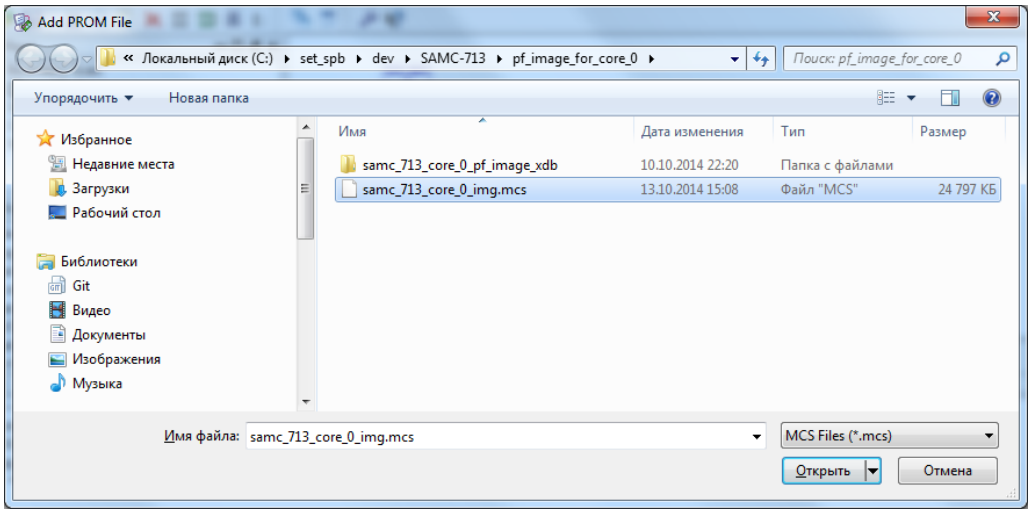

Рисунок A-17: Выбор PROM файла для программируемого FPGA устройства в Xilinx iMPACT

21. В открывшемся окне «Select Attached SPI/BPI» (рисунок A-18) нажмите кнопку «OK».

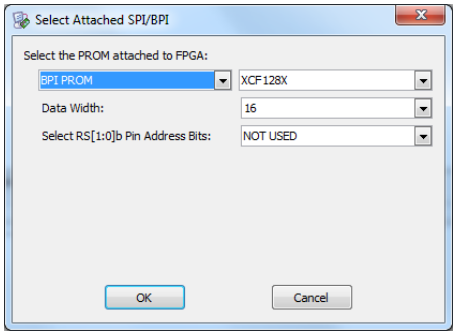

Рисунок A-18: Выбор типа подключенной к FPGA устройству флеш-памяти в Xilinx iMPACT

22. Наведите в пространстве вкладки «Boundary Scan» (рисунок A-19) указатель мыши на изображение флеш-микросхемы и нажатием правой кнопки мыши вызовите всплывающее меню, в котором выберите пункт «Program».

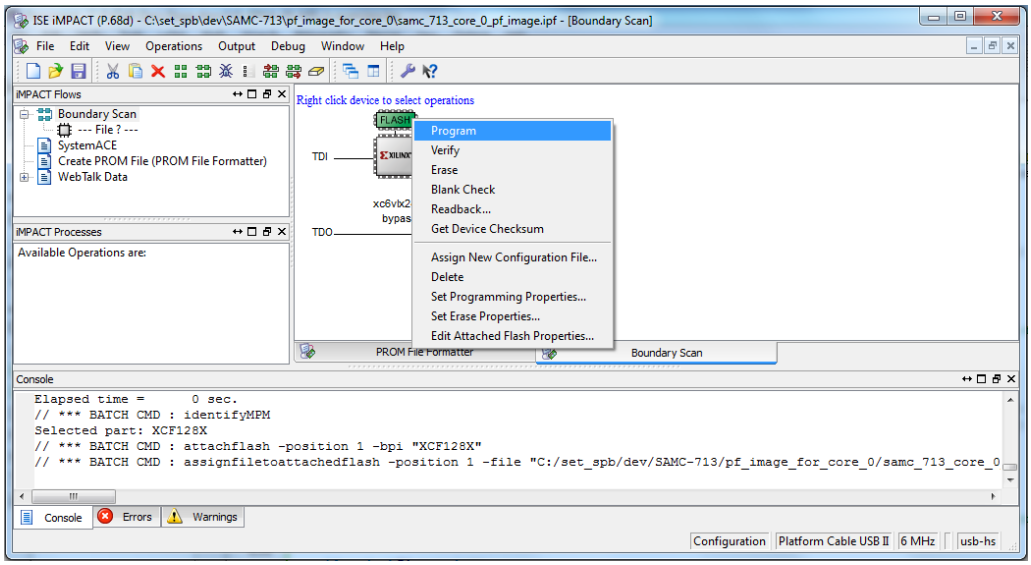

Рисунок A-19: Выбор типа подключенной к FPGA устройству флеш-памяти в Xilinx iMPACT

23. В открывшемся окне «Device Programming Properties - Device 1 Programming Properties» (рисунок A-20) нажмите кнопку «OK».

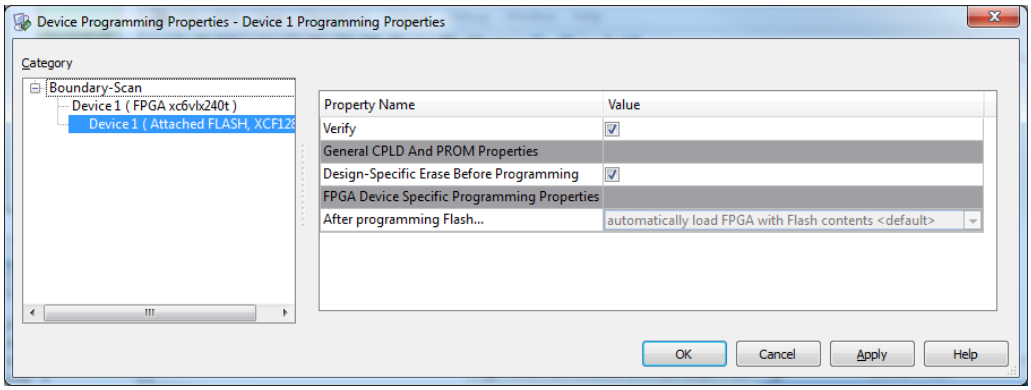

Рисунок A-20: Свойства для программирования FPGA устройства в Xilinx iMPACT

24. Запустится процесс программирования «Platform Flash», как показанно на рисунке A-21.

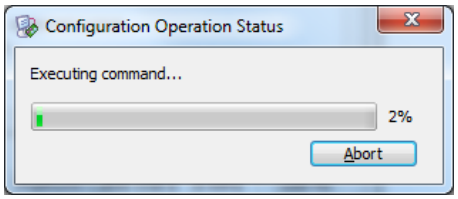

Рисунок A-21: Процесс программирования «Platform Flash» FPGA устройства в Xilinx iMPACT

25. Успешное окончание процесса программирования «Platform Flash» (рисунок A-22) говорит о том, что загрузчик MBL готов автоматически начинать свою работу при каждой загрузке аппаратной платформы в FPGA микросхему после подачи питания на FPGA модуль или при возникновении события аппаратного сброса микропроцессорного ядра MicroBlaze.

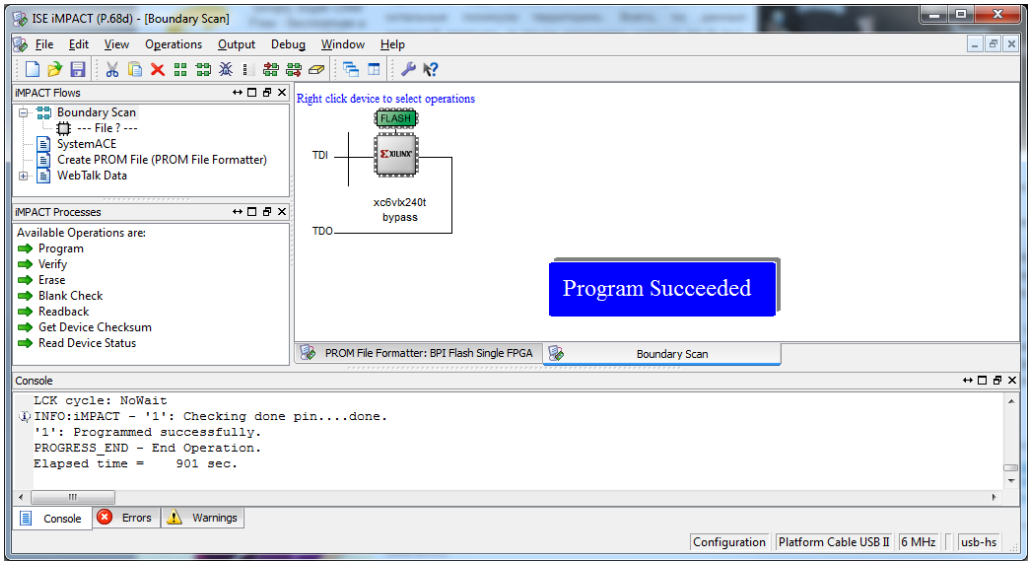

Рисунок A-22: Успешное окончание процесса программирования «Platform Flash» FPGA устройства в Xilinx iMPACT

Приложение Б Запись исполняемого образа приложения пользователя во флеш-память

Как было отмеченно в приложении А, на всех FPGA модулях производства ЗАО «Скан Инжиниринг Телеком» для хранения аппаратной платформы основной FPGA микросхемы существует «Platform Flash» или эмулируется работа этой флеш-памяти сервисной FPGA.

Для хранения загружаемого в SDRAM BRAM двоичного образа приложения пользователя в формате SREC загрузчиком MBL на FPGA модулях с непосредственным подключением «Platform Flash» к основной FPGA микросхеме используется сама «Platform Flash», а для FPGA модулей с эмуляцией работы «Platform Flash» используется отдельная SPI флеш-память, подключаемая непосредственно к основной FPGA микросхеме.

Б.1 Подготовка тестового приложения пользователя

Для того, чтобы продемонстрировать работу загрузчика MBL необходимо наличие приложения пользователя. ВЁкачестве примера такого приложения можно использовать проект «User_App_Demo», находящийся в подкаталоге «utils» (смотрите листинг 2.5).

Процедура Б-1. Подготовка тестового приложения пользователя

1. В среде Xilinx SDK выполните импорт готового проекта «User_App_Demo» в рабочее пространство, где осуществлялась компиляция загрузчика MBL, выполнив последовательность действий, описанную в процедуре 3-5, применительно к проекту «User_App_Demo».

   Например, для модуля «SAMC-713», на котором запускается аппаратная платформа с именем сборки «Core 0», путь к импортируемому проекту «User_App_Demo» выглядит следующим образом: «C:\set_spb\dev\SAMC-713\MBL\utils\user_app_demo».
2. Запустите процесс компиляции проекта «User_App_Demo», выполнив последовательность действий, описанную в процедуре 3-6, применительно к проекту «User_App_Demo».

   В результате компиляции проекта «User_App_Demo» будет получен двоичный образ в формате ELF.

Б.2 Формирование SREC файла

Процедура Б-2. Формирование SREC файла из двоичного образа тестового приложения пользователя

1. После окончания процесса компиляции двоичного образа приложения пользователя выбирете пункт меню «Xilinx Tools > Program Flash», как показано на рисунке Б-1.

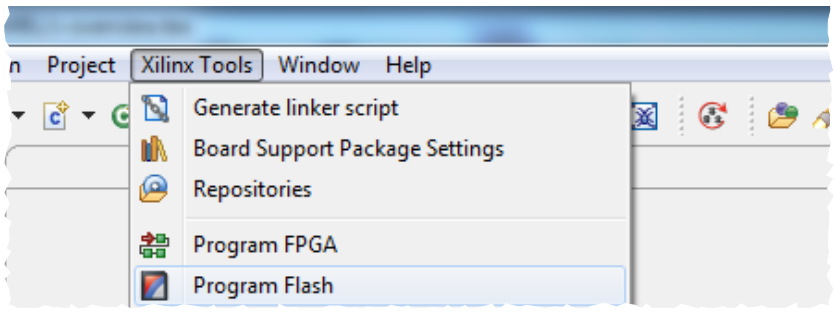

Рисунок Б-1: Выбор пункта меню программирования флеш-памяти в Xilinx SDK

2. В открывшемся окне «Flash Programmer Support Information» (рисунок Б-2) нажмите кнопку «OK».

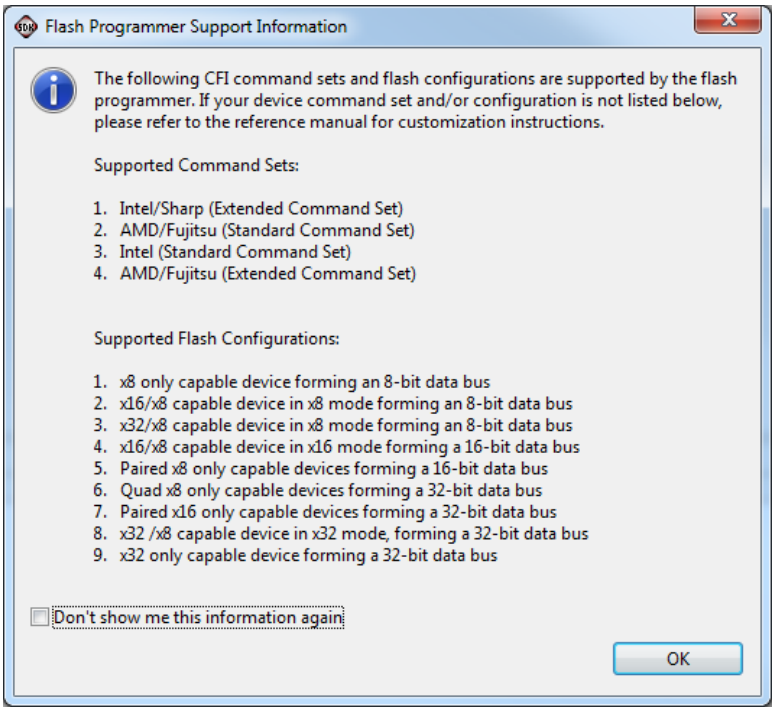

Рисунок Б-2: Поддерживаемая для программирования флеш-памяти в Xilinx SDK

3. В открывшемся окне «Program Flash Memory» (рисунок Б-3) нажмите кнопку «Browse» и укажите место расположения двоичного образа приложения пользователя в формате ELF.

   Например, для модуля «SAMC-713», на котором запускается аппаратная платформа с именем сборки «Core 0», путь к полученному в результате компиляции проекта «User_App_Demo» двоичному образу приложения пользователя будет выглядеть следующим образом: «C:\set_spb\dev\SAMC-713\MBL\utils\user_app_demo\Debug\User_App_Demo.elf».

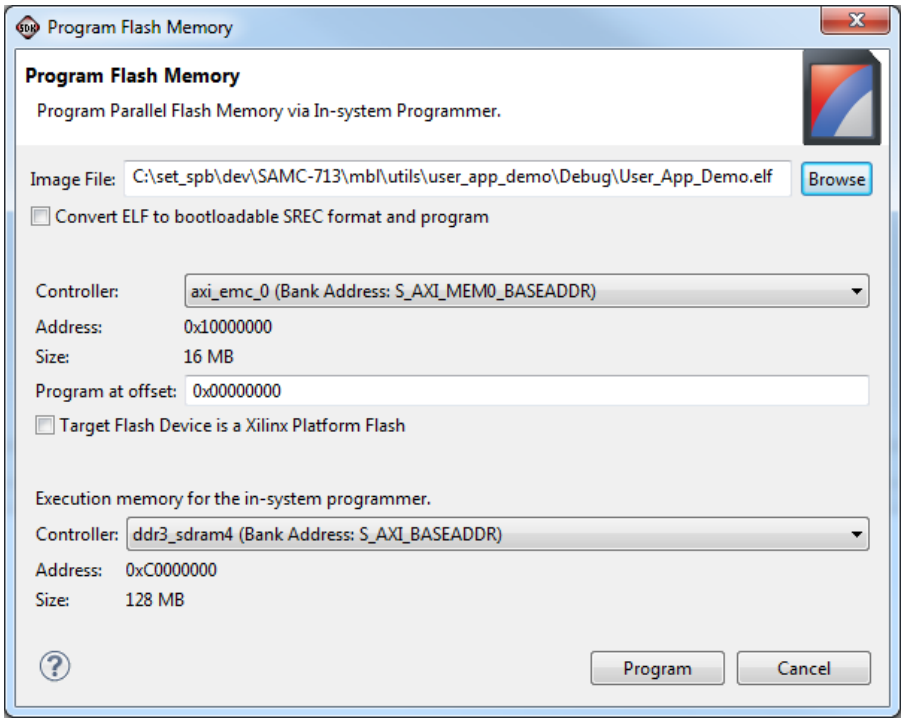

Рисунок Б-3: Выбор ELF файла двоичного образа приложения пользователя в Xilinx SDK

4. В открытом окне «Program Flash Memory» (рисунок Б-3) сделайте выбор в селекторе «Convert ELF bootloadable SREC format and program», установив галку. После установки галки среда Xilinx SDK автоматически сформирует файл SREC из выбранного на предыдущем шаге двоичного образа

приложения пользователя и сохранит файл SREC в каталоге «cache» проекта аппаратной платформы текущего рабочего пространства.

Например, для модуля «SAMC-713», на котором запускается аппаратная платформа с именем сборки «Core 0», путь к полученному SREC двоичному образу приложения пользователя проекта «User_App_Demo» будет выглядеть следующим образом: «C:\set_spb\dev\SAMC-713\core_0\SDK\SDK_Export\core_0_hw_platform\cache\User_App_Demo.elf.srec».

5. Закройте текущее окно (рисунок Б-3), нажав кнопку «Cancel».

B.3 Запись SREC файла в «Platform Flash»

Описанные в данном разделе действия будут относиться к FPGA модулям с непосредственным подключением «Platform Flash» к основной FPGA микросхеме. Примером такого FPGA модуля могут выступить SAMC-713 или SVP-713.

Процедура В-3. Формирование SREC файла из двоичного образа тестового приложения пользователя

1. После окончания процесса формирования файла двоичного образа приложения пользователя в формате SREC загрузите аппаратную платформу в FPGA через JTAG интерфейс. Для этого выберите пункт меню «Xilinx Tools > Program FPGA», как показано на рисунке 4-1;
2. В открывшемся окне «Program FPGA» (рисунок Б-4) выберите в таблице «Software Configuration» загружаемый в FPGA микросхему двоичный образ «bootloop».

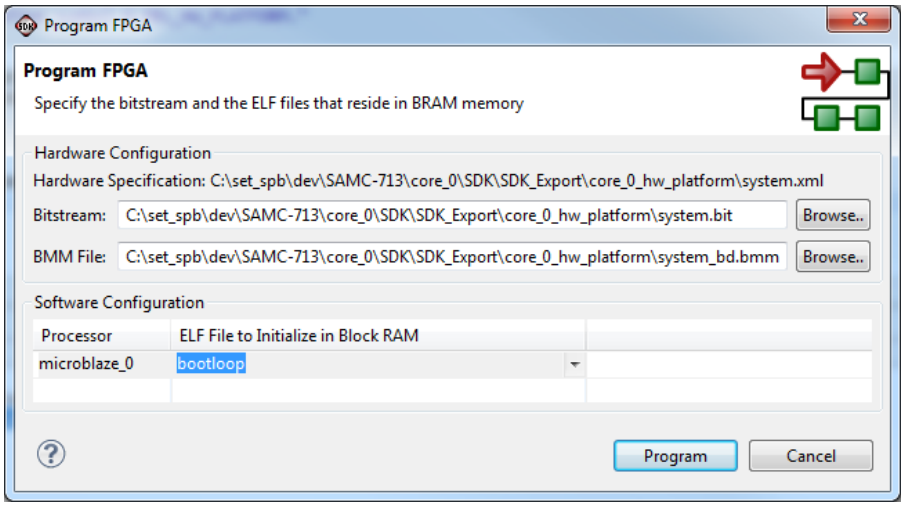

Рисунок Б-4: Выбор загружаемого в FPGA микросхему двоичного образа загрузчика MBL в Xilinx SDK

3. В текущем окне запустите процесс загрузки аппаратной платформы и двоичного образа «bootloop» в FPGA микросхему , нажав кнопку «Program».
4. После окончания процесса загрузки в FPGA микросхему аппаратной платформы и двоичного образа «bootloop», выберите пункт меню «Xilinx Tools > Program Flash», как показано на рисунке Б-1.
5. В открывшемся окне «Flash Programmer Support Information» (рисунок Б-2), нажмите кнопку «OK».
6. В открывшемся окне «Program Flash Memory» (рисунок Б-5) нажмите кнопку «Browse…» и укажите место расположения двоичного образа приложения пользователя в формате SREC.

   Например, для модуля «SAMC-713», на котором запускается аппаратная платформа с именем сборки «Core 0», путь к полученному в результате процесса формирования файла двоичного образа приложения пользователя проекта «User_App_Demo» в формате SREC, будет выглядеть следующим образом: «C:\set_spb\dev\SAMC-713\core_0\SDK\SDK_Export\core_0_hw_platform\cache\User_App_Demo.elf.srec».

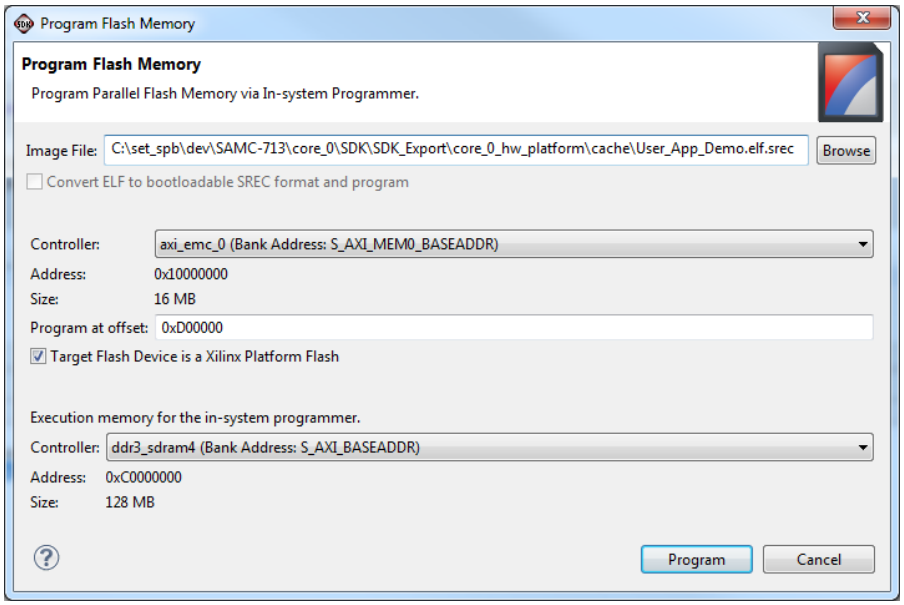

Рисунок Б-5: Выбор SREC файла двоичного образа приложения пользователя в Xilinx SDK

7. В текущем открытом окне «Program Flash Memory» (рисунок Б-5) выполните следующие действия:
   - установите в поле «Program at offset» значение смещения в «0xD00000»;
   - сделайте выбор в селекторе «Target Flash Device is a Xilinx Platform Flash», установив галку.

   Указанное смещение в поле «Program at offset» соответствует размещению в «Platform Flash» двоичного образа в формате SREC приложения пользователя. Значение поля «Program at offset» определяется относительно адреса начала первого сегмента памяти самой «Platform Flash» и может соответствовать любому адресу начала сегмента до которого располагается содержимое двоичного образа, загружаемого в FPGA микросхему аппаратной платформы. На рисунок А-11 занимаемое аппаратной платформой пространство в «Platform Flash» помеченно зеленым цветом как «download.bit». Диапазон адресов размещения содержимого «download.bit» видно из значений таблицы, представленной на рисунке А-10.
8. В текущем открытом окне «Program Flash Memory» (рисунок Б-5) запустите процесс программирования «Platform Flash», нажав кнопку «Program».

B.4 Запись SREC файла в SPI флеш-память

Описанные в данном разделе действия будут относиться к FPGA модулям с эмуляцией работы «Platform Flash» для основной FPGA микросхемы. У таких модулей для хранения двоичного образа приложения пользователя в формате SREC используется отдельная SPI флеш-память, подключаемая непосредственно к основной FPGA микросхеме. Примером такого FPGA модуля может выступить SVP-726.

Для записи двоичного образа приложения пользователя в формате SREC в SPI флеш-памяти необходимо воспользоваться сервисной утилитой «Spi_Flash_Writer» из каталога «utils» (смотрите листинг 2.5).

B.4.1 Компиляция утилиты «Spi_Flash_Writer»

Процедура B-4. Компиляция утилиты «Spi_Flash_Writer»

1. В среде Xilinx SDK выполните импорт проекта «Spi_Flash_Writer» в рабочее пространство, где осуществлялась компиляция загрузчика MBL, выполнив последовательность действий, описанную в процедуре 3-5, применительно к проекту «Spi_Flash_Writer».

   Например, для модуля «SVP-726», на котором запускается аппаратная платформа с именем сборки «Core 1», путь к импортируемому проекту «Spi_Flash_Writer» выглядит следующим образом: «C:\set_spb\dev\SVP-726\MBL\utils\spi_flash_writer».
2. Выбирете пункт меню «Xilinx Tools > Board Support Package Settings», как показано на рисунке Б-6.

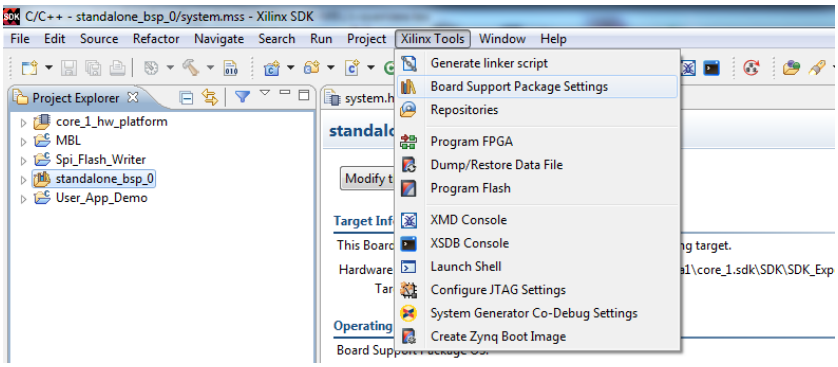

Рисунок Б-6: Выбор BSP пакета поддержки аппаратной платформы в Xilinx SDK

3. В открывшемся окне «Board Support Package Settings» (рисунок Б-7) подключите к BSP проекту поддержки аппаратной платформы библиотеку «Xilinx In-system and Serial Flash Library» и нажмите кнопку «OK».

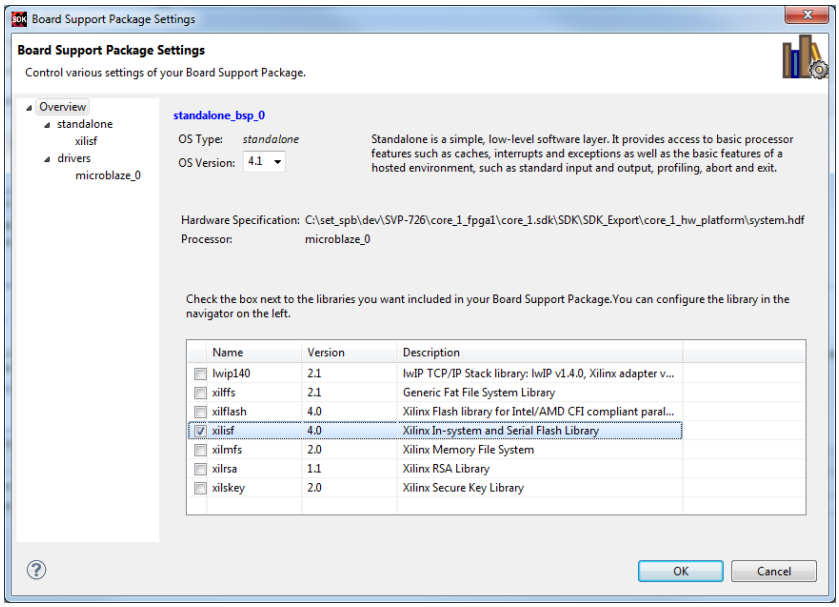

Рисунок Б-7: Включение библиотеки «xilisf» в BSP пакет поддержки аппаратной платформы в Xilinx SDK

4. Запустите процесс компиляции проекта «Spi_Flash_Writer» в режиме «Debug», выполнив последовательность действий, описанную в процедуре 3-6, применительно к проекту «Spi_Flash_Writer».

   В результате компиляции проекта «Spi_Flash_Writer» будет получен двоичный образ в формате ELF.

B.4.2 Запуск утилиты «Spi_Flash_Writer» в режиме отладки

Процедура B-5. Запуск утилиты «Spi_Flash_Writer» и запись SREC файла в SPI флеш-память

1. После окончания процесса компилирования утилиты «Spi_Flash_Writer» загрузите ее в режиме отладки через JTAG интерфейс. Для этого выбирете пункт меню «Xilinx Tools > Program FPGA», как показано на рисунке 4-1.
2. В открывшемся окне «Program FPGA» (рисунок Б-8) нажмите кнопку «Select» для выбора FPGA микросхемы с которой будет осуществляться работа.

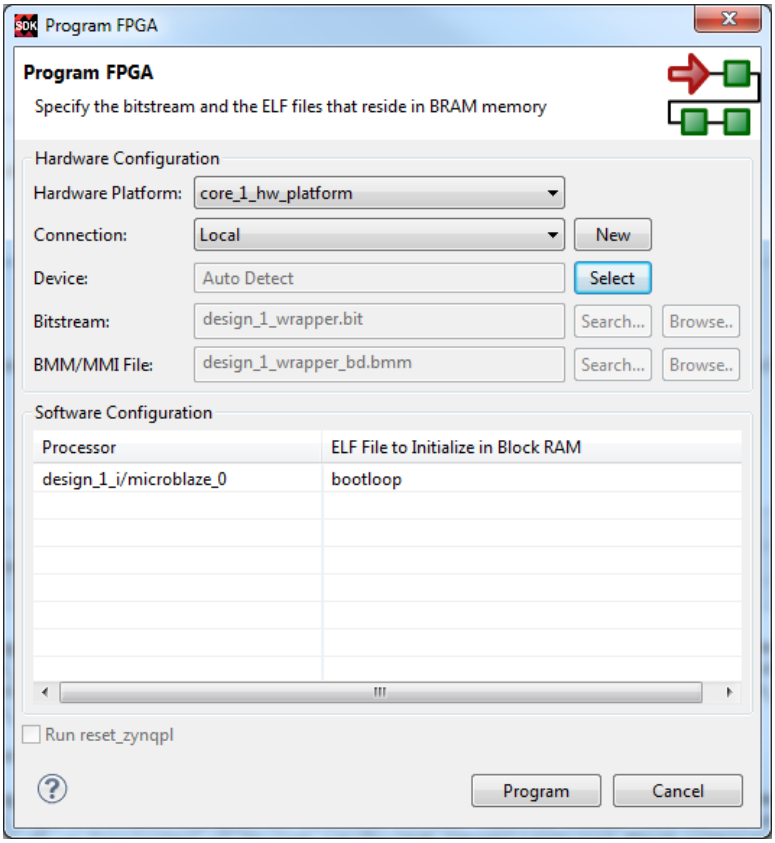

Рисунок Б-8: Программирование FPGA микросхемы через JTAG интерфейс в Xilinx SDK

3. В открывшемся окне «Select the device» (рисунок Б-9) выберите FPGA микросхему с которой будет осуществляться работа и нажмите кнопку «OK».

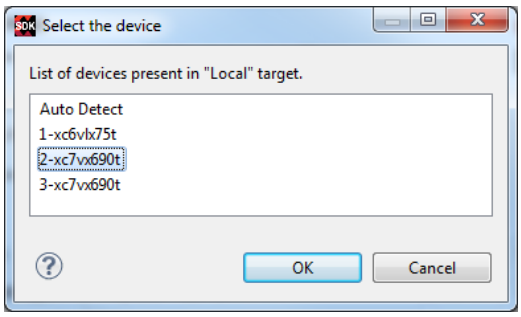

Рисунок Б-9: Выбор для программирования FPGA микросхемы через JTAG интерфейс в Xilinx SDK

4. В текущем открытом окне «Program FPGA» (рисунок Б-8) запустите процесс программирования выбранной FPGA микросхемы, нажав кнопку «Program».
5. После окончания процесса программирования выбранной FPGA микросхемы создайте цель запуска в режиме отладки двоичного образа утилиты «Spi_Flash_Writer». Для этого выбирете пункт меню «Run > Debug As > Launch on Hardware (GDB)», как показано на рисунке Б-10.

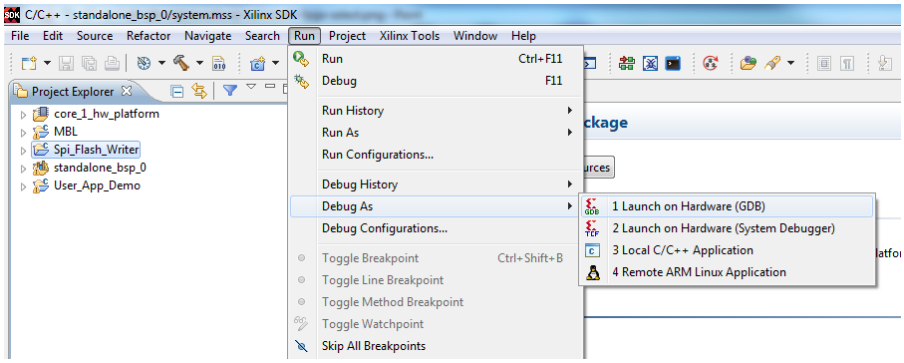

Рисунок Б-10: Создание цели запуска в режиме отладки утилиты «Spi_Flash_Writer» через JTAG интерфейс в Xilinx SDK

6. В открывшемся окне «FPGA Configuration» (рисунок Б-11) нажмите кнопку «OK».

   Данная ошибка возникает всегда, когда в первый раз создаются цели запуска при работе с FPGA модулями, содержащими более одной микросхемы FPGA в цепочке подключения через JTAG интерфейс.

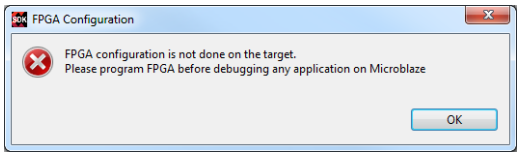

Рисунок Б-11: Ошибка цели запуска через JTAG интерфейс в Xilinx SDK

7. Выбирете пункт меню «Debug Configurations…», как показано на рисунке Б-12.

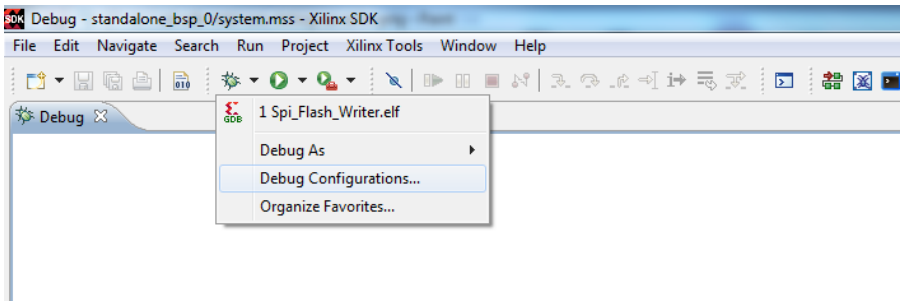

Рисунок Б-12: Выбор конфигураций целей запуска в режиме отладки в Xilinx SDK

8. В открывшемся окне «Debug Configurations» (рисунок Б-13) выполните следующие настройки:
   - выберите FPGA микросхему с которой будет вестись работа, нажав кнопку «Select»;
   - установите значение «Reset Entire System»;
   - сделайте выбор, установив галку напротив «Program FPGA».

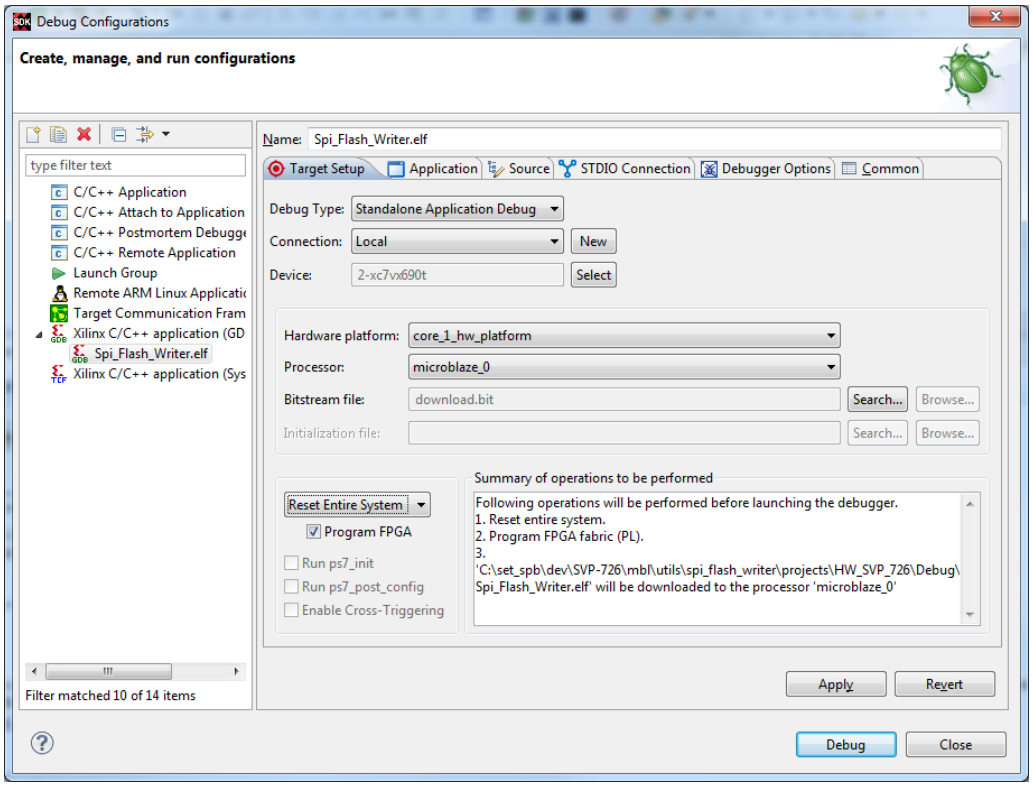

Рисунок Б-13: Настройка конфигураций целей запуска в режиме отладки утилиты «Spi_Flash_Writer» в Xilinx SDK

9. В текущем открытом окне «Debug Configurations» (рисунок Б-13) нажмите кнопку «Apply» и запустите процесс отладки утилиты «Spi_Flash_Writer» через JTAG интерфейс, нажав кнопку «Debug».

   Успешным входом в процесс отладки утилиты «Spi_Flash_Writer» будет содержание главного окна среды разработки Xilinx SDK, которое соотвествует изображению рисунка Б-14.

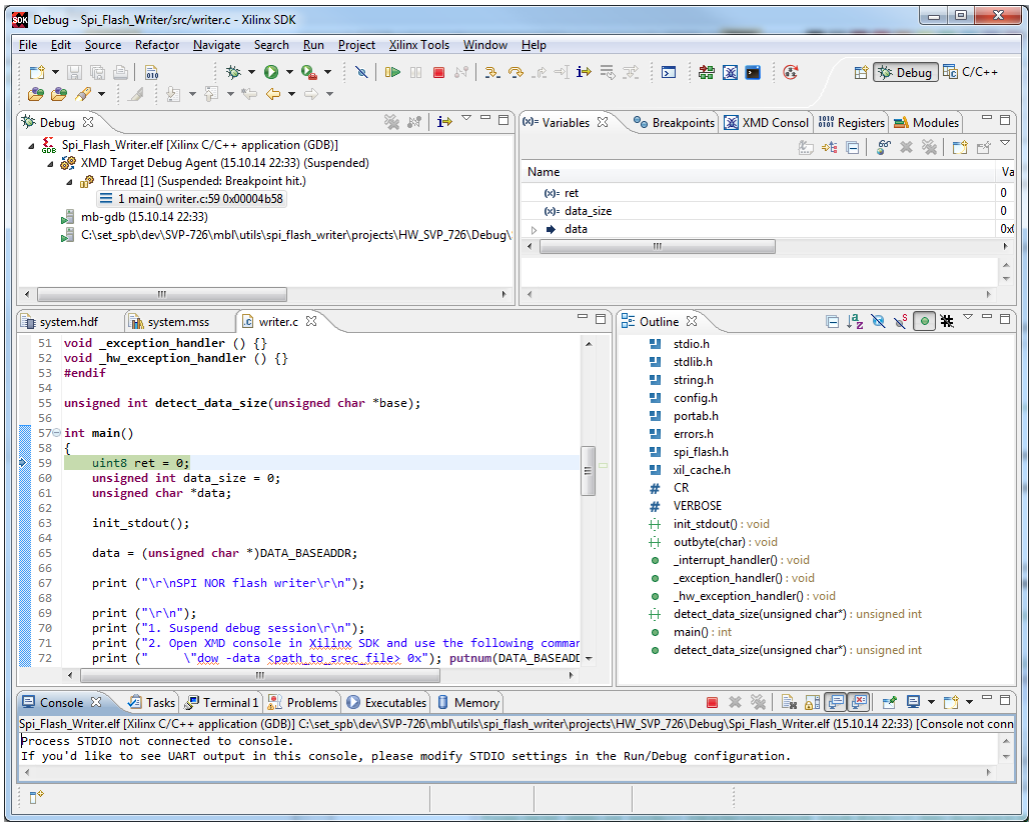

Рисунок Б-14: Окно отладки утилиты «Spi_Flash_Writer» в Xilinx SDK

10. Запустите работу утилиты «Spi_Flash_Writer», нажав кнопку «Resume», показанную на рисунке Б-15.

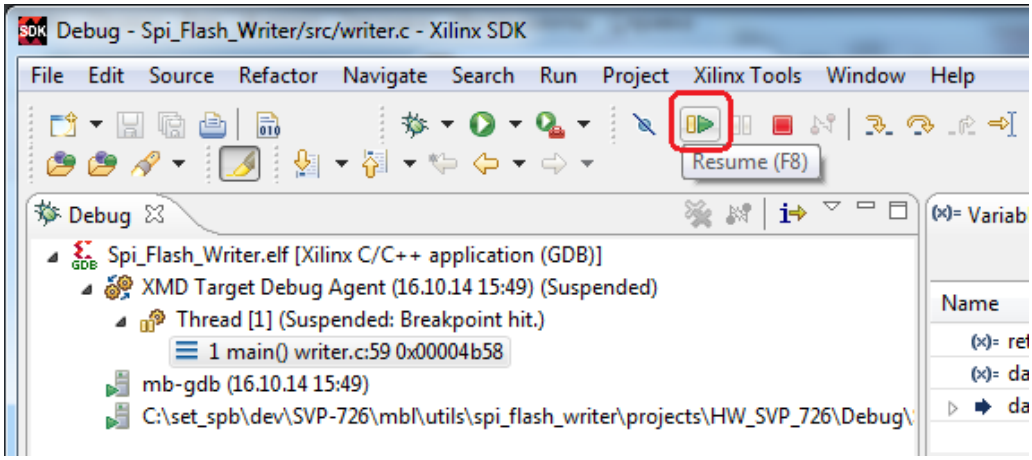

Рисунок Б-15: Запуск работы утилиты «Spi_Flash_Writer» в Xilinx SDK

B.4.3 Запись SREC файла в SPI флеш-память

Процедура B-6. Запись SREC файла в SPI флеш-память утилитой «Spi_Flash_Writer»

1. После запуска работы утилиты «Spi_Flash_Writer» в работающей на персональном компьютере терминальной программе появятся текстовые сообщения (рисунок Б-16), информирующие о необходимости выполнения последовательности действий.

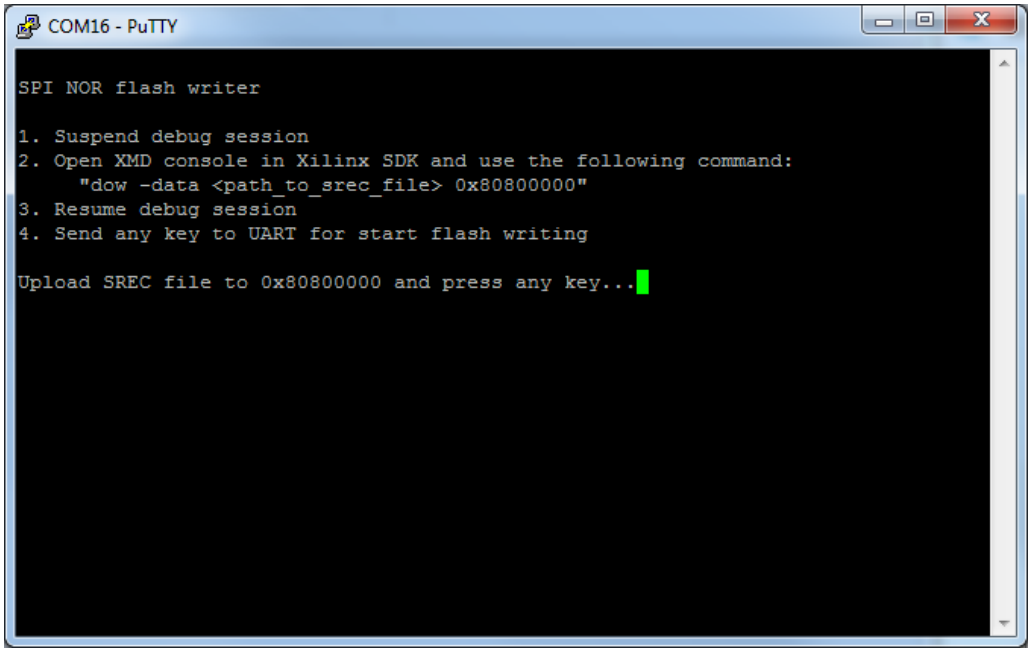

Рисунок Б-16: Терминальный вывод сообщений утилиты «Spi_Flash_Writer»

2. Приостановите работу утилиты «Spi_Flash_Writer», нажав кнопку «Suspend», как показано на рисунке Б-17.

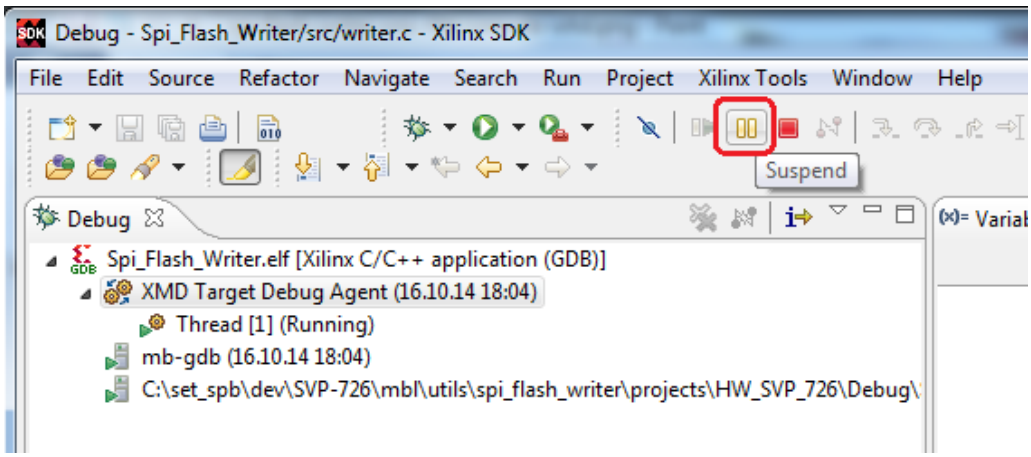

Рисунок Б-17: Приостановка работы утилиты «Spi_Flash_Writer» в Xilinx SDK

3. Перейдите в окно «XDM Console», выбрав пункт меню «Window > Show View > XDM Console», как показано на рисунке Б-18.

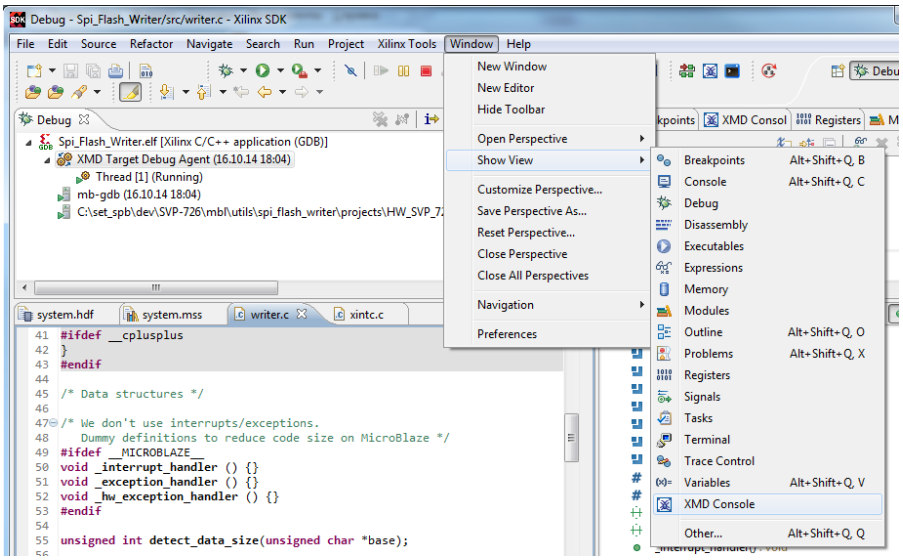

Рисунок Б-18: Переход в «XDM Console» среды Xilinx SDK

4. Введите в командную строку «XDM Console» команду загрузки в SDRAM DDR FPGA модуля двоичного образа приложения пользователя в формате SREC. Формат команды загрузки выглядит следующим образом:

```
dow -data <path_to_srec_file> 0x80800000
```

Например, для модуля «SVP-726», на котором запускается аппаратная платформа с именем сборки «Core 1», путь к двоичному образу приложения пользователя в формате SREC выглядит следующим образом: «C:\set_spb\dev\SVP-726\core_1_fpga1\core_1.sdk\SDK\SDK_Export\core_1_hw_-platform\cache\User_App_Demo.elf.srec».

Результат загрузки будет виден в «XDM Console», смотрите рисунок Б-19.

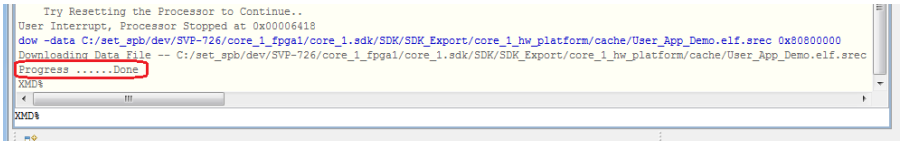

Рисунок Б-19: Результат загрузки двоичного образа приложения пользователя в формате SREC в «XDM Console» среды Xilinx SDK

5. Продолжите работу утилиты «Spi_Flash_Writer», нажав кнопку «Resume», как показано на рисунке Б-20.

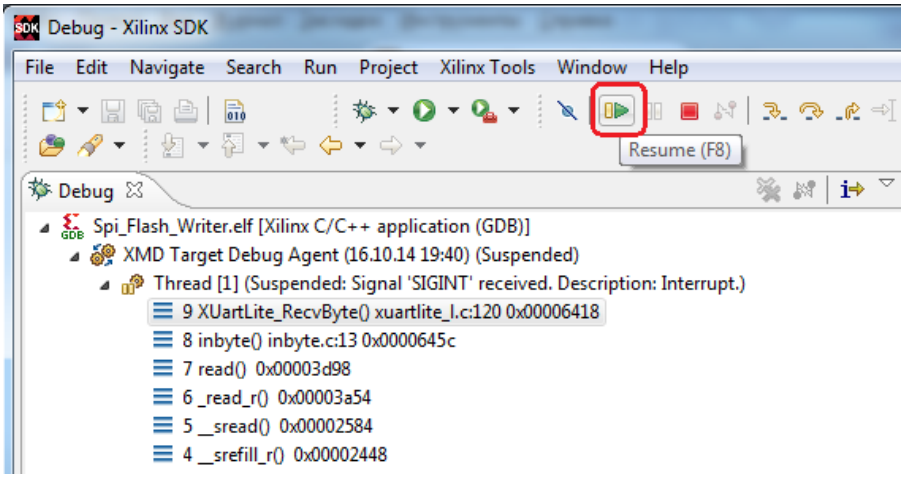

Рисунок Б-20: Продолжение работы утилиты «Spi_Flash_Writer» в Xilinx SDK

6. В окне, работающей на персональном компьютере, терминальной программы (рисунок Б-16) выполните действие за номером четыре, нажав любую клавишу на клавиатуре.

   После нажатия любой клавиши запустится процесс записи в SPI флеш-память содержимого ранее загруженного в SDRAM DDR FPGA модуля двоичного образа приложения пользователя в формате SREC.

   Результат успешного выполнения записи приведен на рисунке Б-21.

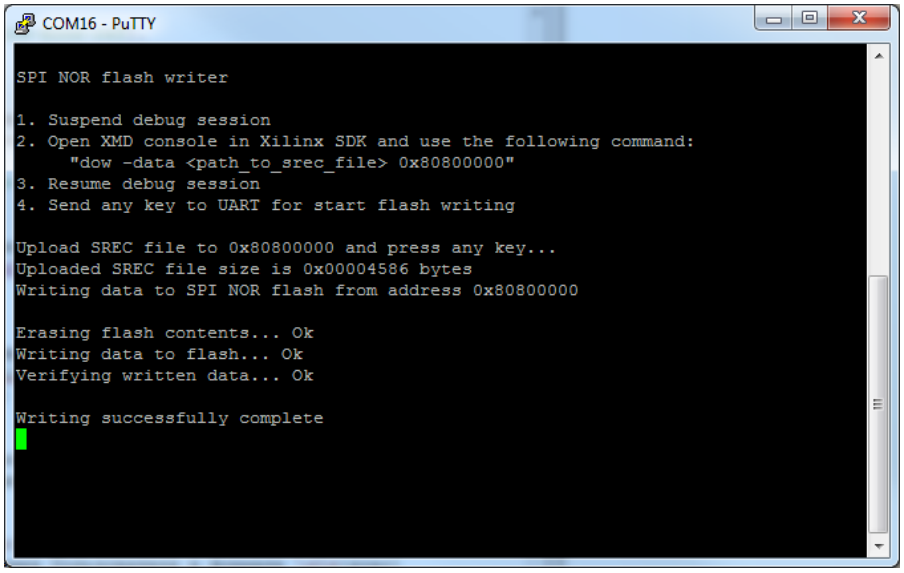

Рисунок Б-21: Терминальный вывод сообщений об успешном окончании работы утилиты «Spi_Flash_Writer»

7. Остановите процесс отладки утилиты «Spi_Flash_Writer», нажав кнопку «Terminate», как показано на рисунке Б-22.

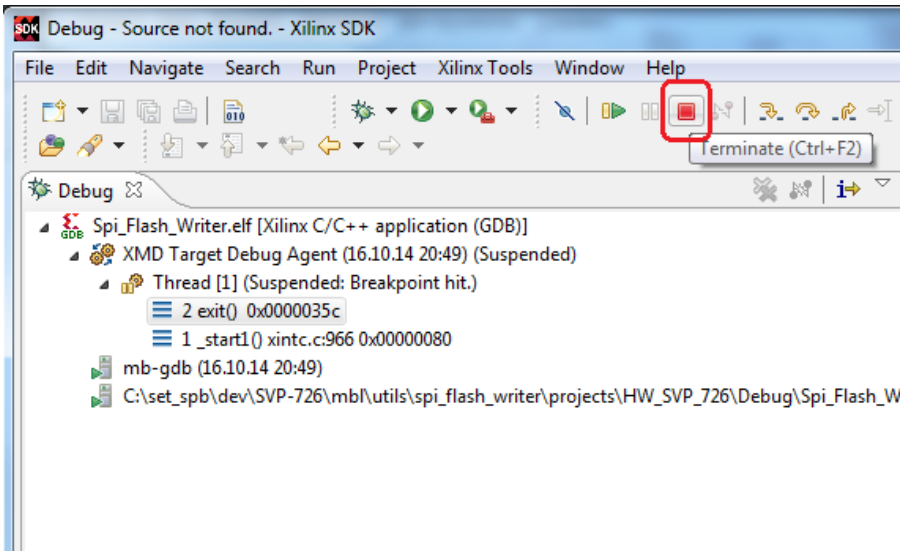

Рисунок Б-22: Остановка процесса отладки утилиты «Spi_Flash_Writer» в Xilinx SDK

Список литературы

1. Инфраструктура IP-ядер «Микропроцессорная система на MicroBlaze». Руководство пользователя. UG-IP-IS-MBS (цит. на с. 7, 8).