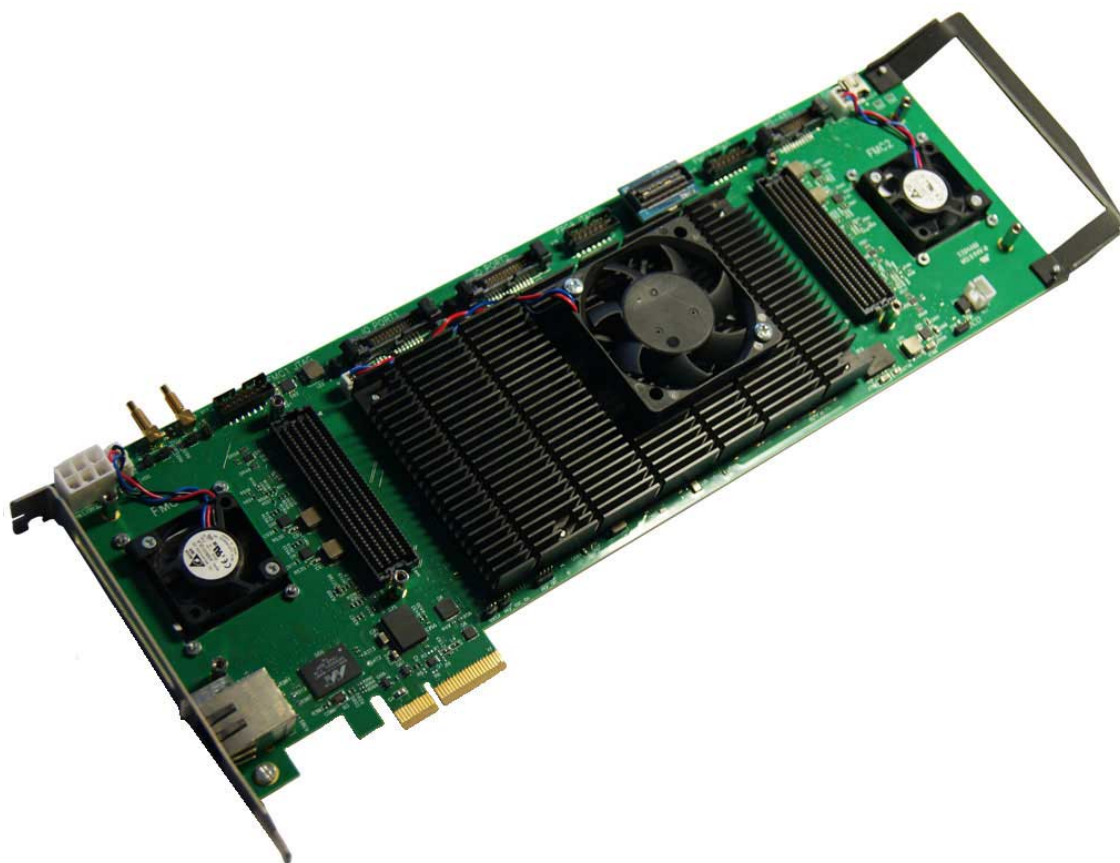




TS-PCIE. Загрузчик IBL

Руководство пользователя

Версия 1.0



Код документа: UG-TS-PCIE-IBL
Дата сборки: 14 марта 2016 г.
Листов в документе: 44

Содержание

Перечень рисунков	3
Перечень таблиц	3
Перечень листингов	3
Перечень процедур	3
Перечень сокращений и условных обозначений	5
1 Общие сведения	6
2 Сборка образа IBL из исходных кодов	7
2.1 Установка MinGW в Windows системе	7
2.2 Конфигурация окружения сборки	10
2.3 Сборка загрузчика	10
3 Скрипт для записи EEPROM и NOR флеш памяти модуля TS-PCIE	12
3.1 Структура каталога «Program»	13
3.2 Работа со скриптом записи NOR флеш и EEPROM памяти	15
3.3 Запись образов в NOR флеш память	16
3.4 Запись образов в EEPROM память	21
4 Конфигурация IBL	23
5 Импорт и запуск целевой конфигурации модуля	30
6 Подготовка образов приложений для загрузки на нескольких ядрах процессора	33
6.1 Компоненты MAD Utils	33
6.2 Режимы работы MAD Utils	34
6.3 Работа с MAD Utils в режиме «Prelinker bypass mode»	34
6.4 Конфигурация MAP Tool	35
6.5 Конфигурационный файл развертывания	35
6.6 Запуск MAP Tool	37
6.7 Загрузка образа	37
Приложение А: Примеры работы скрипта записи EEPROM и NOR флеш памяти	38
Приложение Б: Разделение вывода сообщений (CIO) ядер процессоров	40
Приложение В: Выбор режима загрузки IBL	43
Список литературы	44

Перечень рисунков

2-1	Установка MinGW. Выбор каталога репозитариев	7
2-2	Установка MinGW. Окно согласия с лицензией	8
2-3	Установка MinGW. Выбор пути установки	8
2-4	Установка MinGW. Выбор устанавливаемых компонентов	9
2-5	Приглашение командной строки MinGW Shell	9
2-6	Редактирование файла «setupenvMsys.sh» в редакторе vim	10
3-1	Состояния светодиодов на плате модуля TS-PCIE во время записи	17
3-2	Организация EEPROM памяти на модуле TS-PCIE	21
4-1	Подключение к ядру «DSP1_C6670»	23
4-2	Окно загрузки кода на ядро процессора	23
4-3	Внешний вид окна «Debug» после загрузки кода на ядра процессоров	24
4-4	Внешний вид окна «Console» после запуска кода на процессоре	24
4-5	Пункт главного меню для вызова окна управления GEL файлами	25
4-6	Окно управления GEL файлами	25
4-7	Окно управления GEL файлами с загруженным файлом «i2cConfig.gel»	25
4-8	Пункт главного меню «Scripts > SET TS-PCIE IBL»	26
4-9	Внешний вид окна «Console» после выполнения записи конфигурации в EEPROM память	26
5-1	Пункт меню для отображения окна целевых конфигураций	30
5-2	Меню импорта целевой конфигурации	30
5-3	Окно выбора файла для импорта целевой конфигурации	31
5-4	Окно выбора способа импорта файла целевой конфигурации	31
5-5	Запуск целевой конфигурации	31
5-6	Список ядер процессоров модуля TS-PCIE	32
6-1	Схема работы MAD в режиме «Prelinker bypass mode»	34
Б-1	Контекстное меню целевой конфигурации	40
Б-2	Окно настроек целевой конфигурации	40
Б-3	Открытие второго окна «Console»	41
Б-4	Два окна «Console»	41
Б-5	Выбор ядра для отображения вывода в окне «Console»	42

Перечень таблиц

2-1	Файлы, создаваемые при сборке загрузчика	10
3-1	Параметры командной строки скрипта «tspcie_program.js»	15
4-1	Основные конфигурационные параметры файла «i2cConfig.gel»	28
6-1	Параметры конфигурационного файла MAP Tool	35
6-2	Параметры файла конфигурации развертывания	35
В-1	Положение переключателей модуля TS-PCIE для установки режимов загрузки IBL	43

Перечень листингов

3-1	Скрипт «tspcie_program.bat»	12
3-2	Скрипт «tspcie_program.sh»	12
3-3	Структура каталога «Program»	13
3-4	Вывод скрипта «tspcie_program.bat», запущенного без параметров	15
3-5	Вывод в UART загрузки демонстрационного приложения веб-сервера с NOR флеш памяти	18
3-6	Вывод в UART при запуске теста платформы	18
4-1	Фрагмент GEL файла «i2cConfig.gel» с конфигурационными параметрами загрузчика IBL для модуля TS-PCIE	26
6-1	Пример конфигурационного файла для MAP Tool	35
6-2	Пример файла конфигурации развертывания	36
А-1	Вывод в терминал при запуске команды «tspcie_program.bat NOR all»	38
А-2	Вывод в терминал при запуске команды «tspcie_program.bat EEPROM all»	39

Перечень процедур

3-1	Запись образов в NOR флеш память	16
3-2	Запись образа демонстрационного приложения веб-сервера в NOR флеш память и его загрузка	16
3-3	Запись образов в EEPROM память	21
3-4	Запись образа загрузчика IBL в EEPROM память и его загрузка	21

Перечень сокращений и условных обозначений

BOOTP	Bootstrap Protocol	28, 29, 37
CCS	Code Composer Studio	6, 7, 10–14, 16, 31, 32, 40, 41
CGT	Code Generation Tools	35
CIO	Console Input/Output	2, 14, 15, 32, 40
COFF	Common Object File Format	29
DDR	Double Data Rate	13, 16, 33, 34
DSO	Dynamic Shared Object	33
DSS	Debug Server Scripting	11, 12
EEPROM	Electrically Erasable Programmable Read-Only Memory	2–4, 6, 7, 10–17, 21–24, 26, 34, 38, 39
ELF	Executable and Linkable Format	29, 33
GEL	General Extension Language	3, 10, 24–26
I²C	Inter-Integrated Circuit	7, 15, 17, 21, 34
IBL	Intermediate Boot Loader	2–4, 6, 7, 9, 10, 13, 16, 17, 21–24, 26, 28, 29, 33, 34, 37, 43
IP	Internet Protocol	29
JSON	Java Script Object Notation	35
MAC	Media Access Control	29
MAD	Multicore Application Deployment	2, 3, 33–35, 37
MAP	Multiple Application Pre-linker	2, 33–37
MCSDK	MultiCore Software Development Kit	33, 35, 37
NML	No Man's Land	35
NOR	Not OR	2–4, 6, 12–18, 22, 34, 37, 38
OFD	Object File Dump	35
ROMFS	Read-Only Memory File System	33–36
ROM	Read-Only Memory	34
TFTP	Trivial File Transfer Protocol	6, 13, 34, 37
TI	Texas Instruments	7
UART	Universal Asynchronous Receiver-Transmitter	3, 13, 16, 18
OC	Операционная Система	6

1 Общие сведения

Загрузчик IBL (Intermediate Boot Loader) позволяет выполнять загрузку приложений на процессоры модуля TS-PCIE с NOR флеш памяти или по Ethernet с TFTP сервера в локальной сети. Конкретный режим загрузки выбирается при помощи переключателей на плате модуля TS-PCIE (см. приложение B).

В данном документе дано описание основных возможностей загрузчика IBL, описан процесс сборки образов IBL из исходных кодов для записи в EEPROM память модуля TS-PCIE (раздел 2), описана процедура записи загрузчика IBL в EEPROM модуля TS-PCIE (раздел 3.4), описан процесс конфигурирования уже записанного в EEPROM память загрузчика.

Также, в документе описан процесс записи образов приложений в NOR флеш память модуля TS-PCIE для последующей загрузки этих образов загрузчиком IBL.

В данном документе описана работа со средой разработки CCS (Code Composer Studio) версии 5.4.0.00091. Установочный файл среды разработки CCS можно скачать в сети интернет с официального сайта¹, где доступны версии CCS для Windows и Linux систем. На сопроводительном диске к модулю TS-PCIE в папке «Install» имеется установочный файл для среды разработки CCS версии 5.4.0.00091 для Windows системы (файл «ccs_setup_5.4.0.00091.exe»).

В данном руководстве предполагается, что среда разработки CCS установлена в папку «C:\ti» и используется компьютер с установленной 64-х разрядной версией ОС Windows 7.

Внимание



Для установки некоторых программ при помощи установочных дистрибутивов, содержащихся на сопроводительном диске к модулю TS-PCIE, может потребоваться подключение к сети интернет.

¹ http://processors.wiki.ti.com/index.php/Download_CCS

2 Сборка образа IBL из исходных кодов

Для сборки IBL потребуется установленный компилятор для процессоров Texas Instruments серии C6000. Данный компилятор входит в состав среды разработки CCS компании TI.

Результатом сборки IBL из исходных кодов является образ загрузчика готовый к записи в I²C EEPROM модуля TS-PCIE.

Внимание



Перед выполнением сборки загрузчика, описанной в данном разделе, перепишите с сопроводительного диска к модулю TS-PCIE папку «ibl» на жесткий диск компьютера. Далее, предполагается, что все содержимое папки «ibl» с сопроводительного диска переписано в папку «D:/Dev/Modules/TS-PCIE/IBL».

Для сборки загрузчика IBL в Windows системе, кроме компилятора C6000 процессоров, необходима GNU система сборки MinGW. Скачать последнюю версию MinGW можно на официальном сайте¹.

В данном документе описана работа с MinGW версии 20120426. Установочный дистрибутив MinGW версии 20120426 можно найти на сопроводительном диске к модулю TS-PCIE в папке «Install» (файл «mingw-get-inst-20120426.exe»).

2.1 Установка MinGW в Windows системе

В данном разделе описан процесс установки MinGW версии 20120426 с установочного дистрибутива, содержащегося на сопроводительном диске к модулю TS-PCIE (файл «Install/mingw-get-inst-20120426.exe»).

При установке MinGW в окне выбора каталога репозитория (рисунок 2-1) необходимо выбрать пункт «Use pre-packaged repository catalogues» (использовать каталог репозитория с заранее собранными пакетами).

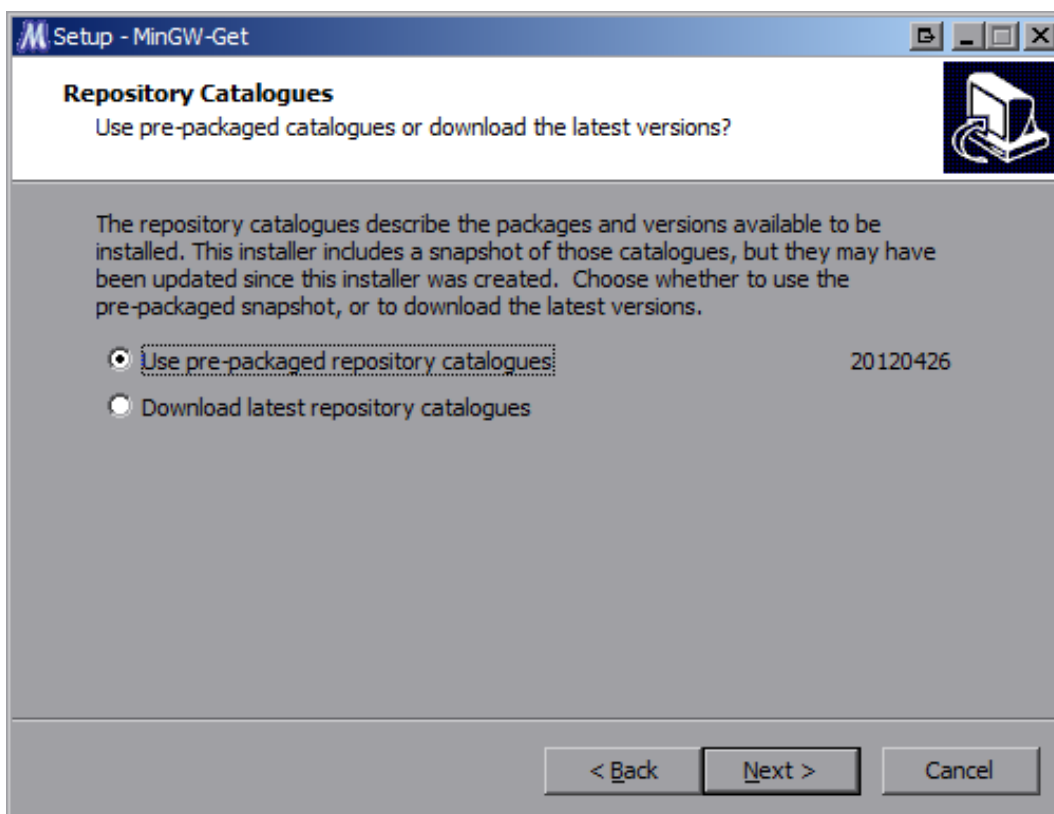


Рисунок 2-1: Установка MinGW. Выбор каталога репозитория

В окне согласия с лицензией (рисунок 2-2), прочитайте текст лицензии, и если вы согласны со всем, что там написано, выберите пункт «I accept the agreement» и нажмите кнопку «Next».

¹ <http://www.mingw.org>

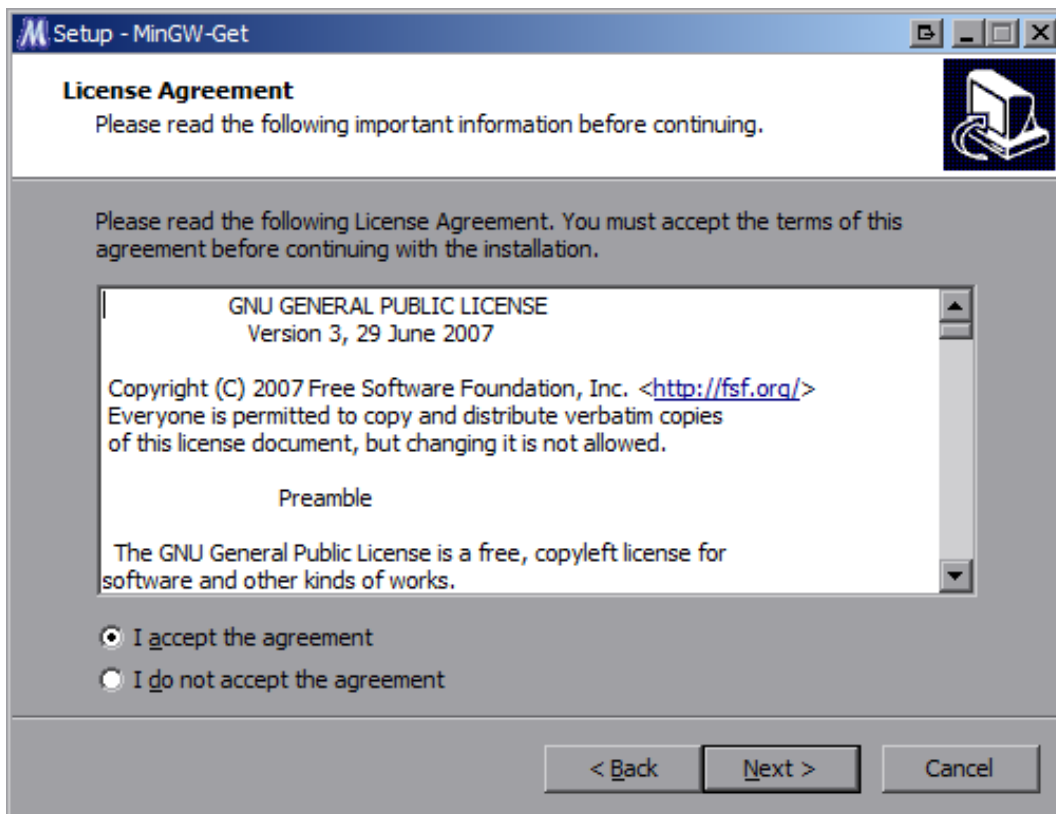


Рисунок 2-2: Установка MinGW. Окно согласия с лицензией

Путь установки MinGW (рисунок 2-3) рекомендуется оставить по умолчанию («C:/MinGW»).

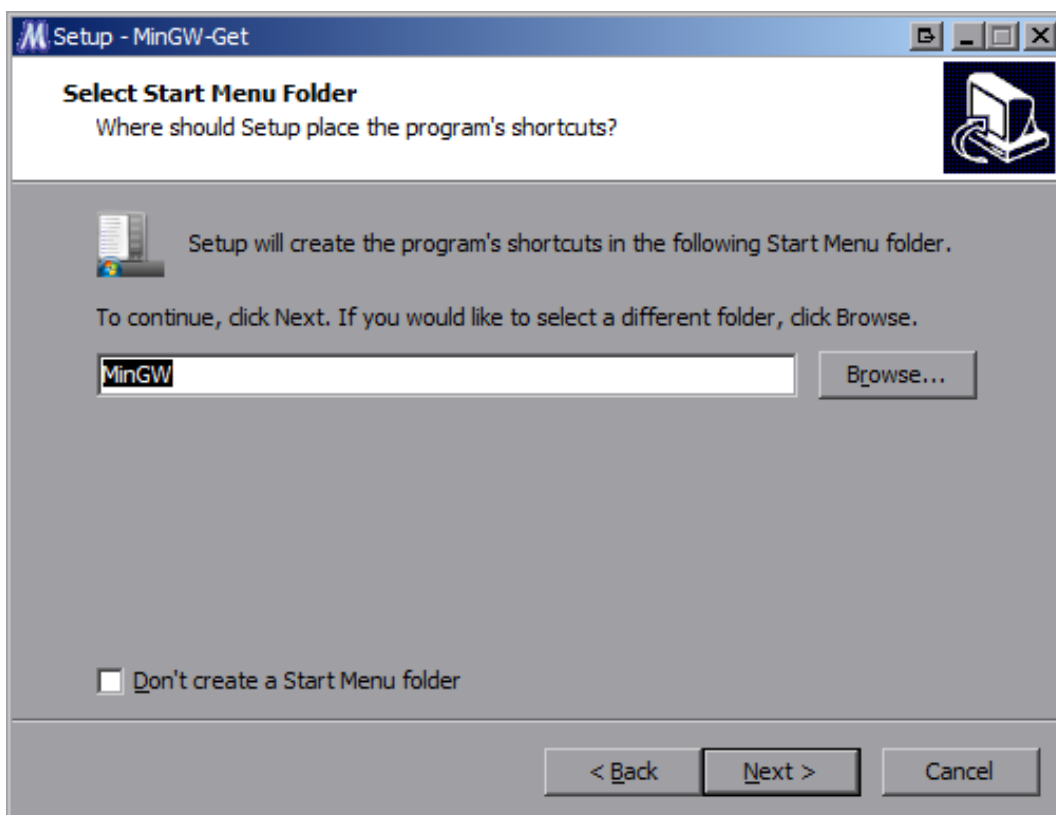


Рисунок 2-3: Установка MinGW. Выбор пути установки

В окне выбора устанавливаемых компонентов (рисунок 2-4) необходимо обязательно отметить следующие компоненты:

- «C Compiler»;
- «MSYS Basic System»;
- «MinGW Developer ToolKit».

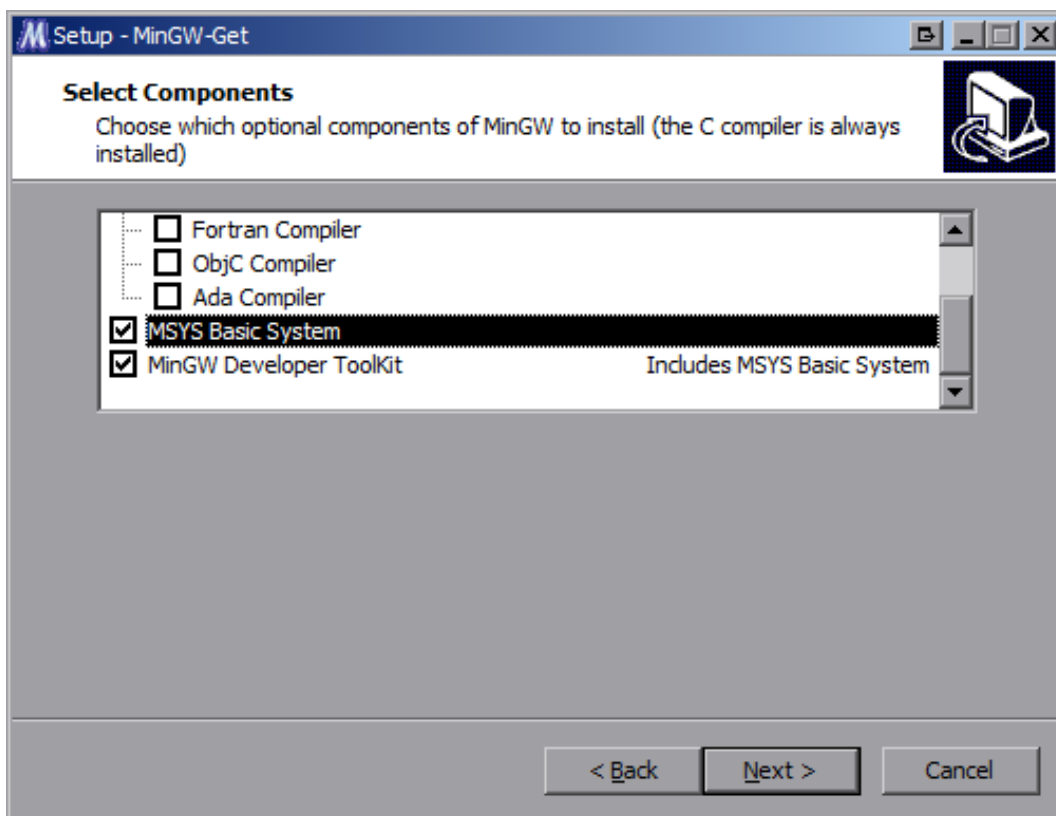


Рисунок 2-4: Установка MinGW. Выбор устанавливаемых компонентов

Остальные компоненты можно отметить на собственное усмотрение. На дальнейший процесс сборки загрузчика IBL их наличие или отсутствие никак не повлияет.

Остальные параметры установки, которые не описаны в данном руководстве, можно оставить в виде, предлагаемом установщиком по умолчанию.

После установки MinGW, через меню «Пуск», запустите «MinGW Shell» (рисунок 2-5). Все последующие действия по сборке IBL будут производиться путем ввода команд в MinGW Shell.

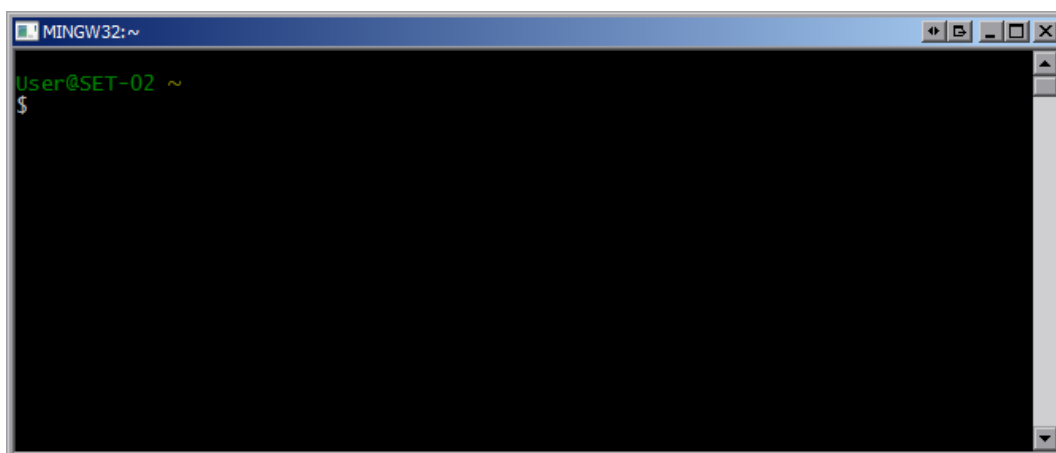
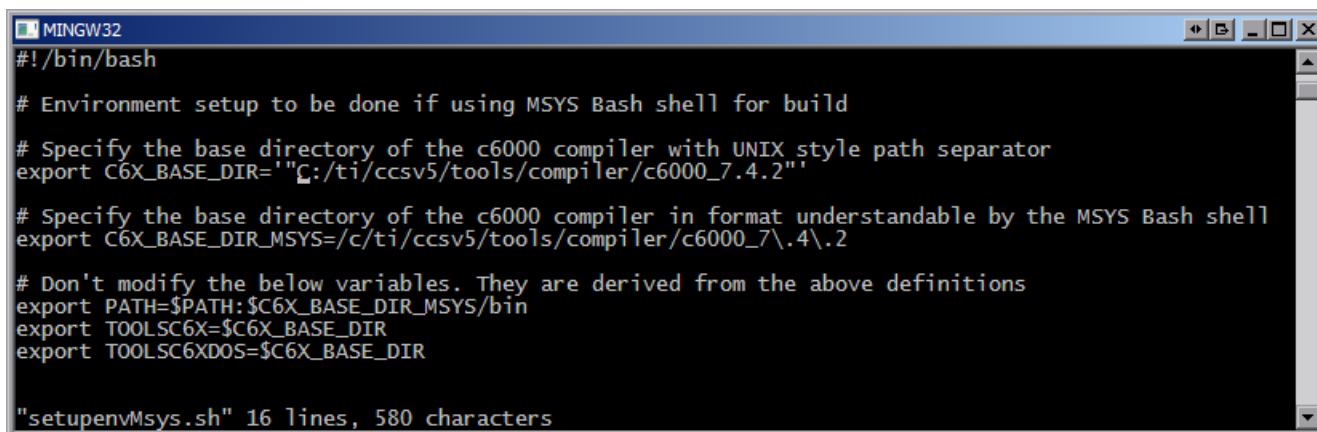


Рисунок 2-5: Приглашение командной строки MinGW Shell

2.2 Конфигурация окружения сборки

В данном разделе предполагается, что исходные коды загрузчика IBL переписаны с сопроводительного диска к модулю TS-PCIE в папку «D:/Dev/Modules/TS-PCIE/IBL», а файлы компилятора для процессоров C6000 серии расположен в папке «C:/ti/ccsv5/tools/compiler/c6000_7.4.2». В том случае, если путь к расположению файлов компилятора для процессоров C6000 отличается, необходимо выполнить соответствующие изменения в скрипте конфигурации окружения сборки «D:/Dev/Modules/TS-PCIE/IBL/src/make/setupenvMsys.sh». В данном файле необходимо правильно указать путь к файлам компилятора для процессоров C6000 серии.

На рисунке 2-6 приведен снимок экрана MinGW Shell с открытым файлом «setupenvMsys.sh» в редакторе vim. На данном рисунке пути установлены в соответствии с путями указанными выше.



```

MINGW32
#!/bin/bash

# Environment setup to be done if using MSYS Bash shell for build

# Specify the base directory of the c6000 compiler with UNIX style path separator
export C6X_BASE_DIR="/c:/ti/ccsv5/tools/compiler/c6000_7.4.2"

# Specify the base directory of the c6000 compiler in format understandable by the MSYS Bash shell
export C6X_BASE_DIR_MSYS="/c:/ti/ccsv5/tools/compiler/c6000_7\4\2"

# Don't modify the below variables. They are derived from the above definitions
export PATH=$PATH:$C6X_BASE_DIR_MSYS/bin
export TOOLSC6X=$C6X_BASE_DIR
export TOOLSC6XD05=$C6X_BASE_DIR

"setupenvMsys.sh" 16 lines, 580 characters

```

Рисунок 2-6: Редактирование файла «setupenvMsys.sh» в редакторе vim

Для редактирования файла «setupenvMsys.sh» в редакторе vim выполните в MinGW Shell команду:

```
vim /d/Dev/Modules/TS-PCIE/IBL/src/make/setupenvMsys.sh
```

В файле «D:/Dev/Modules/TS-PCIE/IBL/src/make/setupenvMsys.sh» путь к файлам компилятора для процессоров C6000 серии необходимо указать в качестве значений двух переменных — «C6X_BASE_DIR» и «C6X_BASE_DIR_MSYS». При этом, в значении переменной «C6X_BASE_DIR_MSYS» такие символы как пробел, точка, открывающая и закрывающая круглые скобки должны обязательно предваряться (экранироваться) символом обратной косой черты «\».

2.3 Сборка загрузчика

Для запуска процесса сборки загрузчика IBL для модуля TS-PCIE, выполните в MinGW Shell последовательно следующие команды:

```
cd /d/Dev/Modules/TS-PCIE/IBL/src/make/
source setupenvMsys.sh
make tspcie
```

После выполнения этих команд, будет запущена сборка загрузчика IBL для модуля TS-PCIE, которая может занять до 30 минут (в зависимости от производительности системы).

В таблице 2-1 перечислены файлы, которые создаются в папке «D:/Dev/Modules/TS-PCIE/IBL/src/make/bin» после выполнения сборки загрузчика IBL для модуля TS-PCIE.

Таблица 2-1: Файлы, создаваемые при сборке загрузчика

Файл	Описание
«i2cConfig.gel»	GEL файл конфигурации IBL для CCS (см. раздел 4)
«i2cparam_0x50_tspcie_le_0x500.out»	Бинарный образ программы конфигурации IBL для процессора TMS320C6670 модуля TS-PCIE (см. раздел 4)
«i2crom_0x50_tspcie_le.bin»	Бинарный образ IBL для процессора TMS320C6670 для загрузки в EEPROM память модуля TS-PCIE (см. раздел 3.4)
«i2crom_0x50_tspcie_le.dat»	Бинарный образ IBL для процессора TMS320C6670 модуля TS-PCIE в формате CCS.

Для записи в EEPROM память модуля TS-PCIE предназначен файл «i2crom_0x50_tspcie_le.bin».

Запись образа загрузчика в EEPROM память производится при помощи специального DSS скрипта для среды разработки CCS, работа с которым рассмотрена в разделе 3.

3 Скрипт для записи EEPROM и NOR флеш памяти модуля TS-PCIE

Для записи образов в EEPROM или NOR флеш память процессора модуля TS-PCIE предназначен специальный DSS скрипт «tspcie_program.js», который расположен в папке «Program» на сопроводительном диске к модулю TS-PCIE.

Перед использованием скрипта «tspcie_program.js» для записи EEPROM или NOR флеш памяти, необходимо скопировать с сопроводительного диска к модулю TS-PCIE папки «Program», «TargetConfigurations» и «GEL» со всем содержимым в папку «D:/Dev/Modules/TS-PCIE»¹.

Скрипт «tspcie_program.js» требует для своей работы установленную систему разработки CCS. Дистрибутив сетевой установки CCS имеется на сопроводительном диске к модулю TS-PCIE (см. раздел 1).

Для запуска скрипта «tspcie_program.js» в папке «Program» имеются вспомогательные скрипты запуска для Windows и Linux систем:

- «tspcie_program.bat»: скрипт для запуска в Windows системе (см. листинг 3-1);
- «tspcie_program.sh»: скрипт для запуска в Linux системе (см. листинг 3-2).

Листинг 3-1: Скрипт «tspcie_program.bat»

```
1 @echo off
2 set PROGRAM_TS_PCIE_TARGET_CONFIG_FILE=../TargetConfigurations/TS-PCIE-LAN560v2-SD.ccxml
3 set DSS_SCRIPT_DIR="C:\ti\ccsv5\ccs_base\scripting\bin"
4 call %DSS_SCRIPT_DIR%\dss.bat tspcie_program.js %*
```

Листинг 3-2: Скрипт «tspcie_program.sh»

```
1 #!/bin/sh
2 export PROGRAM_TS_PCIE_TARGET_CONFIG_FILE=../TargetConfigurations/TS-PCIE-LAN560v2-SD.ccxml
3 export DSS_SCRIPT_DIR=~/.ti/ccsv5/ccs_base/scripting/bin
4 $DSS_SCRIPT_DIR/dss.sh tspcie_program.js $@
```

Как видно из листингов 3-1 и 3-2, в скриптах задаются значения для двух переменных окружения:

- «PROGRAM_TS_PCIE_TARGET_CONFIG_FILE»: путь к файлу описания целевой конфигурации модуля TS-PCIE;
- «DSS_SCRIPT_DIR»: путь к бинарным файлам DSS среды разработки CCS.

Важная информация



Только для Windows: В скрипте «tspcie_program.bat» значение переменной «DSS_SCRIPT_DIR» обязательно должно быть записано в кавычках, а значение переменной «PROGRAM_TS_PCIE_TARGET_CONFIG_FILE», наоборот, не должно содержать кавычек.

После определения значений для переменных окружения, в скрипте происходит вызов скрипта кроссплатформенного DSS скрипта «tspcie_program.js», с передачей ему аргументов командной строки.

Примечание

В данном документе рассматривается работа со скриптом «tspcie_program.bat» для Windows системы. Для выполнения аналогичных действий в Linux системе, достаточно заменить вызовы скрипта «tspcie_program.bat» на вызовы скрипта «tspcie_program.sh».

В разделе 3.1 описывается структура каталога «Program», в котором работает скрипт «tspcie_program.js». Далее, в разделе 3.2, описывается непосредственная работа со скриптом.

¹ Конечная папка может отличаться. В данном документе предполагается, что копирование выполнено именно в эту папку

3.1 Структура каталога «Program»

В листинге 3-3 представлена структура каталога «Program» с сопроводительного диска модуля TS-PCIE, в котором располагаются все необходимые файлы для выполнения записи в EEPROM или NOR флеш память процессора модуля TS-PCIE.

Листинг 3-3: Структура каталога «Program»

```
Program
+- bin
| +- ibl
| | +- i2crom_0x50_tspcie_le.bin
| | +- i2cConfig.ge1
| | +- i2cparam_0x50_tspcie_le_0x500.out
| |
| +- platform_test
| | +- platform_test_tspcie.bin
| |
| +- webservice
| | +- websrv_tspcie.bin
| |
| +- eeprom_0x50_c6670.bin
| +- nor_c6670.bin
|
+- logs
|
+- writers
| +- eepromwriter_c6670.out
| +- norwriter_c6670.out
|
+- tspcie_program.bat
+- tspcie_program.sh
+- tspcie_program.js
```

Кроме файлов скриптов «tspcie_program.bat», «tspcie_program.sh» и «tspcie_program.js», в каталоге «Program» имеет-ся два вложенных каталога:

В каталоге «bin» расположены файлы образов готовые для записи в EEPROM и NOR флеш память процессора модуля TS-PCIE. Пользовательские файлы, подготовленные для записи в EEPROM или NOR флеш память, должны быть помещены в каталог «bin».

Непосредственно в каталоге «bin» расположены два файла, которые используются скриптом «tspcie_program.js» для записи по умолчанию:

- «eeprom_0x50_c6670.bin»: образ для записи в EEPROM память процессора TMS320C6670. По умолчанию, данный файлы представляет собой собранный образ загрузчика IBL, который можно получить путем выполнения шагов, описанных в разделе 2 данного документа.
- «nor_c6670.bin»: образ для записи в NOR флеш память процессора TMS320C6670. По умолчанию, данный файлы представляет собой собранный образ демонстрационного приложения веб-сервера. Подробное описание по сборке и запуску демонстрационного приложения веб-сервера дано в документе [1].

В каталогах «bin/webserver» и «bin/platform_test» расположены образы приложений, готовые для записи в NOR флеш память процессора. Данные образы, также могут применяться для загрузки с TFTP сервера.

В каталоге «bin/platform_test» размещены образы приложения теста платформы (файл «platform_test_tspcie.bin»), которые выполняют следующие тесты: UART, NOR флеш память, EEPROM память, DDR память и тест светодиодов на плате модуля.

Важная информация



При запуске приложения теста платформы следует иметь в виду, что если во время выполнения тестов EEPROM памяти или NOR флеш памяти отключить питание модуля, содержимое тестируемого типа памяти может быть повреждено.

В каталоге «bin/webserver» находится образ собранного приложения демонстрационного веб-сервера (файл «websrv_tspcie.bin»). Подробное описание данного приложение дано в документе [1].

Исходные коды приложения теста платформы находятся проекте «PlatformTest_C6670» рабочего пространства CCS. Папку с рабочим пространством можно найти на сопроводительном диске к модулю TS-PCIE в папке «CCS_Workspace».

В каталоге «writers» расположены файлы программ, для осуществления записи в EEPROM и NOR флеш память процессора. Данные файлы получены путем сборки проектов «EEPROM_Writer» (файл «eepromwriter_c6670.out») и NOR_Writer» (файл «norwriter_c6670.out») рабочего пространства CCS для модуля TS-PCIE. Папку с рабочим пространством со всеми проектами можно найти на сопроводительном диске в папке «CCS_Workspace».

В каталоге «logs» сохраняются логи вывода CIO с процессора в случае запуска скрипта с параметром log (см. таблицу 3-1).

3.2 Работа со скриптом записи NOR флеш и EEPROM памяти

Работа со скриптом записи EEPROM и NOR флеш памяти модуля TS-PCIE осуществляется из командной строки. Выбор того или иного режима работы скрипта осуществляется при помощи параметров командной строки, передаваемых скрипту при запуске.

Если запустить скрипт «tspcie_program.bat» без параметров, будет выдана краткая справка по возможным параметрами командной строки (см. листинг 3-4).

Листинг 3-4: Вывод скрипта «tspcie_program.bat», запущенного без параметров

```

1 D:\Dev\Modules\TS-PCIE\Program>tspcie_program.bat
2
3 Syntax: tspcie_program.js <NOR|EEPROM> <options>
4
5 Mode:
6   NOR      : program NOR flash memory
7   EEPROM   : program I2C EEPROM memory
8
9 Options:
10  image=<image>      : image file for all DSP's
11  bus=<address>      : I2C bus EEPROM address for all DSP's
12                      Address must be a decimal number
13                      Only for EEPROM mode. Default is 80 (0x50)
14  log               : Enable CIO logging
15
16 D:\Dev\Modules\TS-PCIE\Program>
```

Таким образом, первый параметр командной строки задает режим работы скрипта и может принимать одно из значений:

- NOR: режим записи NOR флеш памяти;
- EEPROM: режим записи EEPROM памяти.

Количество остальных параметров командной строки может варьироваться. Доступные параметры приведены в таблице 3-1.

Таблица 3-1: Параметры командной строки скрипта «tspcie_program.js»

Параметр	Описание
image=<image>	Параметр задает имя файла (<image>) образа для записи.
bus=<address>	Параметр позволяет указать адрес (<address>) I ² C шины EEPROM памяти для записи образа. Значение данного параметра учитывается только в режиме EEPROM. По умолчанию, если параметр не задан, используется адрес шины 80 (0x50).
log	Параметр включает запись вывода CIO с процессора модуля TS-PCIE в файлы, которые будут помещены в каталог «logs». Создаваемые файлы лога имеют формат «TS-PCIE-DSP<n>-<time>-cio.txt», где <n> — номер процессора модуля TS-PCIE (всегда 1), <time> — время запуска скрипта «tspcie-program.js». По умолчанию, запись вывода CIO в файлы отключена.

Важная информация



Значение адреса шины для параметров bus обязательно должно указываться в десятичном виде. Соответственно, для адреса шины 0x50 (значение по умолчанию) необходимо указывать значение 80, а для адреса шины 0x51 — значение 81.

3.3 Запись образов в NOR флеш память

Для записи образа в NOR флеш память модуля TS-PCIE выполните шаги процедуры 3-1.

Внимание



Запись в NOR флеш память рекомендуется выполнять при включенном режиме «NOBOOT» на соответствующем процессоре (см. приложение В).

Процедура 3-1. Запись образов в NOR флеш память

1. Выполните подготовку образа(ов).

Внимание



Следует помнить, что объем NOR флеш памяти, установленной на модуле TS-PCIE, составляет 16 Мбайт на каждый процессор. Таким образом, ограничение на размер образов, записываемых в NOR флеш память, составляет 16 Мбайт.

2. Скопируйте подготовленный(е) для записи образ(ы) в каталог «D:/Dev/Modules/TS-PCIE/Program/bin».
3. Перейдите в каталог «D:/Dev/Modules/TS-PCIE/Program».
4. Запустите скрипт «tspcie_program.bat» в режиме NOR с требуемыми параметрами командной строки (см. таблицу 3-1 раздела 3.2).

В процедуре 3-1 представлены общие шаги, необходимые для записи образа в NOR флеш память процессора модуля TS-PCIE.

В качестве примера, в процедуре 3-2 подробно описаны шаги, необходимые для выполнения записи в NOR флеш память процессора модуля TS-PCIE образа приложения демонстрационного веб-сервера [1], начиная от сборки приложения, и заканчивая его загрузкой с NOR флеш памяти при помощи загрузчика IBL.

Важная информация



Записываемые в NOR флеш память файлы образов обязательно должны иметь расширение «.bin».

При сборке приложений в CCS, по умолчанию, файлы имеют расширение «.out». Перед выполнением записи, необходимо переименовать файлы таким образом, чтобы они имели расширение «.bin».

Процедура 3-2. Запись образа демонстрационного приложения веб-сервера в NOR флеш память и его загрузка

1. Выполните сборку приложения веб-сервера для конфигураций сборки «Debug» (процесс сборки приложения веб-сервера подробно описан в документе [1]).

В результате сборки приложений веб-сервера должен быть получен файл «D:/Dev/Modules/TS-PCIE/CCS_Workspace/WebServer_C6670/Debug/websrv_tspcie.out»:

Примечание

Модуль TS-PCIE поставляется с записанным приложением теста платформы, в котором выполняются последовательно тест светодиодов на плате модуля, частичный тест EEPROM памяти, частичный тест NOR флеш памяти и полный тест внешней DDR памяти. Кроме того, записанный в NOR флеш память тест выводит различную информацию о модуле. Пример вывода в UART теста платформы приведен в листинге 3-6.

2. Скопируйте файл, полученный при выполнении шага 1 в папку «D:/Dev/Modules/TS-PCIE/Program/bin». Для этого выполните в терминале последовательно следующие команды (при копировании, также выполняется изменение расширения «.out» на «.bin», что необходимо для правильной записи):

```
d:
cd D:\Dev\Modules\TS-PCIE
copy CCS_Workspace\WebServer_C6670\Debug\websrv_tspcie.out Program\bin\websrv_tspcie.bin
```


3. Установите для процессора модуля TS-PCIE режим «NOBOOT» в соответствии с данными таблицы B-1 приложения B.
4. Подключите отладчик к модулю TS-PCIE и к компьютеру (предполагается использование отладчика Spectrum Digital XDS560v2 STM LAN).
5. Включите модуль TS-PCIE.
6. Перейдите в каталог «D:\Dev\Modules\TS-PCIE\Program». Для этого необходимо выполнить в терминале команды:

```
d:
cd D:\Dev\Modules\TS-PCIE\Program
```

7. Запустите скрипт «tspcie_program.bat» выполнив в терминале команду:

```
tspcie_program.bat NOR image=websrv_tspcie.bin
```

Будет запущен процесс записи образов в NOR флеш память процессора модуля TS-PCIE. При этом, вывод в терминал должен быть аналогичен тому, что приведен в листинге A-1 приложения A. Весь процесс записи должен занимать не более 5 минут.

После вывода в терминал сообщения «Executing writers on DSP(s)...» на процессоре модуля TS-PCIE начинается процесс записи данных образов в NOR флеш память. При этом, светодиоды «VH11» и «VH12» на плате модуля TS-PCIE будут иметь определенное состояние в зависимости от хода процесса записи (см. рисунок 3-1).

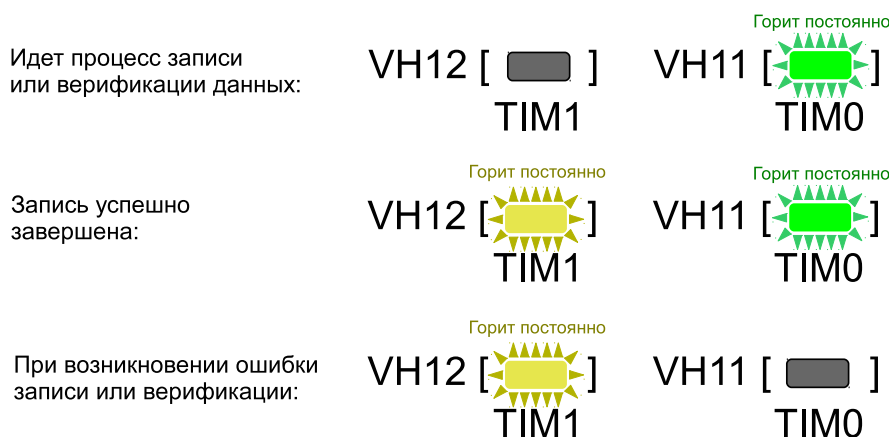


Рисунок 3-1: Состояния светодиодов на плате модуля TS-PCIE во время записи

При нормальном процессе записи и верификации записанных данных, должен гореть только зеленый светодиод «VH11». Желтый светодиод «VH12» гореть не должен.

Если во время записи или верификации происходит ошибка, то загорается желтый светодиод «VH12», зеленый светодиод «VH11» гаснет (см. рисунок 3-1). При этом, работа программы записи прекращается.

В случае успешного завершения записи и верификации записанных данных, загораются оба светодиода «VH11» и «VH12».

8. Выключите модуль TS-PCIE.
9. Если в EEPROM память процессора модуля TS-PCIE еще не записан образ загрузчика IBL, выполните шаги процедуры 3-4.

Примечание

Модуль TS-PCIE поставляется с уже записанным образом загрузчика IBL в EEPROM память (адрес I²C шины 0x50). Поэтому, если в EEPROM память (адрес I²C шины 0x50) не производилась запись каких либо других образов, выполнять запись загрузчика не требуется.

10. Установите для процессора модуля TS-PCIE режим «NOR» в соответствии с данными таблицы B-1 приложения B.
11. Включите модуль TS-PCIE.

После выполнения всех шагов процедуры 3-2, вывод в UART с процессора модуля TS-PCIE при его включении должен выглядеть подобно приведенному в листинге 3-5.

Листинг 3-5: Вывод в UART загрузки демонстрационного приложения веб-сервера с NOR флеш памяти

```

1  IBL Init
2  IBL build: version 1.0.0.16-8, date: Jan 18 2016, time: 21:41:18
3  PCIE: LINK_INIT
4  PCIE: LINK_TRAIN
5  PCIE: LINK_UP
6
7  PCIE: Initialization complete
8      Last state:          0x00000006
9      Counter:            0x03938701
10     Down PMRST reset:    0x00000000
11     Down PMRST turnoff: 0x00000000
12     Down LTSSM:         0x00000000
13     Init:                0x00000001
14     Trained:             0x00000001
15
16
17  IBL build: version 1.0.0.16-8, date: Jan 18 2016, time: 21:41:18
18  IBL start: 0x0080CEC0
19  Device is TS-PCIE
20  GbE switch subsystem initialized
21  Device MAC is 40:5f:c2:c1:e6:80
22  Result code 0
23  Booting from NOR
24  Booting from NOR flash from address
25     = 0x8157c8e0
26  platform_init Done
27  Start BIOS 6
28  QMSS successfully initialized
29  CPPI successfully initialized
30  PA successfully initialized
31  HUA version 2.00.00.04
32  Setting hostname to tidemo-345678
33  MAC Address: 40-5F-C2-C1-E6-80
34  Configuring DHCP client
35  PASS successfully initialized
36  Ethernet subsystem successfully initialized
37  Ethernet eventId : 48 and vectId (Interrupt) : 7
38  Registration of the EMAC Successful, waiting for link up ..
39  Service Status: DHCP   : Enabled   :           : 000
40  Service Status: THHTTP : Enabled   :           : 000
41  Service Status: DHCP   : Enabled   : Running  : 000
42  Network Added: If-1:192.168.2.94
43  Service Status: DHCP   : Enabled   : Running  : 017

```

В листинге 3-6 приведен пример вывода в UART запуска приложения теста платформы модуля TS-PCIE (файл «Program/bin/platform_test_tspcie.bin»).

Листинг 3-6: Вывод в UART при запуске теста платформы

```

1  Platform test for module
2
3  |_____/____| |____\ /____|____|____|
4  | | | (_____|____) |____| | | | |
5  | | \____\____| ____/ |____| | | | |
6  | | ____|____| |____| |____| | | | |
7  | | |____/ | | \____|____|____|
8
9  p_info->version                = 2.00.00.15
10 p_info->board_name              = TS-PCIE
11 p_info->board_id                = default_devid
12 p_info->serial_nbr              = 012345678
13 p_info->slot_n                  = 3
14 p_info->dsp_n                   = DSP1
15 p_info->cpu.core_count          = 4
16 p_info->cpu.name                 = TMS320C6670
17 p_info->cpu.id                  = 21
18 p_info->cpu.revision_id         = 0
19 p_info->cpu.endian               = 1
20 p_info->frequency                = 983 MHz
21 p_info->led[PLATFORM_USER_LED_CLASS].count = 2
22 p_info->led[PLATFORM_SYSTEM_LED_CLASS].count = 0
23 p_info->emac.port_count          = 1

```

```

24  -> EMAC port 0
25      MAC address = 40:5f:c2:c1:e6:80
26
27  NOR device info:
28  p_device->device_id           = 0xbb18
29  p_device->manufacturer_id     = 0x20
30  p_device->width                = 8
31  p_device->block_count         = 256
32  p_device->page_count          = 256
33  p_device->page_size           = 256
34  p_device->spare_size          = 0
35  p_device->handle               = 0xbb18
36  p_device->flags                = 0
37  p_device->bboffset            = 0
38
39  EEPROM device (I2C address 0x50) info:
40  p_device->device_id           = 0x50
41  p_device->manufacturer_id     = 0x1
42  p_device->width                = 8
43  p_device->block_count         = 1
44  p_device->page_count          = 1
45  p_device->page_size           = 65536
46  p_device->spare_size          = 0
47  p_device->handle               = 0x50
48  p_device->flags                = 0
49  p_device->bboffset            = 0
50
51  EEPROM device (I2C address 0x51) info:
52  p_device->device_id           = 0x51
53  p_device->manufacturer_id     = 0x1
54  p_device->width                = 8
55  p_device->block_count         = 1
56  p_device->page_count          = 1
57  p_device->page_size           = 65536
58  p_device->spare_size          = 0
59  p_device->handle               = 0x51
60  p_device->flags                = 0
61  p_device->bboffset            = 0
62  Current core id is 0
63
64  -----
65  LED test
66  -----
67  Testing USER class leds (2 leds)...
68  - LED 0 ON
69  - LED 0 OFF
70  - LED 0 ON
71  - LED 0 OFF
72  - LED 1 ON
73  - LED 1 OFF
74  - LED 1 ON
75  - LED 1 OFF
76  LED test complete
77
78  -----
79  EEPROM test
80  -----
81  test_eeprom: Partial EEPROM test mode
82  test_eeprom: 0x50: Saving EEPROM contents...
83  test_eeprom: 0x50: Writing 1024 bytes...
84  test_eeprom: 0x50: Read and check data...
85  test_eeprom: 0x50: Restoring EEPROM contents...
86  test_eeprom: 0x50: Test passed
87  test_eeprom: 0x51: Saving EEPROM contents...
88  test_eeprom: 0x51: Writing 1024 bytes...
89  test_eeprom: 0x51: Read and check data...
90  test_eeprom: 0x51: Restoring EEPROM contents...
91  test_eeprom: 0x51: Test passed
92  EEPROM test complete
93
94  -----
95  NOR test
96  -----
97  test_nor: Partial NOR test mode
98  test_nor: NOR flash sector size is 65536 bytes
99  test_nor: Test block size is 65536 bytes
100 test_nor: Saving original data from NOR...
101 test_nor: Writing pattern data to NOR...

```

```
102 test_nor: Reading pattern data from NOR...
103 test_nor: Comparing data...
104 test_nor: Data is equal. Test passed.
105 test_nor: Write back original data to NOR...
106 test_nor: Test passed
107 NOR test complete
108
109 -----
110 External memory test
111 -----
112 platform_external_memory_test(start = 0x80000000, end = 0xffffffff) called
113 External memory test passed
114 External memory test complete
```

3.4 Запись образов в EEPROM память

Для записи образа в EEPROM память модуля TS-PCIE выполните шаги процедуры 3-3.

Внимание



Запись в EEPROM память рекомендуется выполнять при включенном режиме «NOBOOT» (см. приложение В).

Процедура 3-3. Запись образов в EEPROM память

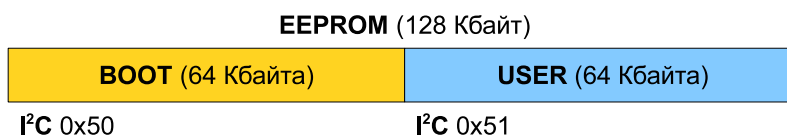
1. Выполните подготовку образа(ов).

Внимание



Следует помнить, что общий объем EEPROM памяти, установленной на модуле TS-PCIE, составляет 128 Кбайт. EEPROM память, установленная на модуле TS-PCIE, разделена на две части (каждая объемом по 64 Кбайта): адресу 0x50 шины I²C соответствует первая часть, а адресу 0x51 шины I²C — вторая часть (см. рисунок 3-2). При этом, загрузка модуля TS-PCIE возможна только из первой части EEPROM с адресом шины I²C 0x50. Таким образом, ограничение на размер образов, записываемых в EEPROM память, составляет 64 Кбайт.

2. Скопируйте подготовленный(е) для записи образ(ы) в каталог «D:/Dev/Modules/TS-PCIE/Program/bin».
3. Перейдите в каталог «D:/Dev/Modules/TS-PCIE/Program».
4. Запустите скрипт «tspcie_program.bat» в режиме EEPROM с требуемыми параметрами командной строки (см. таблицу 3-1 раздела 3.2).



BOOT — область загрузчика IBL;

USER — область пользовательских данных

Рисунок 3-2: Организация EEPROM памяти на модуле TS-PCIE

В процедуре 3-3 представлены общие шаги, необходимые для записи образа в EEPROM память процессора модуля TS-PCIE.

Важная информация



Записываемые в EEPROM память файлы образов обязательно должны иметь расширение «.bin».

В качестве примера, в процедуре 3-4 подробно описаны шаги, необходимые для выполнения записи в EEPROM память процессора модуля TS-PCIE образа загрузчика IBL.

Процедура 3-4. Запись образа загрузчика IBL в EEPROM память и его загрузка

1. Выполните сборку загрузчика IBL (процесс сборки IBL подробно описан в разделе 2).

В результате сборки загрузчика IBL должен быть получен файл «D:/Dev/Modules/TS-PCIE/ibl/src/make/bin/i2crom_0x50_tspcie_le.bin».

Примечание

Модуль TS-PCIE поставляется с уже записанным образом загрузчика IBL в EEPROM память процессора. Образ, записанный в EEPROM память при изготовлении модуля TS-PCIE содержится в файле «i2crom_0x50_tspcie_le.bin». Данный файл расположен в каталоге «D:/Dev/Modules/TS-PCIE/Program/bin/ibl».

2. Скопируйте файл, полученный при выполнении шага 1 в папку «D:/Dev/Modules/TS-PCIE/Program/bin». Для этого выполните в терминале последовательно следующие команды:

```
d:  
cd D:\Dev\Modules\TS-PCIE  
copy ibl\src\make\bin\i2crom_0x50_tspcie_le.bin Program\bin
```

3. Установите для процессора модуля TS-PCIE режим «NOBOOT» в соответствии с данными таблицы B-1 приложения B.
4. Подключите отладчик к модулю TS-PCIE и к компьютеру (предполагается использование отладчика Spectrum Digital XDS560v2 STM LAN).
5. Включите модуль TS-PCIE.
6. Перейдите в каталог «D:/Dev/Modules/TS-PCIE/Program». Для этого необходимо выполнить в терминале команды:

```
d:  
cd D:\Dev\Modules\TS-PCIE\Program
```

7. Запустите скрипт «tspcie_program.bat» выполнив в терминале команду:

```
tspcie_program.bat EEPROM image=i2crom_0x50_tspcie_le.bin
```

Будет запущен процесс записи образа в EEPROM память процессора модуля TS-PCIE. При этом, вывод в терминал должен быть аналогичен тому, что приведен в листинге A-2 приложения A. Весь процесс записи должен занимать не более 5 минут.

Состояния светодиодов «VH11» и «VH12» на плате модуля TS-PCIE во время записи EEPROM памяти соответствуют состояниям светодиодов во время записи NOR флеш памяти, которые описаны в шаге 7 процедуры 3-2.

8. Выключите модуль TS-PCIE.
9. Установите для процессора модуля TS-PCIE требуемый режим загрузки («NOR» или «TFTP») в соответствии с данными таблицы B-1 приложения B.
10. Включите модуль TS-PCIE.

4 Конфигурация IBL

Записанный в EEPROM память загрузчик IBL хранит блок с конфигурационными параметрами в этой же EEPROM памяти с определенным смещением. Конфигурация IBL заключается в изменении данных блока с конфигурационными параметрами в EEPROM памяти.

Для облегчения процесса изменения данных в блоке конфигурационных данных загрузчика, при сборке IBL (см. раздел 2), создается образ специальной программы «i2cparam_0x50_tspcie_le_0x500.out» (см. таблицу 2-1), которая предназначена для загрузки на процессоре модуля TS-PCIE для конфигурации загрузчика.

Примечание

Значение «0x500» в имени файлов «i2cparam_0x50_tspcie_le_0x500.out» определяет величину смещения расположения блока конфигурационных параметров загрузчика IBL относительно начала EEPROM памяти.

Для запуска программы конфигурации загрузчика IBL необходимо, в первую очередь, запустить целевую конфигурацию модуля TS-PCIE, как показано в разделе 5.

После запуска целевой конфигурации, как показано в разделе 5, выполните подключение к ядру «DSP1_C6670» процессора модуля. Для этого, нажмите правой кнопкой мыши на названии ядра «DSP1_C6670_0» и выберите пункт меню «Connect Target» (см. рисунок 4-1).

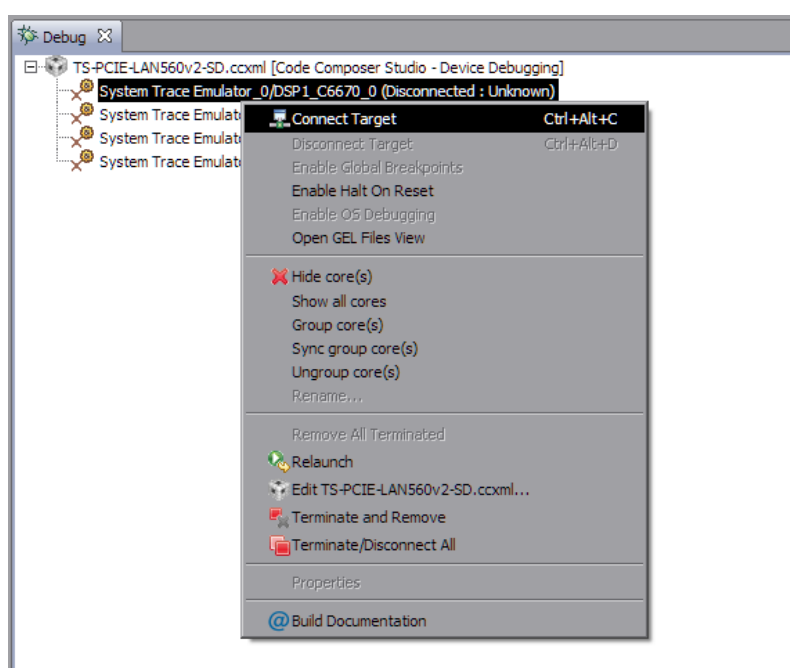


Рисунок 4-1: Подключение к ядру «DSP1_C6670»

Выберите ядро «DSP1_C6670_0» в окне «Debug» (щелкните левой кнопкой мыши по названию ядра). Выберите пункт главного меню «Run > Load > Load Program...». В открывшемся окне (рисунок 4-2) нажмите на кнопку «Browse».

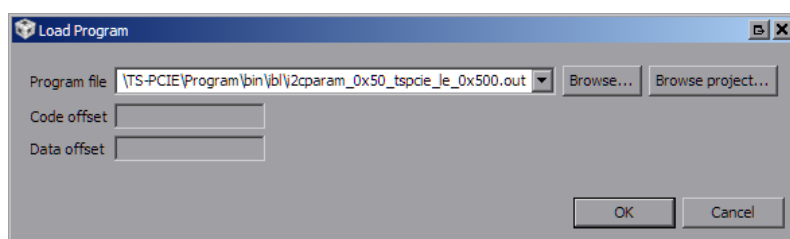


Рисунок 4-2: Окно загрузки кода на ядро процессора

В открывшемся окне выбора файлов необходимо выбрать файл «i2cparam_0x50_tspcie_le_0x500.out» (см. таблицу 2-1) и нажать на кнопку «OK». Данный файл также можно найти в папке «D:/Dev/Modules/TS-PCIE/Program/bin/ibl».

После загрузки кода на ядро процессора, окно «Debug» должно выглядеть, как показано на рисунке 4-3.

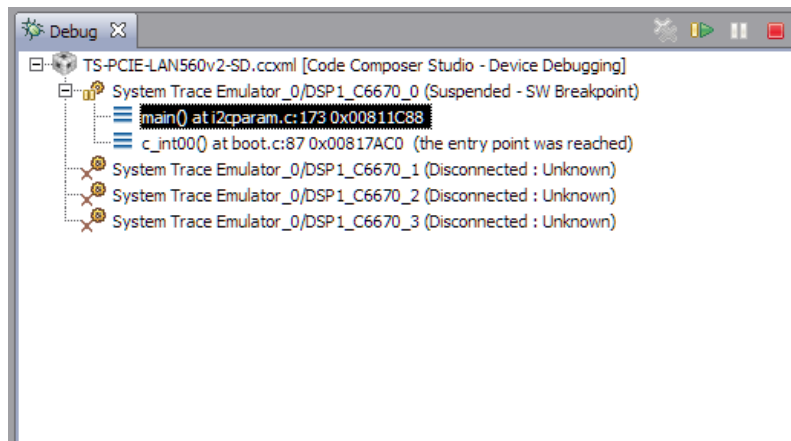




Рисунок 4-3: Внешний вид окна «Debug» после загрузки кода на ядра процессоров

Выделите в окне «Debug» ядро «DSP1_C6670_0» щелкнув по нему левой кнопкой мыши. и запустите выполнение кода, нажав на кнопку  («Resume»). Кнопка  («Resume») находится в верхней части окна «Debug» (см. рисунок 4-3).

После запуска приложения конфигурации загрузчика, в окно «Console» будет выведено сообщение «Run the GEL for the device to be configured, press return to programm the I2C» (см. рисунок 4-4).

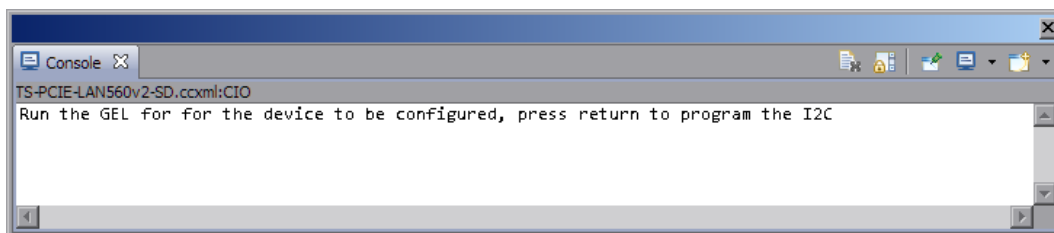


Рисунок 4-4: Внешний вид окна «Console» после запуска кода на процессоре

Данное сообщение означает, что программа ожидает пока будет произведена конфигурация параметров загрузчика IBL и предлагает нажать на клавишу «Enter» для записи выполненной конфигурации в EEPROM память.

Конфигурация параметров загрузчика осуществляется при помощи специального GEL файла «i2cConfig.gel», который можно найти в папке «D:/Dev/Modules/TS-PCIE/Program/bin/ibl».

Для вызова окна управления GEL файлами, выберите пункт главного меню «Tools > GEL Files», как показано на рисунке 4-5.

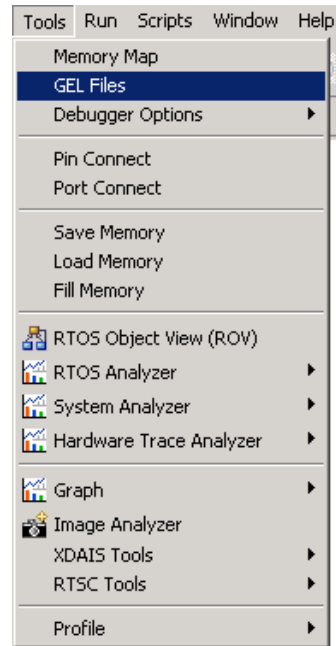


Рисунок 4-5: Пункт главного меню для вызова окна управления GEL файлами

В открывшемся окне «GEL Files» (рисунок 4-6) нажмите правой кнопкой мыши в области со списком загруженных GEL файлов (изначально там должен быть только один файл «tspcie.gel») и выберите пункт меню «Load GEL...».

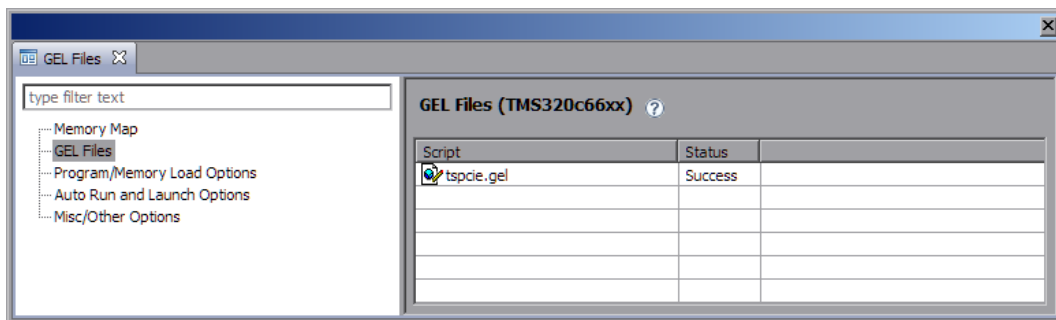


Рисунок 4-6: Окно управления GEL файлами

В открывшемся окне необходимо выбрать файл «D:/Dev/Modules/TS-PCIE/Program/bin/ibl/i2cConfig.gel». После чего, в списке загруженных файлов окна «GEL Files» должен появиться файл «i2cConfig.gel», как показано на рисунке 4-7.

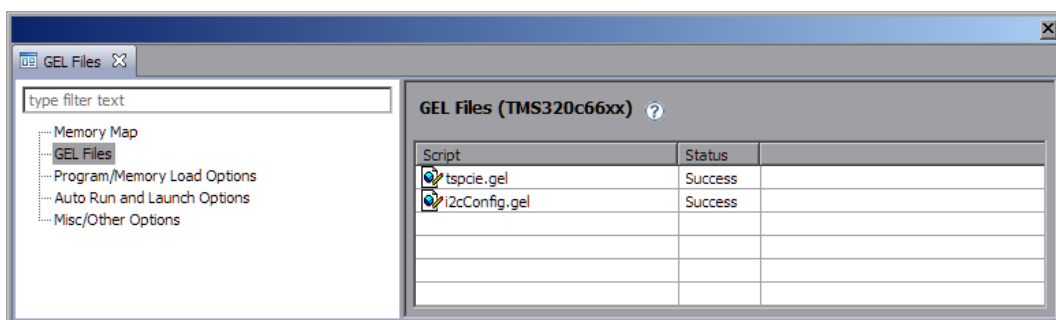


Рисунок 4-7: Окно управления GEL файлами с загруженным файлом «i2cConfig.gel»

После выполнения данных действий, появится пункт меню «Scripts > SET TS-PCIE IBL > setConfig_tspcie_main» при выборе которого будет применена конфигурация загрузчика (см. рисунок 4-8).

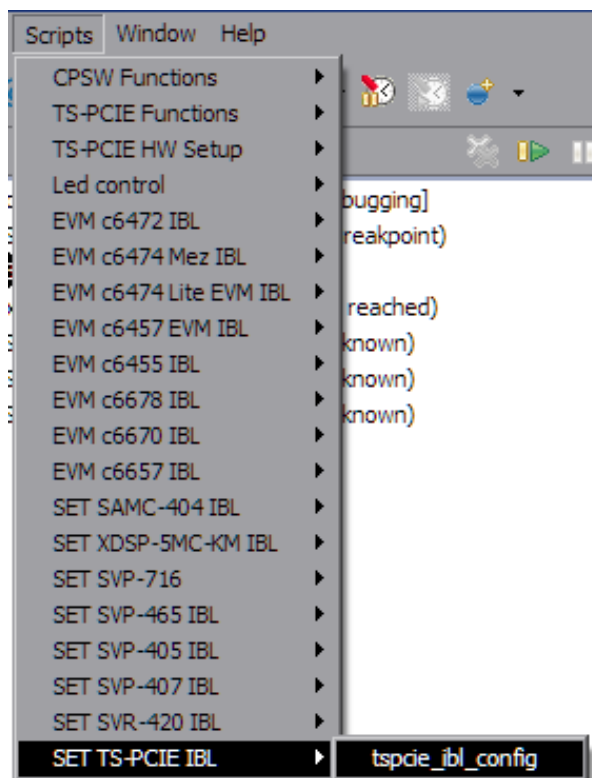


Рисунок 4-8: Пункт главного меню «Scripts > SET TS-PCIE IBL»

Далее, необходимо щелкнуть мышкой в окне «Console» и нажать клавишу «Enter» для выполнения записи конфигурационных параметров загрузчика в EEPROM память. После чего, в окне «Console» должно появиться сообщение «I2c table write complete» (см. рисунок 4-9).

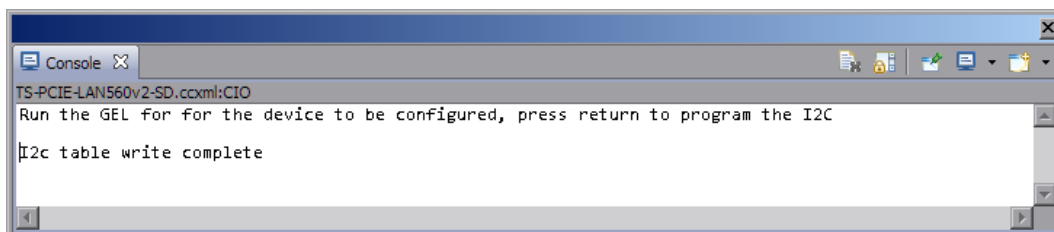


Рисунок 4-9: Внешний вид окна «Console» после выполнения записи конфигурации в EEPROM память

Конфигурационные параметры загрузчика IBL задаются в загружаемом файле «i2cConfig.gel» путем его редактирования. Редактирование данного файла должно производиться до его загрузки в окно «GEL Files». В том случае, если файл редактируется после того, как он загружен в окне «GEL Files», после редактирования, необходимо щелкнуть правой кнопкой мыши по названию файла в окне «GEL Files» и выбрать пункт меню «Reload» (см. рисунок 4-6).

В листинге 4-1 приведен фрагмент файла «i2cConfig.gel» с конфигурационными параметрами загрузчика для модуля TS-PCIE.

Листинг 4-1: Фрагмент GEL файла «i2cConfig.gel» с конфигурационными параметрами загрузчика IBL для модуля TS-PCIE

```

1691 menuitem "SET TS-PCIE IBL";
1692
1693 hotmenu tspcie_ibl_config()
1694 {
1695     ibl.iblMagic = ibl_MAGIC_VALUE;
1696     ibl.iblEvmType = ibl_TS_PCIE;
1697
1698     /* Main PLL: 122.88 MHz reference, 983 MHz output */
1699     ibl.pllConfig[ibl_MAIN_PLL].doEnable = 1;
1700     ibl.pllConfig[ibl_MAIN_PLL].prediv = 1;
1701     ibl.pllConfig[ibl_MAIN_PLL].mult = 16;
1702     ibl.pllConfig[ibl_MAIN_PLL].postdiv = 2;
1703     ibl.pllConfig[ibl_MAIN_PLL].pllOutFreqMhz = 983;
1704
1705     /* DDR PLL */

```

```

1706 ibl.pllConfig[ibl_DDR_PLL].doEnable = 1;
1707 ibl.pllConfig[ibl_DDR_PLL].prediv = 1;
1708 ibl.pllConfig[ibl_DDR_PLL].mult = 20;
1709 ibl.pllConfig[ibl_DDR_PLL].postdiv = 2;
1710 ibl.pllConfig[ibl_DDR_PLL].pllOutFreqMhz = 1333;
1711
1712 /* Net PLL: 122.88 MHz reference, 1044 MHz output (followed by a built in divide by 3 to give 348 MHz to PA) */
1713 ibl.pllConfig[ibl_NET_PLL].doEnable = 1;
1714 ibl.pllConfig[ibl_NET_PLL].prediv = 1;
1715 ibl.pllConfig[ibl_NET_PLL].mult = 17;
1716 ibl.pllConfig[ibl_NET_PLL].postdiv = 2;
1717 ibl.pllConfig[ibl_NET_PLL].pllOutFreqMhz = 1044;
1718
1719
1720 ibl.ddrConfig.configDdr = 1;
1721 ibl.ddrConfig.uEmif.emif4p0.registerMask = ibl_EMIF4_ENABLE_sdRamConfig | ibl_EMIF4_ENABLE_sdRamRefreshCtl |
  ← ibl_EMIF4_ENABLE_sdRamTiming1 | ibl_EMIF4_ENABLE_sdRamTiming2 | ibl_EMIF4_ENABLE_sdRamTiming3 |
  ← ibl_EMIF4_ENABLE_ddrPhyCtl1;
1722
1723 ibl.ddrConfig.uEmif.emif4p0.sdRamConfig = 0x63066A32; // = 0x63C452B2;
1724 ibl.ddrConfig.uEmif.emif4p0.sdRamConfig2 = 0;
1725 ibl.ddrConfig.uEmif.emif4p0.sdRamRefreshCtl = 0x0000514C; // 0x000030D4;
1726 ibl.ddrConfig.uEmif.emif4p0.sdRamTiming1 = 0x1113783C; // 0x0AAA51B;
1727 ibl.ddrConfig.uEmif.emif4p0.sdRamTiming2 = 0x304D7FE3; // 0x2A2F7FDA;
1728 ibl.ddrConfig.uEmif.emif4p0.sdRamTiming3 = 0x559F849F; // 0x057F82B8;
1729 ibl.ddrConfig.uEmif.emif4p0.lpDdrNvmTiming = 0;
1730 ibl.ddrConfig.uEmif.emif4p0.powerManageCtl = 0;
1731 ibl.ddrConfig.uEmif.emif4p0.iODFTTestLogic = 0;
1732 ibl.ddrConfig.uEmif.emif4p0.performCountCfg = 0;
1733 ibl.ddrConfig.uEmif.emif4p0.performCountMstRegSel = 0;
1734 ibl.ddrConfig.uEmif.emif4p0.readIdleCtl = 0;
1735 ibl.ddrConfig.uEmif.emif4p0.sysVbusmIntEnSet = 0;
1736 ibl.ddrConfig.uEmif.emif4p0.sdRamOutImpdedCalCfg = 0;
1737 ibl.ddrConfig.uEmif.emif4p0.tempAlterCfg = 0;
1738 ibl.ddrConfig.uEmif.emif4p0.ddrPhyCtl1 = 0x0010010d;
1739 ibl.ddrConfig.uEmif.emif4p0.ddrPhyCtl2 = 0;
1740 ibl.ddrConfig.uEmif.emif4p0.priClassSvceMap = 0;
1741 ibl.ddrConfig.uEmif.emif4p0.mstId2ClsSvce1Map = 0;
1742 ibl.ddrConfig.uEmif.emif4p0.mstId2ClsSvce2Map = 0;
1743 ibl.ddrConfig.uEmif.emif4p0.eccCtl = 0;
1744 ibl.ddrConfig.uEmif.emif4p0.eccRange1 = 0;
1745 ibl.ddrConfig.uEmif.emif4p0.eccRange2 = 0;
1746 ibl.ddrConfig.uEmif.emif4p0.rdWrtExcThresh = 0;
1747
1748 ibl.sgmiiconfig[0].configure = 1;
1749 ibl.sgmiiconfig[0].adviseAbility = 0x9801;
1750 ibl.sgmiiconfig[0].control = 1;
1751 ibl.sgmiiconfig[0].txConfig = 0x108a1;
1752 ibl.sgmiiconfig[0].rxConfig = 0x700621;
1753 ibl.sgmiiconfig[0].auxConfig = 0x51;
1754
1755 ibl.sgmiiconfig[1].configure = 1;
1756 ibl.sgmiiconfig[1].adviseAbility = 0x9801; // 1
1757 ibl.sgmiiconfig[1].control = 0x20; // 1;
1758 ibl.sgmiiconfig[1].txConfig = 0x108a1;
1759 ibl.sgmiiconfig[1].rxConfig = 0x700621;
1760 ibl.sgmiiconfig[1].auxConfig = 0x51;
1761
1762 ibl.mdioConfig.nMdioOps = 0;
1763
1764 ibl.spiConfig.addrWidth = 24;
1765 ibl.spiConfig.nPins = 5;
1766 ibl.spiConfig.mode = 1;
1767 ibl.spiConfig.csel = 2;
1768 ibl.spiConfig.c2tdelay = 1;
1769 ibl.spiConfig.busFreqMHZ = 20;
1770
1771 ibl.emifConfig[0].csSpace = 2;
1772 ibl.emifConfig[0].busWidth = 8;
1773 ibl.emifConfig[0].waitEnable = 0;
1774
1775 ibl.emifConfig[1].csSpace = 0;
1776 ibl.emifConfig[1].busWidth = 0;
1777 ibl.emifConfig[1].waitEnable = 0;
1778
1779 ibl.bootModes[0].bootMode = ibl_BOOT_MODE_NOR;
1780 ibl.bootModes[0].priority = ibl_HIGHEST_PRIORITY;
1781 ibl.bootModes[0].port = 0;
1782
1783 ibl.bootModes[0].u.norBoot.bootFormat = ibl_BOOT_FORMAT_ELF;
1784 ibl.bootModes[0].u.norBoot.bootAddress[0][0] = 0; /* Image 0 NOR offset byte address in LE mode */
1785 ibl.bootModes[0].u.norBoot.bootAddress[0][1] = 0xA00000; /* Image 1 NOR offset byte address in LE mode */
1786 ibl.bootModes[0].u.norBoot.bootAddress[1][0] = 0; /* Image 0 NOR offset byte address in BE mode */
1787 ibl.bootModes[0].u.norBoot.bootAddress[1][1] = 0xA00000; /* Image 1 NOR offset byte address in BE mode */
1788 ibl.bootModes[0].u.norBoot.interface = ibl_PMEM_IF_SPI;
1789 ibl.bootModes[0].u.norBoot.blob[0][0].startAddress = 0x80000000; /* Image 0 Load start address in LE mode */
1790 ibl.bootModes[0].u.norBoot.blob[0][0].sizeBytes = 0xA00000; /* Image 0 size (10 MB) in LE mode */

```

```

1791  ibl.bootModes[0].u.norBoot.blob[0][0].branchAddress = 0x80000000;    /* Image 0 branch address after Loading in LE mode
    < */
1792  ibl.bootModes[0].u.norBoot.blob[0][1].startAddress = 0x80000000;    /* Image 1 Load start address in LE mode */
1793  ibl.bootModes[0].u.norBoot.blob[0][1].sizeBytes = 0xA000000;        /* Image 1 size (10 MB) in LE mode */
1794  ibl.bootModes[0].u.norBoot.blob[0][1].branchAddress = 0x80000000;    /* Image 1 branch address after Loading in LE mode
    < */
1795  ibl.bootModes[0].u.norBoot.blob[1][0].startAddress = 0x80000000;    /* Image 0 Load start address in BE mode */
1796  ibl.bootModes[0].u.norBoot.blob[1][0].sizeBytes = 0xA000000;        /* Image 0 size (10 MB) in BE mode */
1797  ibl.bootModes[0].u.norBoot.blob[1][0].branchAddress = 0x80000000;    /* Image 0 branch address after Loading in BE mode
    < */
1798  ibl.bootModes[0].u.norBoot.blob[1][1].startAddress = 0x80000000;    /* Image 1 Load start address in BE mode */
1799  ibl.bootModes[0].u.norBoot.blob[1][1].sizeBytes = 0xA000000;        /* Image 1 size (10 MB) in BE mode */
1800  ibl.bootModes[0].u.norBoot.blob[1][1].branchAddress = 0x80000000;    /* Image 1 branch address after Loading in BE mode
    < */
1801
1802  ibl.bootModes[1].bootMode = ibl_BOOT_MODE_NONE;
1803
1804  ibl.bootModes[2].bootMode = ibl_BOOT_MODE_TFTP;
1805  ibl.bootModes[2].priority = ibl_HIGHEST_PRIORITY+1;
1806  ibl.bootModes[2].port = ibl_PORT_SWITCH_ALL;
1807
1808  ibl.bootModes[2].u.ethBoot.doBootp = TRUE;
1809  ibl.bootModes[2].u.ethBoot.useBootpServerIp = TRUE;
1810  ibl.bootModes[2].u.ethBoot.useBootpFileName = TRUE;
1811  ibl.bootModes[2].u.ethBoot.bootFormat = ibl_BOOT_FORMAT_BLOB;
1812
1813
1814  SETIP(ibl.bootModes[2].u.ethBoot.ethInfo.ipAddr, 192,168,2,100);
1815  SETIP(ibl.bootModes[2].u.ethBoot.ethInfo.serverIp, 192,168,2,101);
1816  SETIP(ibl.bootModes[2].u.ethBoot.ethInfo.gatewayIp, 192,168,2,1);
1817  SETIP(ibl.bootModes[2].u.ethBoot.ethInfo.netmask, 255,255,255,0);
1818
1819  /* Use the e-fuse value */
1820  ibl.bootModes[2].u.ethBoot.ethInfo.hwAddress[0] = 0;
1821  ibl.bootModes[2].u.ethBoot.ethInfo.hwAddress[1] = 0;
1822  ibl.bootModes[2].u.ethBoot.ethInfo.hwAddress[2] = 0;
1823  ibl.bootModes[2].u.ethBoot.ethInfo.hwAddress[3] = 0;
1824  ibl.bootModes[2].u.ethBoot.ethInfo.hwAddress[4] = 0;
1825  ibl.bootModes[2].u.ethBoot.ethInfo.hwAddress[5] = 0;
1826
1827
1828  ibl.bootModes[2].u.ethBoot.ethInfo.fileName[0] = 't';
1829  ibl.bootModes[2].u.ethBoot.ethInfo.fileName[1] = 's';
1830  ibl.bootModes[2].u.ethBoot.ethInfo.fileName[2] = 'p';
1831  ibl.bootModes[2].u.ethBoot.ethInfo.fileName[3] = 'c';
1832  ibl.bootModes[2].u.ethBoot.ethInfo.fileName[4] = 'i';
1833  ibl.bootModes[2].u.ethBoot.ethInfo.fileName[5] = 'e';
1834  ibl.bootModes[2].u.ethBoot.ethInfo.fileName[6] = '.';
1835  ibl.bootModes[2].u.ethBoot.ethInfo.fileName[7] = 'b';
1836  ibl.bootModes[2].u.ethBoot.ethInfo.fileName[8] = 'i';
1837  ibl.bootModes[2].u.ethBoot.ethInfo.fileName[9] = 'n';
1838  ibl.bootModes[2].u.ethBoot.ethInfo.fileName[10] = '\0';
1839  ibl.bootModes[2].u.ethBoot.ethInfo.fileName[11] = '\0';
1840  ibl.bootModes[2].u.ethBoot.ethInfo.fileName[12] = '\0';
1841  ibl.bootModes[2].u.ethBoot.ethInfo.fileName[13] = '\0';
1842  ibl.bootModes[2].u.ethBoot.ethInfo.fileName[14] = '\0';
1843
1844  ibl.bootModes[2].u.ethBoot.blob.startAddress = 0x80000000;    /* Load start address */
1845  ibl.bootModes[2].u.ethBoot.blob.sizeBytes = 0x20000000;
1846  ibl.bootModes[2].u.ethBoot.blob.branchAddress = 0x80000000;    /* Branch address after Loading */
1847
1848  ibl.chkSum = 0;
1849 }

```

В таблице 4-1 дано краткое описание основных конфигурационных параметров, из файла «i2cConfig.gel», их назначение и возможные значения.

Параметры, которые не описаны в таблице 4-1, связаны с аппаратной конфигурацией оборудования модуля TS-PCIE и их изменение не рекомендуется.

Таблица 4-1: Основные конфигурационные параметры файла «i2cConfig.gel»

Параметр	Описание
ibl.bootModes[2].u.ethBoot.doBootp	Для режима загрузки TFTP. Если равен TRUE, IBL будет пытаться получить сетевую конфигурацию по протоколу BOOTP. Если равно FALSE, то будут использованы параметры конфигурации сети описанные ниже.

Продолжение таблицы на следующей странице

Продолжение таблицы 4-1

Параметр	Описание
<code>ibl.bootModes[2].u.ethBoot.bootFormat</code>	<p>Задаёт формат загружаемого образа. Может принимать одно из следующих значений:</p> <ul style="list-style-type: none"> <code>ibl_BOOT_FORMAT_COFF</code> — объектный формат <code>COFF</code>. Загружается через встроенный в IBL загрузчик <code>COFF</code> файлов; <code>ibl_BOOT_FORMAT_ELF</code> — объектный формат <code>ELF</code>. Загружается через встроенный в IBL загрузчик <code>ELF</code> файлов; <code>ibl_BOOT_FORMAT_BBLOB</code> — бинарный формат готовый к загрузке на модуле (не требующий соответствующего загрузчика); <code>ibl_BOOT_FORMAT_AUTO</code> — автоматическое определение формата по сигнатуре файла; <code>ibl_BOOT_FORMAT_NAME</code> — автоматическое определение формата по расширению загружаемого файла («.out» — <code>COFF</code>, «.elf» — <code>ELF</code>, «.bin» — <code>BBLOB</code>).
<code>ibl.bootModes[2].u.ethBoot.ethInfo.hwAddress[0...5]</code>	Задаёт значение аппаратного MAC адреса сетевого интерфейса. Если все значения равны 0, используется встроенный производителем в процессор MAC-адрес.
<code>ibl.bootModes[2].u.ethBoot.ethInfo.fileName[0...63]</code>	Задаёт имя файла для загрузки. Максимальная длина имени файла составляет 64 символа. Последним символом имени файла загрузки обязательно должен быть символ «\0».
<code>ibl.bootModes[2].u.ethBoot.ethInfo.ipAddr</code>	Определяет значение фиксированного IP адреса. Октеды IP адреса в файле «i2cConfig.geI» записываются через запятую. Значение параметр используется в случае, если <code>ibl.bootModes[2].u.ethBoot.doBootp = FALSE</code> .
<code>ibl.bootModes[2].u.ethBoot.ethInfo.serverIp</code>	Задаёт IP адрес сервера загрузки в случае, если <code>ibl.bootModes[2].u.ethBoot.doBootp = FALSE</code> .
<code>ibl.bootModes[2].u.ethBoot.ethInfo.gatewayIp</code>	Задаёт IP адрес основного шлюза в случае, если <code>ibl.bootModes[2].u.ethBoot.doBootp = FALSE</code> .
<code>ibl.bootModes[2].u.ethBoot.ethInfo.netmask</code>	Задаёт маску подсети в том случае, если <code>ibl.bootModes[2].u.ethBoot.doBootp = FALSE</code> .
<code>ibl.bootModes[2].u.ethBoot.useBootpServerIp</code>	Если равен <code>FALSE</code> , то в качестве IP адреса сервера загрузки будет использовано значение параметра <code>ibl.bootModes[2].u.ethBoot.ethInfo.serverIp</code> . Если равен <code>TRUE</code> , то будет использован IP адрес сервера загрузки, указанный в <code>BOOTP</code> ответе от <code>BOOTP</code> сервера. Данный параметр имеет значение только в том случае, когда <code>ibl.bootModes[2].u.ethBoot.doBootp = TRUE</code> .
<code>ibl.bootModes[2].u.ethBoot.useBootpFileName</code>	Если равен <code>FALSE</code> , то в качестве имени файла для загрузки будет использовано значение параметра <code>ibl.bootModes[2].u.ethBoot.ethInfo.fileName[0...63]</code> . Если равен <code>TRUE</code> , то будет использовано имя файла загрузки, указанное в <code>BOOTP</code> ответе от <code>BOOTP</code> сервера. Данный параметр имеет значение только в том случае, когда <code>ibl.bootModes[2].u.ethBoot.doBootp = TRUE</code> .
<code>ibl.bootModes[2].u.ethBoot.blob.startAddress</code>	Адрес области памяти куда будет осуществляться загрузка образа.
<code>ibl.bootModes[2].u.ethBoot.blob.sizeBytes</code>	Максимальный допустимый объем образа, который допускается загрузить.
<code>ibl.bootModes[2].u.ethBoot.blob.branchAddress</code>	Адрес точки входа, куда будет осуществлен переход после завершения загрузки образа приложения.

5 Импорт и запуск целевой конфигурации модуля

Для загрузки кода приложений на модуль TS-PCIE, в первую очередь, необходимо запустить целевую конфигурацию модуля TS-PCIE. В папке «TargetConfigurations» сопроводительного диска к модулю TS-PCIE находятся файлы целевых конфигураций для различных отладчиков. В данном документе рассматривается загрузка кода через отладчик Spectrum Digital XDS560v2 STM LAN. Данному отладчику соответствует файл целевой конфигурации «TS-PCIE-LAN560v2-SD.ccxml», который необходимо импортировать в рабочее пространство.

Выберите пункт главного меню «View > Target Configurations» (рисунок 5-1)

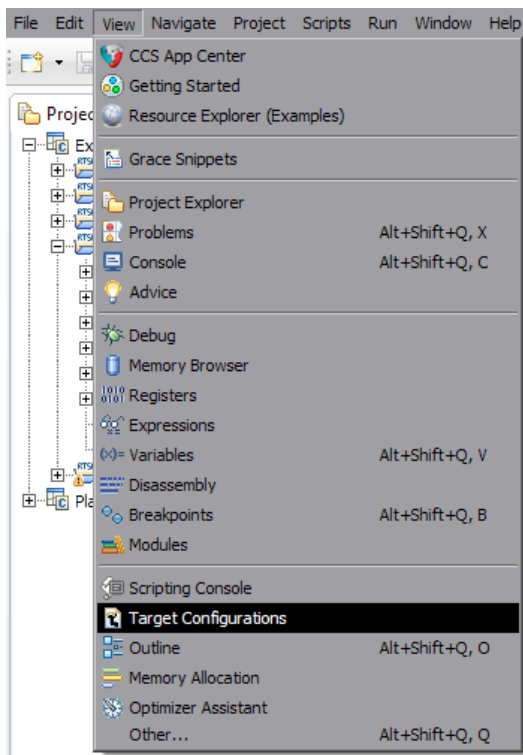


Рисунок 5-1: Пункт меню для отображения окна целевых конфигураций

В окне целевых конфигураций («Target Configurations»), нажмите правой кнопкой мыши на свободной области для вызова контекстного меню и выберите пункт «Import Target Configuration» (см. рисунок 5-2).

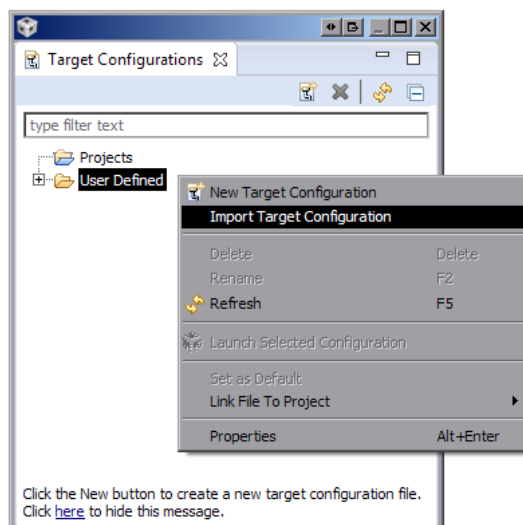


Рисунок 5-2: Меню импорта целевой конфигурации

В появившемся окне выбора файла (см. рисунок 5-3) необходимо выбрать файл «TS-PCIE-LAN560v2-SD.ccxml» и нажать на кнопку «Открыть». В данном документе предполагается, что папка «TargetConfigurations» с сопро-

дителя диска к модулю SVR-420, где расположен файл «TS-PCIE-LAN560v2-SD.ccxml», скопирована в папку «D:/Dev/Modules/TS-PCIE/TargetConfigurations».

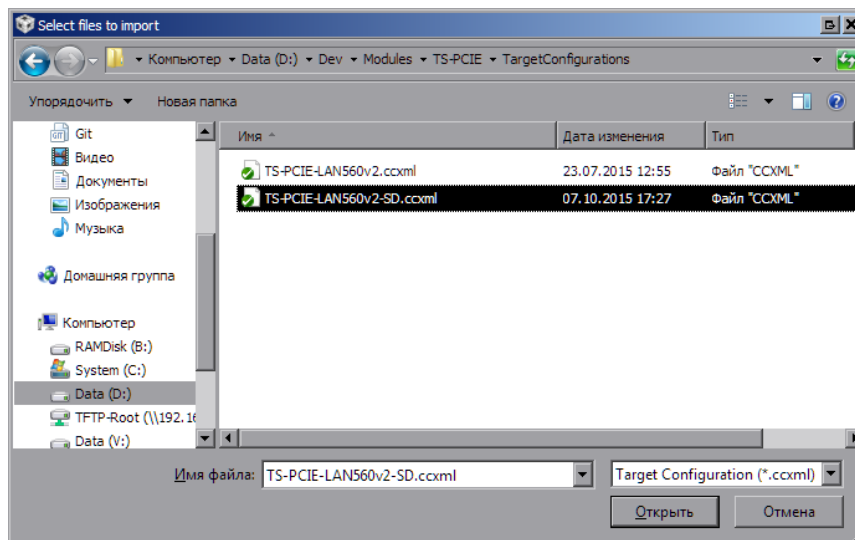


Рисунок 5-3: Окно выбора файла для импорта целевой конфигурации

После нажатия на кнопку «Открыть» появится окно выбора способа импорта файла целевой конфигурации (рисунок 5-4). Необходимо выбрать способ «Link to files» и нажать на кнопку «ОК».

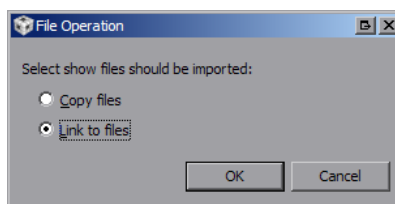


Рисунок 5-4: Окно выбора способа импорта файла целевой конфигурации

Для запуска целевой конфигурации, в окне целевых конфигураций («Target Configurations»), необходимо нажать правой кнопкой мыши на целевой конфигурации и выбрать пункт меню «Launch Selected Configuration» (см. рисунок 5-5).

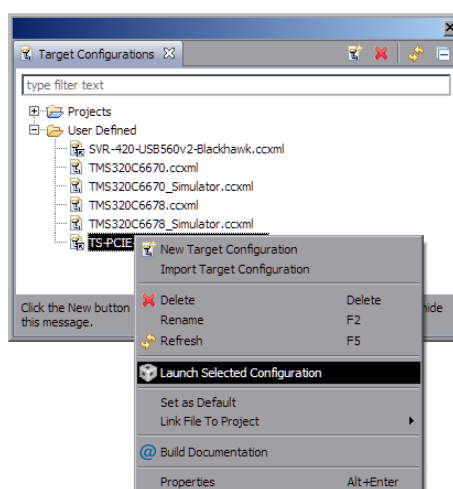


Рисунок 5-5: Запуск целевой конфигурации

После запуска целевой конфигурации модуля TS-PCIE, среда разработки CCS перейдет в режим отладки и в окне «Debug» будет выведен список ядер обоих процессоров модуля TS-PCIE, как показано на рисунке 5-6.

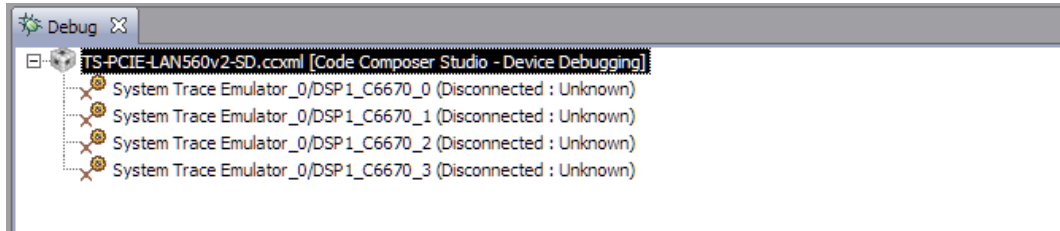


Рисунок 5-6: Список ядер процессоров модуля TS-PCIE

Примечание

По умолчанию, среда CCS настроена таким образом, что при запуске кода сразу на нескольких процессорах или нескольких ядрах одного процессора, вывод (CIO) со всех ядер процессоров будет выводиться в одно и то же окно «Console». В приложении Б даны инструкции по настройке раздельного вывода (CIO) каждого ядра процессора в отдельное окно «Console».

6 Подготовка образов приложений для загрузки на нескольких ядрах процессора

Для подготовки образов приложений, которые необходимо загрузить сразу на несколько ядер процессора, используется набор специальных утилит **MAD** разработанных компанией Texas Instruments.

Набор утилит **MAD** входит в состав **MCSDK** (MultiCore Software Development Kit). Более подробную информацию по возможностям и приемам использования набора утилит **MAD** можно получить ознакомившись с документом [2].

6.1 Компоненты **MAD Utils**

MAD Utils предоставляет набор утилит для достижения следующих целей:

- необходимость выполнения загрузки множества приложений на несколько ядер;
- необходимость сохранения памяти, путем выделения общего кода многоядерных приложений.

MAD Utils состоит из пяти основных утилит, которые можно разделить на две категории: «Утилиты времени сборки» и «Утилиты времени исполнения».

Утилиты времени сборки (build time utilities) включают в себя:

- **Static linker** — статический линковщик, предназначенный для линковки приложений и зависимых динамических общих объектов **DSO** (Dynamic Shared Object).
- **Prelink Tool** — используется для назначения сегментам **ELF** файла виртуальных адресов.
- **MAP Tool** — используется для назначения виртуальных адресов сегментам многоядерных приложений. Пользователь определяет нужные разделы памяти для устройства и высокоуровневые инструкции сегмента размещения в **MAP Tool**. Основываясь на данной информации, **MAP Tool** определяет виртуальные и физические адреса времени исполнения для каждого сегмента **ELF** файла для каждого приложения. Затем вызывается **Prelink Tool** для выделения области для хранения всех приложений и их **DSO**. **MAP Tool** также генерирует набор активационных записей для загрузки приложения на определенное ядро. Активационные записи — это инструкции загрузчика времени исполнения, которые выполняют следующие действия:
 - настройка карты виртуальной памяти и атрибутов доступа и защиты областей памяти;
 - копирование и инициализация загружаемых сегментов памяти на их адреса времени исполнения.

Полученный образ приложения и активационные записи запаковываются в образ **ROMFS**, который предназначен для загрузки на целевом устройстве.

Утилиты времени исполнения (run time utilities) включают в себя:

- Утилита начальной загрузки. В качестве данной утилиты выступает загрузчик **IBL**, который предоставляет функциональность загрузки образа **ROMFS** в общую внешнюю память устройства (**DDR**).
- **MAD Loader** — утилита загрузки времени исполнения, которая обеспечивает функционал запуска приложений на заданном ядре. Данная утилита выполняет следующие действия для обеспечения запуска приложения на ядре:
 - конфигурация карты виртуальной памяти для ядра;
 - конфигурация атрибутов и режимов доступа для каждого раздела памяти;
 - копирование сегментов памяти с локального адреса в адрес времени исполнения;
 - выполнение прединициализационных функций приложения;
 - выполнение инициализационных функций зависимых библиотек и приложений;
 - запуск приложения (переход по адресу точки входа).

6.2 Режимы работы MAD Utils

MAD Utils позволяет работать в двух режимах:

- **Prelinker bypass mode.** В данном режиме утилита MAP Tool не выполняет назначения адресов сегментам приложения и Prelink Tool не вызывается. Данный режим подходит в тех случаях, когда просто требуется выполнить загрузку приложения или нескольких приложений на конкретном ядре или ядрах.
- **Prelinker mode.** В данном режиме утилита MAP Tool выполняет назначение адресов сегментам приложения и вызывает Prelink Tool. Данный режим подходит в тех случаях, когда разработчику требуется, чтобы MAP Tool выполнил присвоение адресов для общего кода между несколькими ядрами, на которых должно работать приложение.

Внимание



В данном документе рассматривается только работа MAD Utils в режиме работы «Prelinker bypass mode». Информацию по работе с MAD Utils в режиме «Prelinker mode» можно получить обратившись к документу [2].

6.3 Работа с MAD Utils в режиме «Prelinker bypass mode»

На рисунке 6-1 изображена схема работы MAD Utils в режиме «Prelinker bypass mode».

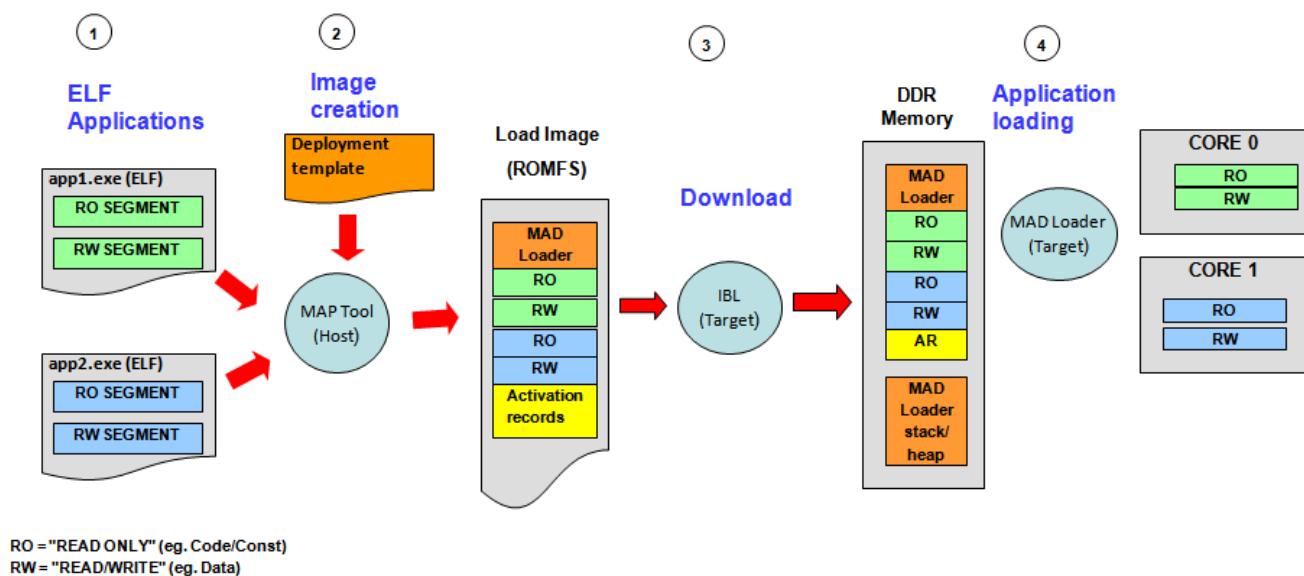


Рисунок 6-1: Схема работы MAD в режиме «Prelinker bypass mode»

Работа с MAD Utils может быть разделена на два этапа — этап подготовки образа для загрузки и этап загрузки образа на целевом устройстве.

Этап подготовки образа состоит из следующих шагов:

- Сборка приложения (статическая линковка по адресам исполнения);
- Создание файла конфигурации развертывания для MAP Tool с определением приложения для каждого ядра.
- Запуск MAP Tool с файлом конфигурации развертывания в качестве входных данных.
- MAP Tool создает образ для загрузки (в формате ROMFS), содержащий в себе активационные записи для каждого приложения.

Этап загрузки образа на целевом устройстве выглядит следующим образом:

- При загрузке, в первую очередь, на устройстве выполняется код загрузчика из ROM памяти. Этот загрузчик выполняет загрузку IBL, который должен быть записан в I²C EEPROM память устройства.
- IBL выполняет загрузку MAD образа с TFTP сервера или NOR флеш памяти в DDR память. При этом, IBL должен быть сконфигурирован таким образом, что бы переход осуществлялся по адресу точки входа MAD Loader.

- MAD Loader выполняет разбор образа ROMFS и выполняет загрузку сегментов приложения по их адресам времени исполнения для каждого ядра.
- MAD Loader выполняет переход по адресу точки входа для каждого ядра, запуская таким образом выполнения приложения на ядрах целевого устройства.

6.4 Конфигурация MAP Tool

Входными данными для MAP Tool является конфигурационный файл в формате JSON. Данный конфигурационный файл должен содержать параметры, приведенные в таблице 6-1.

Таблица 6-1: Параметры конфигурационного файла MAP Tool

Параметр	Описание
deploymentCfgFile	Путь к конфигурационному файлу развертывания.
LoadImageName	Имя файла образа, который будет сгенерирован. Данный файл образа (в формате ROMFS) будет помещен в папку «images».
prelinkExe	Путь к запускаемому файлу Prelink Tool. Данная утилита является частью CGT (Code Generation Tools).
stripExe	Путь к запускаемому файлу Strip Tool. Данная утилита является частью CGT.
ofdTool	Путь к запускаемому файлу OFD Tool. Данная утилита является частью CGT.
malApp	Путь к файлу приложения MAD Loader.
nmlLoader	Путь к файлу приложения NML Loader. NML Loader является составной частью MAD Loader.

В листинге 6-1 приведен пример конфигурационного файла для MAP Tool.

Листинг 6-1: Пример конфигурационного файла для MAP Tool

```

1 {
2   "deploymentCfgFile" : ".\\deploy.json",
3   "LoadImageName"    : "mad_test.bin",
4   "prelinkExe"       : "C:\\ti\\ccsv5\\tools\\compiler\\c6000_7.4.2\\bin\\prelink6x",
5   "stripExe"         : "C:\\ti\\ccsv5\\tools\\compiler\\c6000_7.4.2\\bin\\strip6x",
6   "ofdTool"          : "C:\\ti\\ccsv5\\tools\\compiler\\c6000_7.4.2\\bin\\ofd6x.exe",
7   "malApp"           : "C:\\ti\\mcsdk_2_01_02_06\\tools\\boot_loader\\mad-utils\\mad-loader\\bin\\C6670\\le\\mal_app.exe",
8   "nmlLoader"        : "C:\\ti\\mcsdk_2_01_02_06\\tools\\boot_loader\\mad-utils\\mad-loader\\bin\\C6670\\le\\nml.exe"
9 }

```

В листинге 6-1 указаны пути с учетом следующих предположений:

- используется MCSDK версии 2.01.02.06 установленная в папку «C:/ti/mcsdk_2_01_02_06»;
- используется компилятор версии 7.4.2 установленный в папку «C:/ti/ccsv5/tools/compiler/c6000_7.4.2»;
- файл конфигурации развертывания «deploy.json» находится в той же папке.

6.5 Конфигурационный файл развертывания

В режиме «Prelinker bypass mode» конфигурационный файл развертывания определяет следующую информацию:

- области памяти для загрузки;
- данные приложений для развертывания.

Файл конфигурации развертывания является файлом в формате JSON. Для режима «Prelinker bypass mode» файл должен содержать параметры, приведенные в таблице 6-2.

Таблица 6-2: Параметры файла конфигурации развертывания

Параметр	Описание
deviceName	JSON объект, определяющий целевое устройство. Может принимать следующие значения: <ul style="list-style-type: none"> • C6657 — 2-х ядерный процессор TMS320C6657; • C6670 — 4-х ядерный процессор TMS320C6670; • C6678 — 8-и ядерный процессор TMS320C6678;

Продолжение таблицы на следующей странице

Продолжение таблицы 6-2

Параметр	Описание
partitions	<p>Данный параметр идентифицирует разделы памяти, которые будут загружены из ROMFS образа в память устройства при загрузке. Определение каждого из разделов должно содержать следующие параметры:</p> <ul style="list-style-type: none"> • name — имя раздела. Используется для идентификации раздела памяти в отладочном выводе MAP Tool; • vaddr — виртуальный адрес раздела памяти; • size — размер области памяти в байтах; • loadPartition — определяет будет ли данный раздел загружен или нет.
applications	<p>Данный параметр идентифицирует приложения для развертывания. Определение каждого приложения должно содержать следующие параметры:</p> <ul style="list-style-type: none"> • name — имя приложения. Используется для идентификации в отладочном выводе; • fileName — имя файла образа приложения полученного в результате сборки приложения; • allowedCores — номера ядер, на которых разрешен запуск данного приложения.
appDeployment	<p>Данный параметр определяет приложения для каждого из ядер целевого устройства. Представляет из себя массив (размер массива должен соответствовать количеству ядер целевого устройства), каждый элемент которого должен содержать имя приложения, которое необходимо загрузить на соответствующем ядре целевого устройства. В том случае, если на каком либо ядре загружать приложение не требуется, соответствующий элемент данного массива задается в виде пустой строки ("").</p>

Листинг 6-2: Пример файла конфигурации развертывания

```

1  {
2    "deviceName" : "C6670",
3    "partitions" : [
4      {
5        "name" : "load-partition",
6        "vaddr" : "0x9e000000",
7        "size" : "0x2000000",
8        "loadPartition" : true
9      }
10   ],
11   "applications" : [
12     {
13       "name" : "app1",
14       "fileName" : "../build/app_1.out",
15       "allowedCores" : [0,1,2,3]
16     },
17     {
18       "name" : "app2",
19       "fileName" : "../build/app_2.out",
20       "allowedCores" : [0,1,2,3]
21     }
22   ],
23   "appDeployment" : [
24     "app1",
25     "app1",
26     "",
27     "app2"
28   ]
29 }

```

6.6 Запуск MAP Tool

Для запуска MAP Tool потребуется установленный Python интерпретатор, который можно бесплатно скачать с официального сайта¹.

Запуск MAP Tool в режиме «Prelinker bypass mode» осуществляется путем выполнения команды:

```
python maptool.py <файл_конфигурации> bypass-prelink
```

Внимание



Следует помнить, что образ, предназначенный для загрузки на целевом устройстве, будет создан в папке «images». Имя файла образа задается в конфигурационном файле MAP Tool (см. раздел 6.4).

В качестве параметра <файл_конфигурации> необходимо указать имя файла конфигурации MAP Tool (см. раздел 6.4). Python скрипт «maptool.py» располагается в папке «tools/boot_loader/mad-utils/map-tool» относительно папки установки MCSDK («C:\ti\mcSDK_2_01_02_06»). Для запуска MAP Tool непосредственно из папки установки MCSDK необходимо выполнять команду указав полный путь к файлу «maptool.py»:

```
python "C:\ti\mcSDK_2_01_02_06\tools\boot_loader\mad-utils\map-tool\maptool.py" <файл_конфигурации>
← bypass-prelink
```

Примечание

Инструкции по записи полученного образа в NOR флеш память приведены в разделе 3.3 данного документа.

Инструкции по установке и настройке служб BOOTP и TFTP, необходимых для осуществления загрузки образа по Ethernet с TFTP сервера, можно найти в документе [3].

6.7 Загрузка образа

Для того, чтобы загрузчик IBL мог корректно загружать образы, подготовленные при помощи MAD Utils, необходимо выполнить его конфигурацию. Подробное описание процедуры конфигурации загрузчика IBL приведено в разделе 4 данного документа.

В том случае, если подготовленный образ планируется загружать по Ethernet с TFTP сервера, то необходимо установить следующие параметры:

```
ib1.bootModes[2].u.ethBoot.bootFormat      = ib1_BOOT_FORMAT_BBLOB;
ib1.bootModes[2].u.ethBoot.blob.startAddress = 0x9e000000;
ib1.bootModes[2].u.ethBoot.blob.sizeBytes   = 0x20000000;
ib1.bootModes[2].u.ethBoot.blob.branchAddress = 0x9e001040;
```

Если же загрузка подготовленного образа будет происходить с NOR флеш памяти, то необходимо установить параметры:

```
ib1.bootModes[0].u.norBoot.bootFormat      = ib1_BOOT_FORMAT_BBLOB;
ib1.bootModes[0].u.norBoot.blob[0][0].startAddress = 0x9e000000;
ib1.bootModes[0].u.norBoot.blob[0][0].sizeBytes   = 0x800000;
ib1.bootModes[0].u.norBoot.blob[0][0].branchAddress = 0x9e001040;
```

Дополнительное описание конфигурационных параметров приведено в таблице 4-1 раздела 4 данного документа.

¹ <https://www.python.org/download/>

Приложение А Примеры работы скрипта записи EEPROM и NOR флеш памяти

В листинге A-1 приведен пример вывода в терминал запуска скрипта «tspcie_program.bat» для записи в NOR флеш память образов по умолчанию на все четыре процессора модуля TS-PCIE.

Примечание

В целях сокращения объема листингов, приведенных в данном приложении, некоторые их части вырезаны. В местах вырезанных данных листингов, помещен текст в виде «ВЫРЕЗАНО: Описание вырезанного текста».

Листинг A-1: Вывод в терминал при запуске команды «tspcie_program.bat NOR all»

```
1 D:\Dev\Modules\TS-PCIE\Program>svr420_program.bat NOR all
2
3 TS-PCIE program script started
4 Copyright (c) 2015, Scan Engineering Telecom SPb
5
6 Writing to DSP1 ENABLED
7
8 Target configuration file:
9   "../TargetConfigurations/TS-PCIE-LAN560v2-SD.ccxml"
10
11 Images for DSP's:
12   DSP1: "bin/nor_c6670.bin" (513648 bytes)
13
14 Configuring debug server for TS-PCIE board...
15 Done!
16 Opening a debug session for 1 DSP(s)...
17 Loaded FPGA Image: C:\ti\ccsv5\ccs_base\common\uscif\dtc_top.jbc
18 Done!
19 Connecting to DSP(s)...
20
21 ВЫРЕЗАНО: Вырезана часть с GEL инициализацией ядер процессоров
22
23 Done!
24 Loading binary images to DSP(s) memory...
25   Loading file "bin/nor_c6670.bin" (513648 bytes)...
26 Done!
27 Configuring writers on DSP(s)...
28 Done!
29 Executing writers on DSP(s)...
30 NOR Writer Utility Version 02.00.00.00
31 Copyright (c) 2014, Scan Engineering Telecom SPb
32
33 Write size:   513648 bytes
34 Start address: 0x0
35
36 Flashing sector 0 (0 bytes of 513648)
37 Flashing sector 1 (65536 bytes of 513648)
38 Flashing sector 2 (131072 bytes of 513648)
39 Flashing sector 3 (196608 bytes of 513648)
40 Flashing sector 4 (262144 bytes of 513648)
41 Flashing sector 5 (327680 bytes of 513648)
42 Flashing sector 6 (393216 bytes of 513648)
43 Flashing sector 7 (458752 bytes of 513648)
44 Reading and verifying sector 0 (0 bytes of 513648)
45 Reading and verifying sector 1 (65536 bytes of 513648)
46 Reading and verifying sector 2 (131072 bytes of 513648)
47 Reading and verifying sector 3 (196608 bytes of 513648)
48 Reading and verifying sector 4 (262144 bytes of 513648)
49 Reading and verifying sector 5 (327680 bytes of 513648)
50 Reading and verifying sector 6 (393216 bytes of 513648)
51 Reading and verifying sector 7 (458752 bytes of 513648)
52 NOR programming completed successfully
53 Done!
54 Terminating debug sessions on DSP(s)...
55 Done!
56 Stopping debug server...
57 Done!
58
59
60 D:\Dev\Modules\TS-PCIE\Program>
```

В листинге A-2 приведен пример вывода в терминал запуска скрипта «tspcie_program.bat» для записи в EEPROM память образов по умолчанию на все четыре процессора модуля TS-PCIE.

Листинг A-2: Вывод в терминал при запуске команды «tspcie_program.bat EEPROM all»

```
1
2 D:\Dev\Modules\TS-PCIE\Program>tspcie_program.bat EEPROM all
3
4 TS-PCIE program script started
5 Copyright (c) 2015, Scan Engineering Telecom SPb
6
7 Writing to DSP1 ENABLED
8
9 Target configuration file:
10  "../TargetConfigurations/TS-PCIE-LAN560v2-SD.ccxml"
11
12 Images for DSP's:
13   DSP1: "bin/eeeprom_0x50_c6670.bin" (60492 bytes)
14   DSP2: "bin/eeeprom_0x50_c6670.bin" (60492 bytes)
15
16 Configuring debug server for TS-PCIE board...
17 Done!
18 Opening a debug session for 1 DSP(s)...
19 Loaded FPGA Image: C:\ti\ccsv5\ccs_base\common\uscif\dtc_top.jbc
20 Done!
21 Connecting to DSP(s)...
22
23 ВЫРЕЗАНО: Вырезана часть с GEL инициализацией ядер процессоров
24
25 Loading binary images to DSP(s) memory...
26   Loading file "bin/eeeprom_0x50_c6670.bin" (60492 bytes)...
27 Done!
28 Configuring writers on DSP(s)...
29 Done!
30 Executing writers on DSP(s)...
31 EEPROM Writer Utility Version 02.00.00.00
32 Copyright (c) 2014, Scan Engineering Telecom SPb
33
34 Write size:          60492 bytes
35 I2C bus address:    0x50
36 I2C start address: 0x0
37 Swap data:         no
38
39 Writing 60492 bytes from DSP memory address 0x0c000100 to EEPROM (0x0050)...
40 Reading 60492 bytes from EEPROM (0x0050) to DSP memory address 0x0c010100...
41 Verifying data read...
42 EEPROM programming completed successfully
43 Done!
44 Terminating debug sessions on DSP(s)...
45 Done!
46 Stopping debug server...
47 Done!
48
49
50 D:\Dev\Modules\TS-PCIE\Program>
```

Приложение Б Разделение вывода сообщений (CIO) ядер процессоров

По умолчанию, среда CCS настроена таким образом, что при запуске кода сразу на нескольких процессорах или нескольких ядрах одного процессора, вывод (CIO) со всех ядер процессоров будет выводиться в одно и то же окно «Console». В данном приложении даны инструкции по настройке отдельного вывода (CIO) каждого ядра процессора в отдельное окно «Console».

После запуска целевой конфигурации, в окне «Debug», нажмите правой кнопкой мыши на названии файла целевой конфигурации и выберите пункт меню «Edit X...», где X — имя файла целевой конфигурации (см. рисунок Б-1).

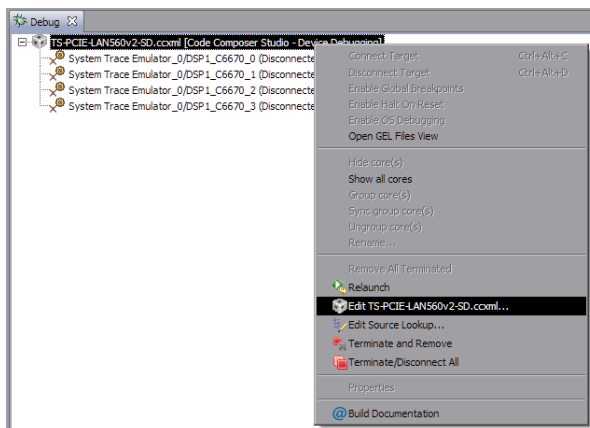


Рисунок Б-1: Контекстное меню целевой конфигурации

В открывшемся окне, снимите галочку с опции «Use the same console for the CIO of all CPUs» (см. рисунок Б-2) и нажмите на кнопки «Apply» и «Continue».

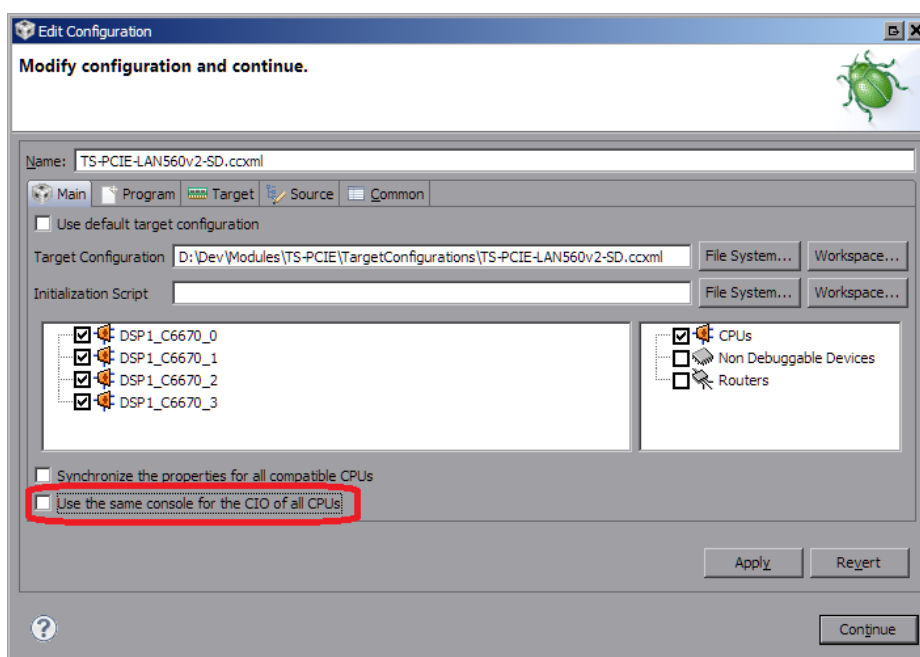



Рисунок Б-2: Окно настроек целевой конфигурации

В окне «Console» нажмите на кнопку  («Open Console») и выберите пункт меню «New Console View» (см. рисунок Б-3).

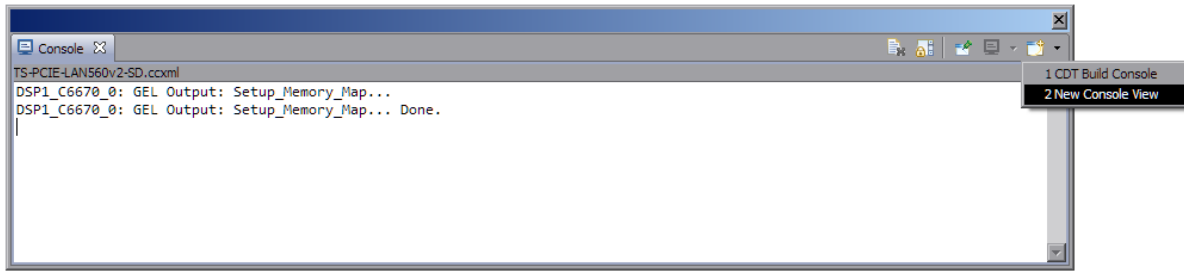


Рисунок Б-3: Открытие второго окна «Console»

После этого, будет открыто еще одно окно «Console», которое можно переместить в любое удобное место окна CCS, например, как показано на рисунке Б-4.

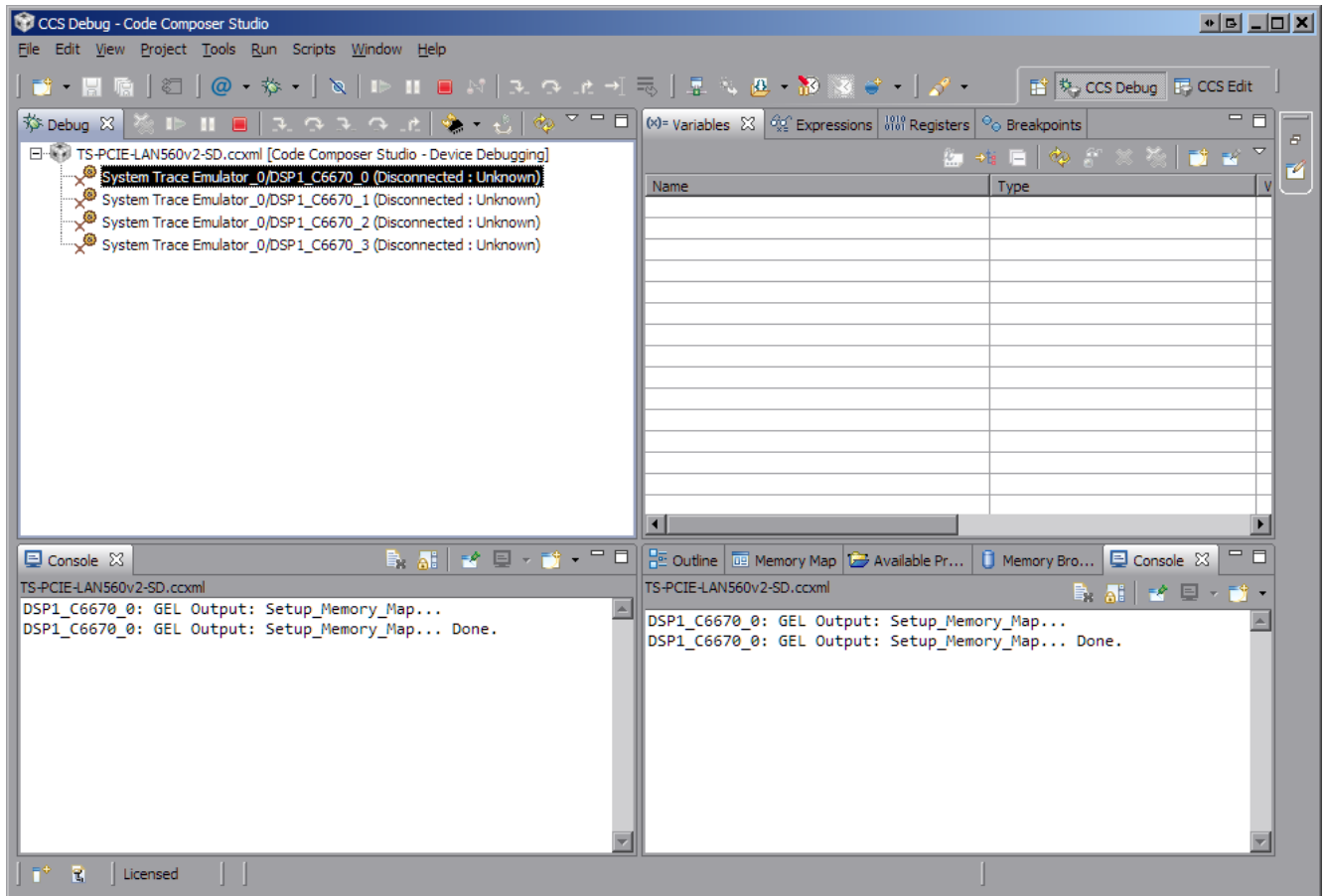



Рисунок Б-4: Два окна «Console»

Теперь, если запустить приложения на различных ядрах процессоров, для вывода с каждого отдельного ядра будет происходить в отдельное окно. Для того, чтобы выбрать вывод какого ядра необходимо отображать в конкретном окне «Console», необходимо нажать на кнопку  («Display Selected Console») этого окна и выбрать пункт меню соответствующий нужному ядру (см. рисунок Б-5).

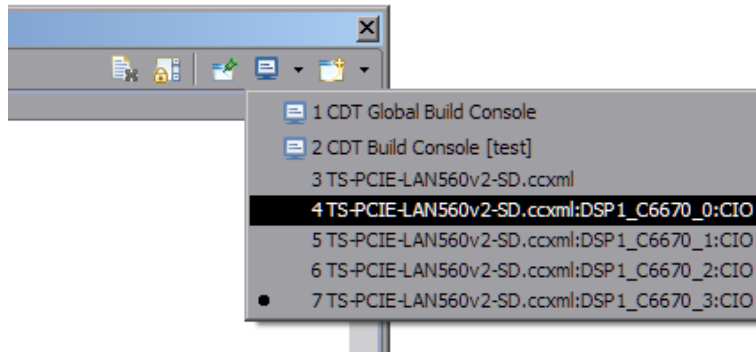


Рисунок Б-5: Выбор ядра для отображения вывода в окне «Console»

Следует иметь в виду, что в данном списке (рисунок Б-5) можно выбрать только те ядра, с которых уже был произведен какой либо вывод.

Для закрепления окна «Console» за конкретным ядром используется кнопка  («Pin Console»).

Приложение В Выбор режима загрузки IBL

Выбор режима загрузки IBL осуществляется при помощи переключателей S10.1–2 на плате модуля TS-PCIE.

В таблице В-1 представлены положения переключателей S10.1–2 для всех возможных режимов загрузки IBL процессора модуля TS-PCIE.

Таблица В-1: Положение переключателей модуля TS-PCIE для установки режимов загрузки IBL

Режим	Переключатель	
	S10.1	S10.2
NOBOOT (DEBUG)	OFF	OFF
NOR	OFF	ON
TFTP (BOOTP)	ON	OFF

Для режима NOBOOT устанавливаются значения $BOOTMODE[7:0] = 0000000b$ для соответствующего процессора. Для режима NOR устанавливаются значения $BOOTMODE[7:0] = 00000101b$ для соответствующего процессора. Для режима TFTP устанавливаются значения $BOOTMODE[7:0] = 00100101b$ для соответствующего процессора. для соответствующего процессора.

Примечание

$BOOTMODE[7:0]$ является частью регистра $DEVSTAT[8:1]$, а $PCIESSMODE[1:0]$ является частью регистра $DEVSTAT[15:14]$. В процессоре TMS320C6670 32-х битный регистр $DEVSTAT$ доступен по адресу 0x02620020.

Список литературы

1. TS-PCIE. Сборка и запуск приложения веб-сервера. Руководство пользователя. [UG-TS-PCIE-WEB](#) (цит. на с. 13, 16).
2. Multicore Application Deployment (MAD) Utilities. User's Guide.
URL: http://processors.wiki.ti.com/index.php/MAD_Utils_User_Guide (цит. на с. 33, 34).
3. Установка и настройка сервера сетевой загрузки (BOOTP и TFTP). Руководство пользователя.
[UG-CMN-BOOTP-TFTP](#) (цит. на с. 37).