

Инструменты командной строки Altera для разработки и отладки программы

В этой секции описываются инструменты командной строки Altera для разработки и отладки программы.

nios2-terminal

Эта команда устанавливает соединения с подсистемами **stdin**, **stdout** и **stderr** процессора Nios II. Подсистемы **stdin**, **stdout** и **stderr** разводятся через UART (стандартный UART или JTAG UART) внутри системы.

Команда **nios2-terminal** предоставляет вам мониторинг над обоими **stdout**, **stderr**, и предоставляет вход к процессорной подсистеме Nios II через **stdin**. Эта команда работает также, как и команда **nios2-configure-sof**, описанная в секции "nios2-configure-sof" на стр. 4-3, в отношении JTAG кабелей и чипов. Однако в системе может быть несколько модулей JTAG UART, поэтому команде **nios2-terminal** необходимо указать корректный модуль JTAG UART. Определите модуль с помощью опции командной строки **-instance**. Первый модуль в вашем проекте - 0 (**-instance "0"**). Остальные модули имеют инкрементную нумерацию, начиная с 1 (**-instance "1"**).

nios2-download

Эта команда анализирует Nios II **.elf** файлы, загружает их в работающий процессор Nios II, а также опционально запускает **.elf** файл.

Также как и для других команд, вы можете использовать опцию информации в командной строке, с помощью опции **--help**. Команда **nios2-download** имеет те же самые опции, что и команда **nios2-terminal** для работы с несколькими JTAG кабелями и процессорными подсистемами Nios II.

Пример использования nios2-download

Для загрузки (и запуска) Nios II **.elf** программы:

1. Откройте командную среду Nios II.
2. Смените на директорию размещения вашего **.elf** файла. Если вы используете для работы Nios II IDE, правильное расположение - в поддиректориях **Debug** или **Release** вашего проекта верхнего уровня.
3. В командной среде введите следующую команду для загрузки и запуска вашей программы:

```
nios2-download -g <project name>.elf ↵
```

4. Вы также можете использовать команду для отображения выхода или предоставления входа для вашей запущенной программы.

nios2-stackreport

Эта команда возвращает письменный отчет об объеме памяти, доступной для стека и кучи из вашего **.elf** файла.

Эта команда не позволит вам определить объем стека или кучи вашего кода, расходуемого на стадии прогона, но она позволит вам знать объем памяти под работу вашего кода.

В примере 4-4 показана информация, отображаемая этой командой.

Example 4–4. nios2-stackreport Command and Response

```
[SOPC Builder]$ nios2-stackreport <your project>.elf
Info: (<your project>.elf) 6312 KBytes program size (code + initialized data).
Info:                      10070 KBytes free for stack + heap.
```

Пример использования nios2-stackreport

Для использования команды nios2-stackreport выполните следующие пункты:

1. Откройте командную среду Nios II.
2. Смените директорию на директорию размещения вашего **.elf** файла.
3. В командной строке введите команду:

```
nios2-stackreport <your project>.elf ↵
```

validate_zip

Nios II IDE использует эту команду для подтверждения того, что файлы, которые вы используете для файловой системы Zip только для чтения являются не сжатыми. Вы можете использовать её для этого же.

Пример использования validate_zip

Для использования команды validate_zip выполните следующие пункты:

1. Откройте командную среду Nios II.
2. Смените директорию на директорию размещения вашего **.zip** файла.
3. В командной строке введите команду:

```
validate_zip <file>.zip ↵
```

Если отклик отсутствует, то ваш **.zip** файл является не сжатым.

nios2-ide

На платформах Linux и Windows вы сможете ввести nios2-ide в командной строке для запуска Nios II IDE. На платформе Windows вы можете использовать иконку запуска Nios II IDE в SOPC Builder.

Команда nios2-ide не может напрямую вызывать исполняемый файл, взамен этого, она запускает простой скрипт-упаковщик среды Bourne, который вызывает исполняемый файл **nios2-ide**. Версии скрипта-упаковщика для платформ Linux и Windows приведены ниже.

Скрипт-упаковщик для Linux

```
#!/bin/sh
# This is the linux-gtk version of the nios2-ide launcher script
# set the default workspace location for linux
WORKSPACE="$HOME/nios2-ide-workspace-7.2"
WORKSPACE_ARGS="-data $WORKSPACE"
# if -data is already passed in, we can't specify it
# again when calling nios2-ide
for i in $*
do
    if [ "x$i" = "x-data" ]; then
```

```

        WORKSPACE_ARGS=""
    fi
done
exec $SOPC_KIT_NIOS2/bin/eclipse/nios2-ide -configuration
$HOME/.nios2-ide-6.1 $WORKSPACE_ARGS "$@"

```

Скрипт-упаковщик для Windows

```

#!/bin/sh
# This is the win32 version of the nios2-ide launcher script
# It simply invokes the binary with the same arguments as
# passed in.
# By doing this, the user will default to the same workspace as
# when launched using the Windows shortcut, as "persisted"
# in the configuration/.settings/org.eclipse.ui.ide.prefs file.
cd "$SOPC_KIT_NIOS2/bin/eclipse"
exec ./nios2-ide-console "$@"

```

nios2-gdb-server

Эта команда запускает отладчик GNU (GDB) - JTAG отвод, который ожидает по специальному TCP порту подключения клиента GDB, например, клиента nios2-elf-gdb. Иногда вы можете завершать сессию с сервером GDB. Если вы длительное время не собираетесь иметь доступ к командной среде Nios II, в которой запущена сессия сервера GDB, или во время сессии отладки в Nios II IDE определён источник сбоя, вы можете остановить процесс **nios2-gdb-server.exe** на платформе Windows, или ввести следующую команду на платформе Linux:

```
kill -9 -f nios2-gdb-server
```

Пример использования nios2-gdb-server

Nios II IDE и большинство других доступных отладчиков используют для отладки команды nios2-gdb-server и nios2-elf-gdb. И вы можете никогда не использовать эти инструменты на этом низком уровне. Однако, если вы предпочитаете делать это, в этой секции будет рассказано, как использовать эти команды, а также будут приведены примеры сессии отладки GDB.

Вам необходимо выполнить следующие пункты для запуска сессии отладки GDB:

1. Откройте командную среду Nios II.
2. В командной строке введите следующую команду для запуска сервера GDB на машине, подключенной через интерфейс JTAG к отлаживаемой системе Nios II:

```
nios2-gdb-server --tcpport 2342 --tcpersist
```

Если порт протокола 2342 уже использован, выберите другой порт. Ответ системы будет следующий:

```
Using cable "USB-Blaster [USB-0]", device 1, instance 0x00
```

```
Pausing target processor: OK
```

```
Listening on port 2342 for connection from GDB:
```

Теперь вы можете подключить свой сервер (локальный или удалённый) и запустить отладку.

3. Введите следующую команду для запуска GDB клиента для вашего .elf файла:

```
nios2-elf-gdb <file>.elf
```

В примере 4-5 отображён пример сессии.

Example 4-5. Sample Debugging Session

```
GNU gdb 6.1
Copyright 2004 Free Software Foundation, Inc.
GDB is free software, covered by the GNU General Public License, and you are
welcome to change it and/or distribute copies of it under certain conditions.
Type "show copying" to see the conditions.
There is absolutely no warranty for GDB. Type "show warranty" for details.
This GDB was configured as "--host=i686-pc-cygwin --target=nios2-elf"...
(gdb) target remote <your_host>:2342
Remote debugging using <your_host>:2342
OS_TaskIdle (p_arg=0x0) at sys/alt_irq.h:127
127 {
(gdb) load
Loading section .exceptions, size 0x1b0 lma 0x1000020
Loading section .text, size 0x3e4f4 lma 0x10001d0
Loading section .rodata, size 0x4328 lma 0x103e6c4
Loading section .rwdata, size 0x2020 lma 0x10429ec
Start address 0x10001d0, load size 281068
Transfer rate: 562136 bits/sec, 510 bytes/write.
(gdb) step
.
.
.
(gdb) quit
```

Конечно команды содержат стандартные команды отладчика: load, step, continue, run и quit. Нажмите Ctrl+c для завершения сессии GDB сервера.

nios2-debug

Эта команда является упаковщиком Tcl/Tk базы Insight GDB GUI, которая устанавливается с пакетом Altera GNU GDB, являющимся частью Nios II EDS. Опция командной строки -save-gdb-script сохраняет скрипт сессии, а опция -command=<GDB script name> восстанавливает предыдущую сессию GDB, исполняя предыдущий сохранённый GDB скрипт. Используйте эту опцию для восстановления точек останова и точек наблюдения.

За информацией о Insight GDB GUI обратитесь к документации Insight, доступной по адресу: sources.redhat.com.

Пример использования nios2-debug

Когда вы вручную или с помощью Nios II IDE сгенерировали .elf файл, выполните следующие пункты для открытия сессии Insight отладчика:

1. Откройте командную среду Nios II.
2. Смените директорию на директорию размещения вашего .elf файла.

Если вы используете для работы Nios II IDE, правильное расположение в поддиректориях **Debug** или **Release** вашего головного проекта.

3. В командной строке введите команду:

```
nios2-debug <file>.elf ↵
```

Ваш .elf файл анализируется и загружается в память вашей подсистемы Nios II, затем открывается основное окно отладчика с выделенной первой исполняемой строкой в функции main(). В этом окне отладчика отображается Insight сессия отладки.

Просто кликайте на элемент меню **Continue** для запуска вашего кода, или установите несколько точек останова для эксперимента.