



This document addresses known errata and documentation issues for the Nios® II Embedded Design Suite (EDS) version 6.0. Errata are functional defects or errors, which might cause the product to deviate from published specifications. Documentation issues include errors, unclear descriptions, or omissions from current published specifications or product documents. Errata items discovered after the release of Nios II EDS version 6.0 are marked with the date the items were added to this document.

Table of Contents:

- Nios II Processor Core..... 3**
 - "nios_cpu: Unknown Break Location nios_cpu/jtag_debug_module" error message [June 2006]..... 3
 - VHDL sensitivity list warnings during Quartus II synthesis [June 2006]..... 3
 - Double-precision Floating-Point operations with floating-point custom instructions..... 3
- Peripherals..... 3**
 - JTAG UART is unstable after device-wide reset..... 3
- Host Platform..... 4**
 - Linux: Out of memory error in IDE..... 4
 - Linux: F1 Help in the Nios II IDE does not function on Linux 4
 - Linux: Debugging with the Nios II ISS target can cause a process leak on Linux..... 4
 - Linux: The Quartus II stand-alone programmer is not supported on Linux..... 4
 - Windows: Frisk antivirus software causes SOPC Builder and Nios II Command Shell to be unresponsive 4
- Device..... 4**
 - Stratix II EP2S60 ES devices cannot use MRAM byte enables..... 5
- Nios II IDE 5**
 - Building Projects 5
 - Nios II IDE unnecessarily updates the SOPC Builder system file (.ptf) [June 2006] 5
 - "FATAL: can't create obj/<object filename>.o: Permission denied" error message [June 2006] 5
 - Incorrect address assignment for dual-port memory mastered by two different masters [June 2006]..... 5
 - Building Nios II Advanced project using Build Project invokes 'make clean all' by default 5
 - Build errors after changing component names in SOPC Builder 6
 - Debugging Projects..... 6
 - "Step failed. Target is not responding (timed out)" error message [June 2006]..... 6
 - Incorrect breakpoint filtering on threads..... 6
 - Uninitialized Memory Error when executing from ISS 6
 - Nios2-gdb-server fails to terminate after setting a watch point..... 6
 - Watchpoints do not work when set on variables whose size are not 32-bits..... 6
 - Debugger cannot step into __sflags, and continues execution instead 7
 - Missing traced load/store instruction and data in the Trace view..... 7
 - Cannot use watchpoints in the Nios II IDE when the FS2 console is open..... 7
 - Breakpoints on adjacent lines of assembly fail to halt the processor 7
 - Navigating Projects..... 7
 - Nios II IDE might not prompt for workspace at startup..... 7
 - Nios II IDE stops responding after you double-click a .o file 8
 - Error importing a software project created in Nios II IDE version 5.1 into Nios II IDE version 6.0..... 8
 - Resource(s) out of sync with the file system when searching for files in the workspace..... 8
 - C/C++ Scanner does not understand certain C/C++ constructs 8
- C2H Compiler 8**

Accelerator returns -1 if C/C++ project doesn't rebuild before running	8
Incorrect results from logical or conditional operation with side-effects	9
Closed system library while working with the C2H Compiler	9
Launch SOPC Builder button in C2H view	9
Build clean causes build failure	9
Multiple clock domains causes hardware accelerator to fail	9
Comma operators are not supported	9
Array elements of structs do not copy correctly	10
Simulating on ISS is not supported	10
Hardware accelerators remain after deleting the software project	10
Changing build configurations produces unexpected results	11
Flash Programmer	11
elf2flash elf size limit	11
Download Cables & Debug Hardware	11
Communication errors during run/debug sessions using older download cables	11
Development Boards	11
Intermittent failures while accessing CompactFlash card	12
Toolchain (gcc, gdb, etc.)	12
Breakpoints in C++ constructors fail to halt the processor	12
Target Software	12
malloc(), realloc() failures with MicroC/OS-II	12
cout from MicroC/OS-II task will not send data to STDOUT	13
Problems using HAL drivers with Toshiba Flash	13
Creating new custom HAL components	13
Legacy SDK	13
SOPC Builder and Quartus II Software	13
Cannot connect Nios II tightly-coupled instruction and data masters to the same dual-port memory	13
Example Designs	14
Hardware Designs	14
Software Designs	14
RAM test failure when running Memory Test software template on the ISS	14
Networking Examples	14
Hardware Simulation	14
Simulation failure if reset address is set to EPCS	14
Uninitialized BSS variables in simulation	14
ModelSim fails to load large memory models	15
Documentation Issues	15
EPCS Controller memory usage	15
Error in UART Core with Avalon Interface Chapter	15
Nios II IDE help system fails to display content	15
byteenables in Avalon Interface Specification	16
burstcount in Avalon Interface Specification	16
Contact Information	16
Revision History	16



For the most up-to-date errata for this release, refer to the errata sheet on the Altera website:
http://www.altera.com/literature/es/es_nios2eds_60.pdf

Nios II Processor Core

This section lists any issues related to the Nios II processor cores.

"nios_cpu: Unknown Break Location nios_cpu/jtag_debug_module" error message [June 2006]

If you modify the Nios II core in an existing system by disabling the JTAG debug module in the Nios II configuration wizard, you might receive the error message "**nios_cpu: Unknown Break Location nios_cpu/jtag_debug_module**" during SOPC Builder generation.

Workaround: Delete the Nios II core from the system and re-add it, ensuring that the JTAG debug core is disabled in the wizard.

VHDL sensitivity list warnings during Quartus II synthesis [June 2006]

You may receive warnings similar to the following in the Quartus II software when compiling a VHDL Nios II system containing the JTAG debug module.

```
Warning (10492): VHDL Process Statement warning at
cpuname_jtag_debug_module.vhd(254): signal "usr1" is read inside the Process
Statement but isn't in the Process Statement's sensitivity list
Warning (10492): VHDL Process Statement warning at
cpuname_jtag_debug_module.vhd(254): signal "ena" is read inside the Process
Statement but isn't in the Process Statement's sensitivity list
```

These warnings are benign and can be ignored. The warnings are a result of the `ena` and `usr1` signals not being included in the debug module's sensitivity list.

Double-precision Floating-Point operations with floating-point custom instructions

Calls to double-precision floating-point functions in `math.h` will return less-precise results on Nios II processors using the floating-point custom instruction. Floating-point constants are forced to single-precision when the floating-point custom instructions is present, which affects the constants for the double-precision floating-point functions in `libm`.

Peripherals

This section lists any issues related to the Altera embedded peripherals included in the Quartus II software.

JTAG UART is unstable after device-wide reset

If the `DEV_CLRn` pin on the FPGA input has been assigned (in Quartus® II software) to generate a device-wide reset, and the FPGA is reset while the JTAG UART is active, then the JTAG UART might become unstable.

Workaround: Do not use the `DEV_CLRn` function on the FPGA. Turn off the Enable device wide reset (`DEV_CLRn`) setting in Quartus II software.

Host Platform

This section lists any issues related specifically to the host platform.

Linux: Out of memory error in IDE

On Linux, you might receive an out of memory error in the Nios II IDE when it is launched from SOPC Builder.

Workaround: Run the IDE directly from the command-line.

Linux: F1 Help in the Nios II IDE does not function on Linux

There is currently no workaround. This will be addressed in a future version.

Linux: Debugging with the Nios II ISS target can cause a process leak on Linux

If you try to interrupt or terminate a debug session targeting the Nios II instruction set simulator (ISS), you might see an error message "Interrupt Failed or Terminate Failed". This means that the nios2-iss process failed to terminate. The debug session appears to have terminated in the IDE, but the nios2-iss process still remains alive.

Workaround: Open a command shell and kill the nios2-iss process.

Linux: The Quartus II stand-alone programmer is not supported on Linux

There is no Quartus II stand-alone programmer for Linux. As a result, in the Nios II IDE the Quartus II Programmer command on the Tools menu has no effect. The IDE does not automatically launch the programmer when you attempt to download software to a board that does not match the expected hardware image.

Workaround: Launch the Quartus II software to access the Quartus II Programmer.

Windows: Frisk antivirus software causes SOPC Builder and Nios II Command Shell to be unresponsive

The SOPC Builder and Nios II Command Shell might become unresponsive if run while the Frisk antivirus software is running.

Workaround: Turn off the Dynamic Virus Checking feature of the Frisk software before running SOPC Builder or the Nios II Command Shell.

Device

This section lists any device-related issues.

Stratix II EP2S60 ES devices cannot use MRAM byte enables

Early shipments of the Nios II Development Kit, Stratix II Edition include an EP2S60 engineering sample (ES) device. Stratix II EP2S60 ES devices have a silicon problem that prevents the use of byte enables on MRAM blocks.

Workaround: Refer to the Stratix II FPGA Family Errata Sheet for details.

Nios II IDE

This section lists any issues relating to the Nios II IDE.

Building Projects

Nios II IDE unnecessarily updates the SOPC Builder system file (.ptf) [June 2006]

The Nios II IDE opens the SOPC Builder system file (**.ptf**) by invoking SOPC Builder during certain operations, which might cause SOPC Builder to change the date stamp of the file even though the system was not modified. This might cause problems if you are using a version control system.

Workaround: If you are not using the Nios II C2H Compiler, you can change the PTF file properties to read-only to prevent the IDE from changing the file.

"FATAL: can't create obj/<object filename>.o: Permission denied" error message [June 2006]

You might see the error message "FATAL: can't create obj/<object filename>.o: Permission denied" in the Nios II IDE if you attempt to rebuild a system library project while the C/C++ indexer is active.

Workaround: Ensure that the indexer is not active when you initiate a rebuild.

Incorrect address assignment for dual-port memory mastered by two different masters [June 2006]

If you have a dual-port memory in your Nios II system, and only the second slave port is mastered by the CPU, you might see an overlapping section error during the linking stage of building your software.

Workaround: In SOPC Builder, ensure that the first slave port of the dual-port memory is mastered by the Nios II CPU. The second port does not have to be mastered by the Nios II CPU.

Building Nios II Advanced project using Build Project invokes 'make clean all' by default

Nios II IDE Advanced projects created with earlier (pre-6.0) versions of the Nios II IDE call 'make clean all' when the project is built, rather than 'make all'. Also, on the Project menu, clicking Clean for these projects has no effect.

Workaround: To change the behavior of the IDE so that the project invokes make all by default for a build and make clean by default for a clean build, do the following:

1. Right click on the project and click **Properties**.
2. Click the C/C++ **Make Project** page.

3. In the **Workbench Build Behavior** section, change the value for **Rebuild (Full Build)** to **all**.
4. Turn on **Clean** and type `clean` in the corresponding text box.

Build errors after changing component names in SOPC Builder

If you rename components in the SOPC Builder system and then regenerate the SOPC Builder system, Nios II IDE system library projects based on that system will have build errors.

Workaround: After regenerating the SOPC Builder system, create a new system library project for the SOPC Builder system. Alternately, you can delete the system library project from the workspace without deleting the contents from the file system, and then re-import the project, selecting the appropriate SOPC Builder system.

Debugging Projects

"Step failed. Target is not responding (timed out)" error message [June 2006]

The Nios II IDE debugger might hang and report the above message if your code contains large arrays declared as local variables on the stack.

Workaround: Place the array and any other large buffers on the heap rather than on the stack.

Incorrect breakpoint filtering on threads

If you enable breakpoint filtering for a thread and later turn off filtering for the thread, the debugger might incorrectly continue to filter the thread.

Uninitialized Memory Error when executing from ISS

Under some cases the ISS does not ignore uninitialized memory reads, even when Uninitialized memory reads is set to Ignore on the ISS Settings tab of the run configuration.

Nios2-gdb-server fails to terminate after setting a watch point

You might be unable to terminate `nios2-gdb-server` after setting a watchpoint in the Nios II IDE debugger and resuming execution past the end of main. You will see an error "Terminate failed". You will not be able to start the debugger again; you will see a message reading "Another application is using the target processor..." in the Console view.

Workaround: Terminate the `nios2-gdb-server.exe` process manually using the Windows Task Manager.

Watchpoints do not work when set on variables whose size are not 32-bits

Workaround: Change the type of global and static local variables to `int`, `long`, or `unsigned long` before setting watchpoints on them.

Debugger cannot step into __sflags, and continues execution instead

The Nios II IDE debugger is unable to step into some low-level C library functions, such as `__sflags()` which is called from `_fopen_r()`. (`_fopen_r()` is called from `fopen()`.) If you try to step into such a function, execution will proceed as if you had indicated the debugger should resume execution.

Workaround: Step over such functions. Or, if execution continues after trying to step in, click **Suspend** on the Run menu.

Missing traced load/store instruction and data in the Trace view

If the trace options **Include load addresses**, **Include store addresses** or **Include data values** are enabled during debug, the load and store address and data will not appear at the first breakpoint after starting debugging. They will appear at successive breakpoints.

Workaround: To see load or store addresses and data in the instruction trace prior to main, turn on **Break at alt_main()** located on the **Debugger** tab for your debug configuration.

Cannot use watchpoints in the Nios II IDE when the FS2 console is open

Watchpoints do not work in the Nios II IDE when the **Use FS2 console window for trace and watchpoint support** setting is turned on in the **Debugger** tab of the Debug configuration. You will see an error message "The execution of program is suspended because of error." with details indicating that hardware watchpoints could not be inserted and deleted.

Workaround: If the FS2 console is open, you must use it to control watchpoints. For details, see the FS2 documentation.

Breakpoints on adjacent lines of assembly fail to halt the processor

Setting breakpoints on adjacent lines of assembly code might cause the Nios II processor to stop responding to the debugger.

Workaround: When debugging in mixed mode or disassembly view, separate breakpoints by at least one assembly instruction. This issue does not affect Nios II cores that do not have hardware breakpoints enabled in the JTAG debug module.

Navigating Projects

Nios II IDE might not prompt for workspace at startup

The Nios II IDE might not prompt for the workspace at startup.

Workaround: Specify the workspace at the command line using the `-data` argument. For example:

```
nios2-ide -data <path to workspace>
```

If the workspace does not exist, it will be created.

Nios II IDE stops responding after you double-click a .o file.

Double-clicking a `.o` file in the Nios II IDE on Windows might cause the IDE to stop responding. This issue occurs only on systems with a separate installation of Cygwin in addition to Cygwin installed with the Nios II development tools.

Workaround: Ensure that only the `cygwin1.dll` for the Nios II tools is in the PATH when launching Nios II IDE.

Error importing a software project created in Nios II IDE version 5.1 into Nios II IDE version 6.0

If you create a software project in Nios II IDE version 5.1 and then install Nios II version 6.0 and try to import that same project into your workspace, you might get an error "Problem deleting folder Debug".

Workaround: If you get this error, close the Nios II IDE and any other application which has files open from the project in question, then manually delete the Debug and/or Release directories from the file system. Re-import the project into the Nios II IDE version 6.0

Resource(s) out of sync with the file system when searching for files in the workspace

When searching through files in the IDE workspace, you might get an error message saying that one or more resources are out of sync with the file system.

Workaround: Right click in the Navigator view and click **Refresh**, and then perform the search again.

C/C++ Scanner does not understand certain C/C++ constructs

The C/C++ scanner is used for C/C++ Search, navigation, open declaration and parts of content assist. Due to limitations of the C/C++ Scanner, these features will not work with the following code constructs:

- Kernighan & Ritchie-style C
- Functions that take a function-pointer as an argument

Workaround: If the C/C++ Search fails, use the File Search facility.

C2H Compiler

This section lists any issues related to the Nios II C-to-Hardware Acceleration (C2H) Compiler.

Accelerator returns -1 if C/C++ project doesn't rebuild before running

A call to an accelerated function might return the invalid result -1 if you do not build your C application project before running it on hardware. The Nios II IDE does not detect that a new FPGA configuration file (`.sof`) needs to be downloaded if the system library is not rebuilt, which might result in attempting to execute an accelerated function on hardware that does not contain the hardware accelerator.

Workaround: Build your C/C++ application project before running it on hardware.

Incorrect results from logical or conditional operation with side-effects

The C2H Compiler always evaluates both operands of logical (&&, ||) and conditional (?:) operators. This is different from expected ANSI C behavior, for which operands are evaluated left-to-right, and unnecessary operands are skipped. For example, in the expression `(i-- && j--)`, if the value of `i` is zero, the right-hand-side (RHS) expression should not evaluate (i.e., `j` should not be decremented). However, this C2H Compiler erroneously evaluates both sides unconditionally, causing `j` to be decremented. The following example expressions could suffer from the same issue: `(i-- || j--)`, `(cond ? i-- : j--)`

Workaround: Use logical and conditional operations whose operators have no side effects. Side effects include pre-/post-fix increment operations (`++`, `--`), memory operations (`*`, `[]`, `.`, `->`), and function calls.

Closed system library while working with the C2H Compiler

The C2H Compiler requires the system library to obtain important details about the system, and cannot function if the system library is closed.

Workaround: Ensure that the system library project in the Nios II IDE is open prior to building an application project that contains a hardware accelerator.

Launch SOPC Builder button in C2H view

When the Nios II IDE workspace contains multiple projects with multiple system libraries, the incorrect SOPC Builder system might open when you click the Launch SOPC Builder in the C2H view.

Workaround: Keep only one system library project open at a time while using the C2H Compiler.

Build clean causes build failure

Performing a clean build on a Nios II IDE project that contains a hardware accelerator can cause the next build to fail in the IDE, because the clean build erroneously deletes a file required by the C2H Compiler.

Workaround: Do not perform a clean build on projects that use hardware accelerators. If you have already performed a clean build, recompile with option **Build software, generate SOPC Builder system, and run Quartus II compilation** to regenerate the necessary files.

Multiple clock domains causes hardware accelerator to fail

If a hardware accelerator and the components connected to its master ports are in different clock domains, the accelerator might behave incorrectly.

Workaround: Assign a single clock to a hardware accelerator and all the slave ports it connects to. It is acceptable for the system to contain multiple clock domains.

Comma operators are not supported

The C2H Compiler does not support comma operators, such as the following example:

```
for(i = 0, j = 3; i < 10; i++, j++)
{
```

```

    /* statements */
}

```

Workaround: You can manually duplicate the same functionality, such as:

```

j = 3;
for(i = 0; i < 10; i++)
{
    /* statements */
    j++;
}

```

Array elements of structs do not copy correctly

C2H hardware accelerators do not correctly copy array elements of structs. For example:

```

typedef struct my_struct {
    int a;
    int b;
    int buf[BUF_SIZE];
}MY_STRUCT;

MY_STRUCT struct_a = {1, 2, {3, 3, 3, 3}};
MY_STRUCT struct_b = {9, 8, {7, 7, 7, 7}};

struct_a = struct_b;

```

In this example, the `a` and `b` elements of the struct will copy correctly, but the `buf` element will not. After this assignment, `struct_a` will equal `{9, 8, {3, 3, 3, 3}}`.

Workaround: Copy the array elements explicitly, as follows:

```

{
    int i=0;
    do
    {
        struct_a.buf[i] = struct_b.buf[i];
        i++;
    } while (i<LENGTH_OF_BUF_ELEMENT)
}

```

Simulating on ISS is not supported

The Nios II instruction set simulator (ISS) does not support custom SOPC Builder components, and therefore cannot simulate systems that use hardware accelerators. You might get the following internal error if attempting to simulate using the ISS:

```

Internal Error (unhandled exception) in file cosim_main.cpp

```

Workaround: Run the program on a hardware system that includes the hardware accelerator.

Hardware accelerators remain after deleting the software project

If a system contains C2H hardware accelerators, deleting the software project that defines the accelerators does not remove the accelerators from the hardware system, and the accelerator logic remains in the SOPC Builder system.

Workaround: To remove an accelerator from a system, delete the accelerator from the C2H view in the Nios II IDE first, and then recompile the software project. The C2H Compiler then removes the accelerator from the

SOPC Builder system. Once the compilation is complete then the software application can be deleted from the workspace.

Changing build configurations produces unexpected results

The C2H Compiler does not support multiple build configurations (e.g. Release or Debug) in the Nios II IDE. After creating one or more accelerators in a particular configuration, the C2H Compiler will produce undefined results if you switch to a different build configurations and create more accelerators.

Workaround: For a specific SOPC Builder system and Nios II IDE project, specify C2H accelerators in only one build configuration. Note that you can still use multiple build configurations, as long as only one configuration specifies C2H Compiler settings.

Flash Programmer

This section lists any issues relating to the Nios II IDE.

elf2flash elf size limit

The elf2flash utility supports .elf files up to approximately 24 MBytes in size. The elf2flash utility might fail with error "java.lang.OutOfMemoryError" on files larger than 24 MBytes.

Workaround: You can either lower the number of symbols in your elf file by turning off debug symbols, or specify less initialized data in the application.

Download Cables & Debug Hardware

This section lists any issues related to download cables and other debug hardware.

Communication errors during run/debug sessions using older download cables

Debugging with the following Altera download cables might fail, due to electrical noise-related JTAG communication failures: USB-Blaster™ Rev A, ByteBlaster™, ByteBlasterMV™, ByteBlaster II, and MasterBlaster™ cables.

Currently, the only fully supported cable for downloading, debugging, or communicating with Nios II systems is the USB-Blaster Rev B cable or later. Revision B cables are clearly labeled as Revision B. (Revision A cables have no revision label.)

Workaround: Use a USB-Blaster Rev B cable. Older cables can be used, but they might encounter JTAG communication failures.

Development Boards

This section lists any issues related to Altera development boards.

Intermittent failures while accessing CompactFlash card

The Nios II Development Kit version 5.0 and higher includes a CompactFlash controller peripheral suitable for interfacing to CompactFlash cards in True IDE mode on Nios development boards. In order for True IDE mode to operate, CompactFlash cards require that the ATASEL_N input be driven to ground during power-up.

The CompactFlash controller peripheral includes a configurable power register used to power-cycle CompactFlash cards in Nios II software through a MOSFET on the Nios development boards. However, in certain development boards, power to the CompactFlash card will not turn off completely during this power cycle operation. Because of this, the CompactFlash might not sample the ATASEL_N pin during the power-cycle operation after FPGA configuration when this pin is driven to ground. Instead, the CompactFlash card might sample the ATASEL_N pin when power is first applied to the development board, when I/O are not yet driven by the FPGA (before FPGA configuration).

Workaround: If you encounter errors with CompactFlash when using the Nios development boards, try one of the following:

- Use a different CompactFlash card. Certain cards are more susceptible to the power-cycling issue than others.
- Modify the Nios development board. This is recommended for users who are familiar and comfortable with board-level modifications. Disconnect pin 9 (ATASEL_N) on the CompactFlash socket on your Nios development board and tie this pin to ground. Note that the CompactFlash socket uses a staggered numbering on the pins (starting from pin 1: 1, 26, 2, 27, ...); refer to the CompactFlash Association specification for right-angle surface-mount connectors for exact specifications on this connector. This modification will permanently enable True-IDE mode operation.

Toolchain (gcc, gdb, etc.)

This section lists any issues related to the Nios II compiler toolchain, such as gcc and gdb.

Breakpoints in C++ constructors fail to halt the processor

Breakpoints set in a C++ constructor might not halt the processor due to a widespread GNU GCC, GDB issue. This is not a Nios II IDE-specific issue.

Workaround: You can work around this issue by moving all of your constructor source code into another class method, called `init`. Then invoke this method from within the constructor.

Target Software

This section lists any issues related to software or drivers that target the Nios II processor.

malloc(), realloc() failures with MicroC/OS-II

When using the MicroC/OS-II RTOS, calls to `malloc()` and `realloc()` might fail if successive calls to `malloc()` or `realloc()` within a MicroC/OS-II task occur after changing the task priority of the task in which a memory block was originally allocated.

Workarounds:

- Allocate and/or reallocate memory blocks outside of MicroC/OS-II tasks, before task switching starts. Changing thread priorities at runtime is now possible.
- Allocate fixed areas of memory using arrays (rather than using `malloc()`) before task switching starts. Changing thread priorities at runtime is now possible.
- Allocate memory using `malloc()` or `realloc()` from a MicroC/OS-II task. You may change task priorities at runtime, but only for tasks that have not used `malloc()` or `realloc()`.

cout from MicroC-OS/II task will not send data to STDOUT

If neither `printf()` or `cout` is used from `main()` before tasks are started, `cout` will not work from a task.

Workaround: Add the following C++ code to the beginning of `main()`:

```
std::ios_base::sync_with_stdio(false);
```

Problems using HAL drivers with Toshiba Flash

The HAL CFI Flash driver might not work for Toshiba flash memory that claims to be CFI compliant.

Workaround: In the `altera_avalon_cfi_flash_table.c` file, change the `#define READ_ARRAY_MODE` from `(alt_u8)0xFF` to `(alt_u8)0xF0` and rebuild the project.

Creating new custom HAL components

When you first create a component's inc directory or HAL header file, you might first need to perform a clean build (i.e., rebuild) of existing system library projects for the new files to be detected.

Legacy SDK

Support for the Legacy SDK mode is removed in version 6.0 of the Nios II Embedded Design Suite.

SOPC Builder and Quartus II Software

This section lists any issues related to the Quartus II software or SOPC Builder that specifically affect Nios II designers.



For further information on the Quartus II software, refer to the latest Quartus II release notes on the Altera web site:

<http://www.altera.com/literature/lit-qts.jsp>

Cannot connect Nios II tightly-coupled instruction and data masters to the same dual-port memory

SOPC Builder does not generate an error if you connect Nios II tightly-coupled instruction and data masters to both ports on a dual-port on-chip memory. However, this configuration is not supported in hardware.

Workaround: Do not connect Nios II tightly-coupled masters to both ports of a dual-port memory.

Example Designs

This section lists any issues related to the example designs included with the Nios II Embedded Design Suite.

Hardware Designs

There are no known issues at this time.

Software Designs

RAM test failure when running Memory Test software template on the ISS

An issue in the instruction set simulator (ISS) model of the JTAG UART can cause a console communication error during the RAM test when running the Memory Test software template on the ISS.

Networking Examples

If you are running a networking example design and you are asked for a 9-digit number after the letters 'ASJ', and your Nios II development board does not have a sticker with a 9-digit number after the letters 'ASJ', please enter a unique 9-digit number when prompted. Ensure that this number is unique to each Nios board connected to your network to avoid network address conflicts.

Hardware Simulation

This section lists issues related to simulating Nios II processor systems on an RTL simulator, such as the ModelSim® simulator.

Simulation failure if reset address is set to EPCS

Running ModelSim RTL simulation of a Nios II system fails if the reset address of the Nios II processor is set to an EPCS Serial Flash Controller.

Workaround: To simulate your system, temporarily set the Reset Address of the Nios II CPU to the memory that your application code will reside (for example, SDRAM), then re-generate the system in SOPC Builder and run RTL simulation again. Before booting the Nios II CPU from EPCS flash on your target board, change the Nios II Reset Address back to the EPCS Controller peripheral and re-generate the system in SOPC Builder and re-compile in the Quartus II software to produce an updated FPGA configuration file with the Nios II CPU booting from EPCS flash.

Uninitialized BSS variables in simulation

If your program reads the value of an uninitialized BSS variable during HDL simulation when the HAL system library has been compiled with the **ModelSim only, no hardware support** property enabled in Nios II IDE, a warning will be produced about unfiltered data being 'x'. This occurs because when this property is enabled, the code that clears the BSS memory region is omitted to speed up HDL simulation so this memory region is

uninitialized. The BSS region contains global and static local variables that are not initialized by the application so they default to a value of zero. When the Nios II CPU reads uninitialized variables, it displays a warning and converts any of the bits of the uninitialized data to zero which correctly mimics the effect of the missing BSS clearing code. The HAL code that executes before and after main() may use BSS variables so these warnings might be generated even if your application doesn't use the BSS.

ModelSim fails to load large memory models

The ModelSim tool might fail to load simulation models for memory arrays larger than 128M bytes, halfwords or words in size. If the sum of the following parameters is greater than 27, the ModelSim tool will fail to load:

- address_bits (i.e. 14)
- column bits (i.e. 11)
- $\log_2(\text{number of banks})$ (number of banks is usually 4, so this term is usually 2)
- $\log_2(\text{chipselects})$ (number of chipselects is usually 1, so this term is usually 0)

Workaround: Simulate using a smaller SDRAM than the SDRAM implemented in hardware. This is possible if the entire memory space doesn't need to be simulated.

Documentation Issues

This section lists errors, unclear descriptions, or omissions from current published specifications or product documents.

EPCS Controller memory usage

The chapter *EPCS Device Controller Core with Avalon Interface* in the *Quartus II Handbook volume 5: Embedded Peripherals* utilizes 1Kbyte on-chip memory for its boot-loader program. However, when targeting Stratix II devices, the on-chip memory required for the boot-loader increases and requires two M4K blocks.

Error in UART Core with Avalon Interface Chapter

The chapter *UART Core with Avalon Interface* in the *Quartus II Handbook volume 5: Embedded Peripherals* incorrectly states: "When parity is Even, the parity bit is 1 if the character has an even number of 1 bits; otherwise the parity bit is 0. Similarly, when parity is Odd, the parity bit is 1 if the character has an odd number of 1 bits."

This should read: "When parity is Even, the parity bit is 1 if the number of 1's in the character plus the parity bit is even; otherwise the parity bit is 0. Or in other words, the Parity bit is set to '0' when there is an even number of '1' bits in the character. Similarly, when parity is Odd, the parity bit is 1 if the number of 1's in the character plus the parity bit has an odd number of 1 bits. Or in other words, the Parity bit is set to '0' when there is an odd number of '1' bits in the character."

Nios II IDE help system fails to display content

To display the help content, your computer must be able to recognize itself on the network. Incorrect proxy settings can cause the help content to fail to display.

Workaround: Specify valid browser proxy settings.

byteenables in Avalon Interface Specification

The *Avalon Interface Specification* incorrectly describes the behavior for `byteenable` during burst transfers. The *Avalon Interface Specification States*: "The master port can assert or deassert the `byteenable` lines as necessary for each individual transfer within the burst." During Avalon master transfers, the master port must assert all `byteenable` for all transfers in the burst.

Workaround: Assert all `byteenable` lines during master burst transfers.

burstcount in Avalon Interface Specification

The *Avalon Interface Specification* incorrectly describes the behavior for `address` and `burstcount` during burst transfers. The *Avalon Interface Specification states*: "The start of a write burst is similar to the start of a fundamental master write transfer. The master port asserts `address`, `writedata`, `write`, and `byteenable` (if present) in addition to `burstcount`. ... This is the only time that the Avalon switch fabric captures `burstcount` and `address`; the master port can deassert them through the remainder of the burst." However, during Avalon master transfers, the master port must assert constant values on `address` and `burstcount` for the duration of the burst.

Workaround: During Avalon master transfers, assert constant values on `address` and `burstcount` for the duration of the burst.

Contact Information

For more information, contact Altera's mySupport website at www.altera.com/mysupport. Click **Create New Service Request**, and choose the **Product Related Request** form.

Revision History

Table 1 shows the revision history for the Nios II Embedded Design Suite v6.0 Errata Sheet.

Table 1. Nios II Embedded Design Suite v6.0 Errata Sheet Revision History

Version	Date	Errata Summary
1.0	May 2006	First release
1.1	June 2006	Added new items: <ul style="list-style-type: none"> • "nios_cpu: Unknown Break Location nios_cpu/jtag_debug_module" error message • VHDL sensitivity list warnings during Quartus II synthesis • Nios II IDE unnecessarily updates the SOPC Builder system file (.ptf) • "FATAL: can't create obj/<object filename>.o: Permission denied" error message • Incorrect address assignment for dual-port memory mastered by two different masters • "Step failed. Target is not responding (timed out)" error message



101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
www.altera.com
Applications Hotline:
(800) 800-EPLD
Literature Services:
literature@altera.com

© 2006 Altera Corporation. All rights reserved. Altera, The Programmable Solutions Company, the stylized Altera logo, specific device designations, and all other words and logos that are identified as trademarks and/or service marks are, unless noted otherwise, the trademarks and service marks of Altera Corporation in the U.S. and other countries. All other product or service names are the property of their respective holders. Altera products are protected under numerous U.S. and foreign patents and pending applications, maskwork rights, and copyrights. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

