

Introduction

Each release of the Nios® II Embedded Design Suite (EDS) introduces improvements to the Nios II processor, the software development tools, or both. This document catalogs the history of revisions to the Nios II processor; it does not track revisions to development tools, such as the Nios II integrated development environment (IDE). This chapter contains the following sections:

- “Nios II Versions” on page 6-1
- “Architecture Revisions” on page 6-2
- “Core Revisions” on page 6-3
- “JTAG Debug Module Revisions” on page 6-6

Improvements to the Nios II processor might affect:

- Features of the Nios II architecture—An example of an architecture revision is adding instructions to support floating-point arithmetic.
- Implementation of a specific Nios II core—An example of a core revision is increasing the maximum possible size of the data cache memory for the Nios II/f core.
- Features of the JTAG debug module—An example of a JTAG debug module revision is adding an additional trigger input to the JTAG debug module, allowing it to halt processor execution on a new type of trigger event.

Altera implements Nios II revisions such that code written for an existing Nios II core also works on future revisions of the same core.

Nios II Versions

The number for any version of the Nios II processor is determined by the version of the Nios II EDS. For example, in the Nios II EDS version 8.0, all Nios II cores are also version 8.0.

Table 6-1 lists the version numbers of all releases of the Nios II processor.

Table 6-1. Nios II Processor Revision History (Part 1 of 2)

Version	Release Date	Notes
8.0	May 2008	<ul style="list-style-type: none"> ■ Added an optional memory management unit (MMU). ■ Added an optional memory protection unit (MPU). ■ Added advanced exception checking. ■ Added the <code>initda</code> instruction.
7.2	October 2007	Added the <code>jmp<i>i</i></code> instruction.
7.1	May 2007	No changes.
7.0	March 2007	No changes.

Table 6-1. Nios II Processor Revision History (Part 2 of 2)

Version	Release Date	Notes
6.1	November 2006	No changes.
6.0	May 2006	The name Nios II Development Kit describing the software development tools changed to Nios II Embedded Design Suite.
5.1 SP1	January 2006	Bug fix for Nios II/f core.
5.1	October 2005	No changes.
5.0	May 2005	<ul style="list-style-type: none"> ■ Changed version nomenclature. Altera® now aligns the Nios II processor version with Altera's Quartus II® software version. ■ Memory structure enhancements: <ol style="list-style-type: none"> (1) Added tightly-coupled memory. (2) Made data cache line size configurable. (3) Made cache optional in Nios II/f and Nios II/s cores. ■ Support for HardCopy® devices.
1.1	December 2004	<ul style="list-style-type: none"> ■ Minor enhancements to the architecture: Added <code>cpuid</code> control register, and updated the <code>break</code> instruction. ■ Increased user control of multiply and shift hardware in the arithmetic logic unit (ALU) for Nios II/s and Nios II/f cores. ■ Minor bug fixes.
1.01	September 2004	■ Minor bug fixes.
1.0	May 2004	Initial release of the Nios II processor.

Architecture Revisions

Architecture revisions augment the fundamental capabilities of the Nios II architecture, and affect all Nios II cores. A change in the architecture mandates a revision to all Nios II cores to accommodate the new architectural enhancement. For example, when Altera adds a new instruction to the instruction set, Altera consequently must update all Nios II cores to recognize the new instruction. [Table 6-2](#) lists revisions to the Nios II architecture.

Table 6-2. Nios II Architecture Revisions (Part 1 of 2)

Version	Release Date	Notes
8.0	May 2008	<ul style="list-style-type: none"> ■ Added an optional MMU. ■ Added an optional MPU. ■ Added advanced exception checking to detect division errors, illegal instructions, misaligned memory accesses, and provide extra exception information. ■ Added the <code>initda</code> instruction.
7.2	October 2007	Added the <code>jmp</code> instruction.
7.1	May 2007	No changes.
7.0	March 2007	No changes.
6.1	November 2006	No changes.
6.0	May 2006	Added optional <code>cpu_resetrequest</code> and <code>cpu_resettaken</code> signals to all processor cores.

Table 6-2. Nios II Architecture Revisions (Part 2 of 2)

Version	Release Date	Notes
5.1	October 2005	No changes.
5.0	May 2005	Added the <code>flushda</code> instruction.
1.1	December 2004	<ul style="list-style-type: none"> ■ Added <code>cpuid</code> control register. ■ Updated <code>break</code> instruction specification to accept an immediate argument for use by debugging tools.
1.01	September 2004	No changes.
1.0	May 2004	Initial release of the Nios II processor architecture.

Core Revisions

Core revisions introduce changes to an existing Nios II core. Core revisions most commonly fix identified bugs, or add support for an architecture revision. Not every Nios II core is revised with every release of the Nios II architecture.

Nios II/f Core

Table 6-3 lists revisions to the Nios II/f core.

Table 6-3. Nios II/f Core Revisions (Part 1 of 2)

Version	Release Date	Notes
8.0	May 2008	<ul style="list-style-type: none"> ■ Implemented the optional MMU. ■ Implemented the optional MPU. ■ Implemented advanced exception checking. ■ Implemented the <code>initda</code> instruction.
7.2	October 2007	Implemented the <code>jmp_i</code> instruction.
7.1	May 2007	No changes.
7.0	March 2007	No changes.
6.1	November 2006	No changes.
6.0	May 2006	Cycle count for <code>flushi</code> and <code>initi</code> instructions changes from 1 to 4 cycles.
5.1 SP1	January 2006	<p>Bug Fix:</p> <p>Back-to-back store instructions can cause memory corruption to the stored data. If the first store is not to the last word of a cache line and the second store is to the last word of the line, memory corruption occurs.</p>
5.1	October 2005	No changes.

Table 6-3. Nios II/f Core Revisions (Part 2 of 2)

Version	Release Date	Notes
5.0	May 2005	<ul style="list-style-type: none"> ■ Added optional tightly-coupled memory ports. Designers can add zero to four tightly-coupled instruction master ports, and zero to four tightly-coupled data master ports. ■ Made the data cache line size configurable. Designers can configure the data cache with the following line sizes: 4, 16, or 32 bytes. Previously, the data cache line size was fixed at 4 bytes. ■ Made instruction and data caches optional (previously, cache memories were always present). If the instruction cache is not present, the Nios II core does not have an instruction master port, and must use a tightly-coupled instruction memory. ■ Full support for HardCopy devices (previous versions required a work around to support HardCopy devices).
1.1	December 2004	<ul style="list-style-type: none"> ■ Added user-configurable options affecting multiply and shift operations. Now designers can choose one of three options: <ol style="list-style-type: none"> (1) Use embedded multiplier resources available in the target device family (previously available). (2) Use logic elements to implement multiply and shift hardware (new option). (3) Omit multiply hardware. Shift operations take one cycle per bit shifted; multiply operations are emulated in software (new option). ■ Added <code>cpuid</code> control register. ■ Bug Fix: Interrupts that were disabled by <code>wrctl ienable</code> remained enabled for one clock cycle following the <code>wrctl</code> instruction. Now the instruction following such a <code>wrctl</code> cannot be interrupted.
1.01	September 2004	<ul style="list-style-type: none"> ■ Bug Fixes: <ol style="list-style-type: none"> (1) When a store to memory is followed immediately in the pipeline by a load from the same memory location, and the memory location is held in the data cache, the load may return invalid data. This situation can occur in C code compiled with optimization off (-O0). (2) The SOPC Builder top-level system module included an extra, unnecessary output port for systems with very small address spaces.
1.0	May 2004	Initial release of the Nios II/f core.

Nios II/s Core

Table 6-4 lists revisions to the Nios II/s core.

Table 6-4. Nios II/s Core Revisions (Part 1 of 2)

Version	Release Date	Notes
8.0	May 2008	Implemented the illegal instruction exception.
7.2	October 2007	Implemented the <code>jmp</code> instruction.
7.1	May 2007	No changes.
7.0	March 2007	No changes.
6.1	November 2006	No changes.
6.0	May 2006	Cycle count for <code>flushi</code> and <code>initi</code> instructions changes from 1 to 4 cycles.

Table 6-4. Nios II/s Core Revisions (Part 2 of 2)

Version	Release Date	Notes
5.1	October 2005	No changes.
5.0	May 2005	<ul style="list-style-type: none"> ■ Added optional tightly-coupled memory ports. Designers can add zero to four tightly-coupled instruction master ports. ■ Made instruction cache optional (previously instruction cache was always present). If the instruction cache is not present, the Nios II core does not have an instruction master port, and must use a tightly-coupled instruction memory. ■ Full support for HardCopy devices (previous versions required a work around to support HardCopy devices).
1.1	December 2004	<ul style="list-style-type: none"> ■ Added user-configurable options affecting multiply and shift operations. Now designers can choose one of three options: <ol style="list-style-type: none"> (1) Use embedded multiplier resources available in the target device family (previously available). (2) Use logic elements to implement multiply and shift hardware (new option). (3) Omit multiply hardware. Shift operations take one cycle per bit shifted; multiply operations are emulated in software (new option). ■ Added user-configurable option to include divide hardware in the ALU. Previously this option was available for only the Nios II/f core. ■ Added <code>cpuid</code> control register.
1.01	September 2004	<p>Bug fix:</p> <p>The SOPC Builder top-level system module included an extra, unnecessary output port for systems with very small address spaces.</p>
1.0	May 2004	Initial release of the Nios II/s core.

Nios II/e Core

Table 6-5 lists revisions to the Nios II/e core.

Table 6-5. Nios II/e Core Revisions

Version	Release Date	Notes
8.0	May 2008	Implemented the illegal instruction exception.
7.2	October 2007	Implemented the <code>jmp i</code> instruction.
7.1	May 2007	No changes.
7.0	March 2007	No changes.
6.1	November 2006	No changes.
6.0	May 2006	No changes.
5.1	October 2005	No changes.
5.0	May 2005	Full support for HardCopy devices (previous versions required a work around to support HardCopy devices).
1.1	December 2004	Added <code>cpuid</code> control register.
1.01	September 2004	<p>Bug fix:</p> <p>The SOPC Builder top-level system module included an extra, unnecessary output port for systems with very small address spaces.</p>
1.0	May 2004	Initial release of the Nios II/e core.

JTAG Debug Module Revisions

JTAG debug module revisions augment the debug capabilities of the Nios II processor, or fix bugs isolated within the JTAG debug module logic.

Table 6–6 lists revisions to the JTAG debug module.

Table 6–6. JTAG Debug Module Revisions

Version	Release Date	Notes
8.0	May 2008	No changes.
7.2	October 2007	No changes.
7.1	May 2007	No changes.
7.0	March 2007	No changes.
6.1	November 2006	No changes.
6.0	May 2006	No changes.
5.1	October 2005	No changes.
5.0	May 2005	Full support for HardCopy devices (previous versions of the JTAG debug module did not support HardCopy devices).
1.1	December 2004	Bug fix: When using the Nios II/s and Nios II/f cores, hardware breakpoints may have falsely triggered when placed on the instruction sequentially following a <code>jmp</code> , <code>trap</code> , or any branch instruction.
1.01	September 2004	<ul style="list-style-type: none"> ■ Feature enhancements: <ol style="list-style-type: none"> (1) Added the ability to trigger based on the instruction address. Uses include triggering trace control (trace on/off), sequential triggers, and trigger in/out signal generation. (2) Enhanced trace collection such that collection can be stopped when the trace buffer is full without halting the Nios II processor. (3) Armed triggers – Enhanced trigger logic to support two levels of triggers, or "armed triggers"; enabling the use of "Event A then event B" trigger definitions. ■ Bug fixes: <ol style="list-style-type: none"> (1) On the Nios II/s core, trace data sometimes recorded incorrect addresses during interrupt processing. (2) Under certain circumstances, captured trace data appeared to start earlier or later than the desired trigger location. (3) During debugging, the processor would hang if a hardware breakpoint and an interrupt occurred simultaneously.
1.0	May 2004	Initial release of the JTAG debug module.

Referenced Documents

This chapter references no other documents.

Document Revision History

Table 6–7 shows the revision history for this document.

Table 6-7. Document Revision History

Date & Document Version	Changes Made	Summary of Changes
November 2008 v8.1.0	Maintenance release.	—
May 2008 v8.0.0	<ul style="list-style-type: none"> ■ Added MMU information. ■ Added MPU information. ■ Added advanced exception checking information. ■ Added <code>initda</code> instruction information. 	Added MMU, MPU, advanced exception checking, and <code>initda</code> instruction.
October 2007 v7.2.0	<ul style="list-style-type: none"> ■ Added <code>jmp</code> instruction information. ■ Added exception handling information. 	Added <code>jmp</code> instruction
May 2007 v7.1.0	<ul style="list-style-type: none"> ■ Updated tables to reflect no changes to cores. ■ Added table of contents to Introduction section. ■ Added Referenced Documents section. 	—
March 2007 v7.0.0	Updated tables to reflect no changes to cores.	—
November 2006 v6.1.0	Updated tables to reflect no changes to cores.	—
May 2006 v6.0.0	Updates for Nios II cores version 6.0.	—
October 2005 v5.1.0	Updates for Nios II cores version 5.1.	—
May 2005 v5.0.0	Updates for Nios II cores version 5.0.	—
December 2004 v1.1	Updates for Nios II cores version 1.1.	—
September 2004 v1.0	Initial release.	—

