



This document guides users through the process of migrating an existing embedded system that uses the Altera® Nios® II Classic embedded processor to the Nios II Gen2 processor. This document discusses all the necessary hardware and software changes to use the Nios II Gen2 processor, as well as optional changes that can be made to further enhance system performance and functionality.

The Nios II embedded processor family and development kits have been adopted by many engineering teams worldwide in part because of their ease of use both in development and implementation of system-on-a-programmable-chip (SOPC) designs. The Nios II processor represents the next revolutionary step in embedded design. Nios II Gen2 processor cores are a binary compatible update to the Nios II Classic processor architecture. These new cores offer several new features over the existing cores with minor effort to port over from a Nios II Classic project. Upgrading a system from the Nios II Classic processor requires minor system changes that are covered in this document. These steps include replacing the Nios II Classic processor with a Nios II Gen2 processor in Qsys manually or by using a script.

## Prerequisites

You should ensure that the following tools are installed:

- Quartus® II 14.0 or higher
- Nios II Embedded Design Suite 14.0 or higher

For more information, refer to the "Instantiating the Nios II Gen2 Processor" chapter in the *Nios II Gen2 Processor Reference Guide* and the *Nios II Hardware Development Tutorial*.

### Related Information

- [Nios II Gen2 Processor Reference Guide](#)
- [Nios II Gen2 Hardware Development Tutorial](#)

## Nios II Gen2 Processor Feature Enhancements

The Nios II Gen2 processor family consists of /e and /f cores. It offers improvements over the Nios II Classic processor cores:

- Optional full 32-bit address space
- Optional user-defined Peripheral address region for data cache bypass
- Improved Qsys interface

The Nios II Gen2 /e core is completely backwards compatible with the Nios II Classic /s core. The Nios II Classic /s core has no direct equivalent in the Nios II Gen 2 family, however the Nios II Gen2 /f processor (as it has a more flexible configuration capability) can be configured to have the same feature set as the Nios II Classic /s core.

The Nios II Gen2 /f core offers the following feature enhancements over the Nios II Classic /f core:

- Optional full ECC support, including data cache and TCMs (Tightly-coupled Memories)
- Optional static branch prediction
- Higher performance multiplier
- Improved and more deterministic divider
- 64-bit multiply supported on all devices
- Improved low-cost shifter implementation up to 4 bits/cycle
- Instruction cache is now optional even when JTAG debug is present
- New system interface for system trace

For more detailed information on any of these features, refer to the *Nios II Gen2 Processor Reference Guide*.

#### Related Information

[Nios II Gen2 Processor Reference Guide](#)

## Hardware Migration from Nios II Classic to Nios II Gen2 Processors

The Qsys IP and system interface for the Nios II Classic and Nios II Gen2 processors are the same. This enables you to directly replace the Nios II Classic processor in your existing project with the new Nios II Gen2 IP with minimum effort. Qsys re-generation and project re-compilation is required after replacing the existing Nios II Classic IP.

Starting in the Altera Complete Design Suite (ACDS) 14.1, migration of Nios II Classic to Nios II Gen2 is automated. You will have a choice of upgrading your Nios II system when a Nios II Classic project is opened in Quartus 14.1 and above.

### Manual Update with Migration Tcl script

In ACDS 14.0, a migration script has been created that migrates hardware designs from Nios II Classic to Nios II Gen2. To migrate your existing Qsys design to the Nios II Gen2 processor, follow the steps below:

1. Download **nios2\_gen2\_migration.tcl** and put the script in your project directory
2. Run Nios II Command Shell in a Windows system or console in a Linux system.
3. Browse to your project directory and locate your system Qsys project file:

**<your\_qsys\_project name>.qsys**

4. Once in the directory run the following command:

```
qsys-script--script=nios2_gen2_migration.tcl--system-file=<your_qsys_project name>.qsys
```

5. Regenerate Qsys and recompile your project after applying the above Tcl script in Quartus II.
6. Open the Nios II SBT and run "generate bsp-editor" in Nios II EDS to update the BSP.

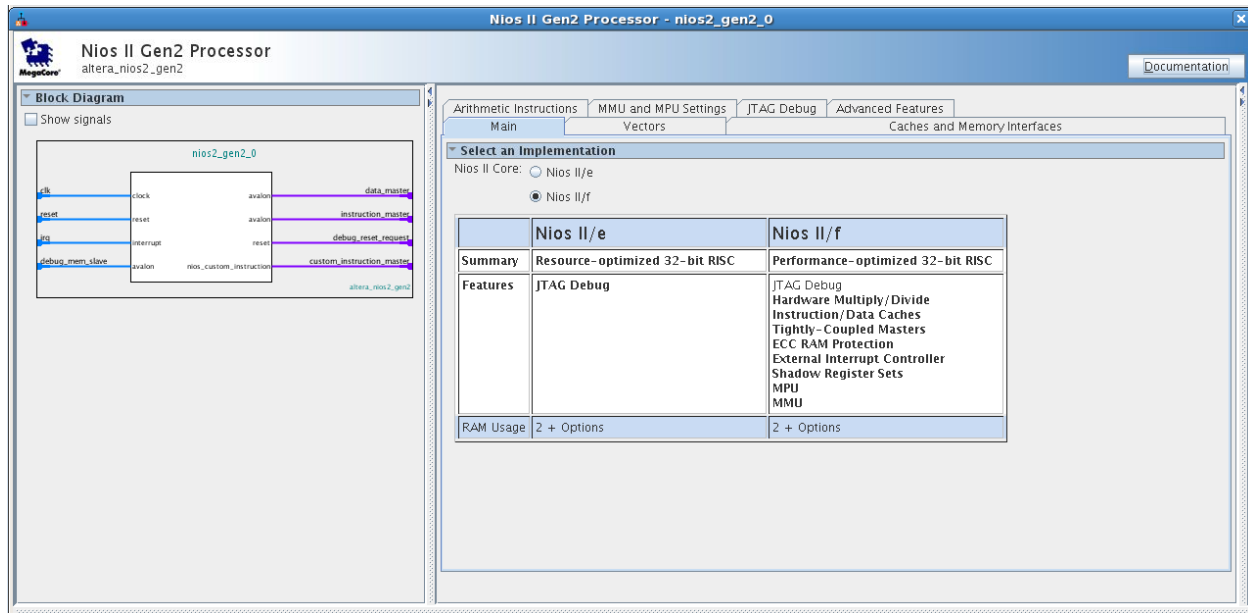
#### Related Information

[nios2\\_gen2\\_migration.tcl](#)

### Qys settings for migrate Nios II Classic /s core to Nios II Gen2 /f core without using Tcl script

Unlike Nios II Classic, the Nios II Gen2 family does not have a standard /s core. The steps below describe how to configure a Nios II Gen2 /f core to have an equivalent feature set to the Nios II Classic /s core.

Figure 1: Nios II Gen2 Qsys Interface



In Qsys instantiate a Nios II Gen2 processor and open the processor configuration window. Select the following settings in the tabs:

Main tab:

- Nios II Core: Select **Nios II /f**

Caches and Memory Interfaces tab:

- Instruction cache: Select the size required
- Data Cache: size = None

Advanced Features tab:

- Branch prediction = Static

## Software Migration to Nios II Gen2 Processors

Altera recommends that all BSP packages in the Nios II Software Build Tools projects are rebuilt after hardware migration (using the new **.sopcinfo** file) to ensure the correct hand off of information from the hardware.

**Note:** If you add new Nios II Gen2 features (peripheral memory regions or 32-bit addressing), ensure that the software has been modified to take this into account.

## Software Limitations - Uncached Memory Regions

When a processor with a data cache issues a read and the data is not in the cache, the cache will load a small block or 'line' of data from memory into the cache. When the processor issues a write, the new value is stored in the data cache. With a write back data cache as in the Nios II processor, new data values are only written back to the real physical location when the cache is flushed or data is evicted from the cache. This kind of operation is not acceptable for regions of memory where the data must be written

immediately to the hardware to affect the function/operation desired such as shared memory or writes to peripheral/hardware control registers. You are required to either flush the cache immediately, or use a memory operation that bypasses the cache and goes directly to the hardware preferred. Sometimes there may be other reasons why you don't want the data to be cached (e.g. if the data is never to be read back and/or it is not desirable to evict other data from the cache for performance or other reasons).

With the Nios II Classic processor cache bypass can be achieved by using special I/O read/write instructions (e.g. `ldwio/stwio`), by setting bit-31 of the data read/write address, or by implementing an MMU/MPU and setting it up appropriately.

**Note:** The MMU and bit-31 options are mutually exclusive. Nios II Gen2 cores can offer all the same mechanisms for cache bypass, but the caching implementation is slightly different and could potentially cause an issue with Nios II Classic to Nios II Gen2 software migration.

If the Nios II Classic processor performs an uncached write and the memory location (and hence associated cache line) is present in the data cache, the new data value is also written to the cache despite this being an uncached write. This means that the data in the cache is always up to date, and if the software design causes data from locations that have been subject to uncached memory writes to be loaded from the cache, there is no loss of data coherency. Hence the Nios II Classic processor can support uncached memory regions of any size and any address alignment as it allows mixing of cached and uncached regions of memory on the same cache line. This is not true for Nios II Gen2 cores.

In the Nios II Gen2 processor core, uncached transactions truly bypass the data cache and the cache contents are not updated (standard practice for most processors). If an uncached memory region shares a cache line with a cached region, the uncached region will be mapped into the cache and the memory may be updated with useless data when the cache is flushed. You have to ensure that uncached and cached memory regions are clearly separated and do not share the same cache line when mapped into the data cache. This is achieved by ensuring that uncached memory regions are aligned to the size of the data cache line (32 bytes) and are the size of one or more data cache lines (multiples of 32 bytes). This prevents mixing of cacheable and non-cacheable data on the same cache line and hence write back of stale/invalid data. Altera recommends that base addresses of peripheral register sets are aligned with the data cache line boundaries and that drivers use the cache bypass I/O read write instructions. So it is unlikely that modifications to drivers will be required. Other software that uses cache bypass/uncached memory regions should be checked carefully to ensure that cache lines do not map to regions containing both cached and uncached data.

You should consider the potential effects of using the new 32-bit address range and peripheral memory region options if the hardware configuration is changed to use them. The software should be modified to support these features or detect them and issue error messages.

**Note:** Software that uses the bit-31 cache bypass feature will no longer work if the Gen2 32-bit address range option is selected, the cache bypass I/O instructions or the peripheral memory region should be used instead.

The HAL and Linux driver code has all been updated so as long as a new BSP has been generated and compiled there will be no issues with the Nios II software libraries.

## Conclusion

Using this document, you can migrate a Nios II Classic embedded system design to the new Nios II Gen 2 processor with minimum effort.