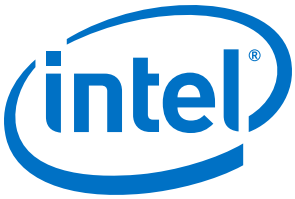


AN 891: Using the Reset Release Intel FPGA IP

Updated for Intel® Quartus® Prime Design Suite: **19.3**

AN-891 | 2019.09.30
Latest document on the web: **PDF** | **HTML**

Contents

1. AN 891: Using the Intel FPGA Reset Release FPGA IP

Related Information

AN 891: Using the Reset Release Intel FPGA IP Archives on page 10

1.1. Introduction

Intel® Stratix® 10 devices use a parallel, sector-based architecture that distributes the core fabric logic across multiple sectors. Device configuration proceeds in parallel with each Local Sector Manager (LSM) configuring its own sector. Consequently, FPGA registers and core logic are not released from reset at exactly the same time, as has always been the case in previous families.

The continual increases in clock frequency, device size, and design complexity now necessitate a well-thought out reset strategy that considers the possible effects of slight differences in the release from reset. This reset strategy must hold the device in reset until all registers and core logic are in user mode. Intel strongly recommends that you use the `nINIT_DONE` output of the Reset Release Intel FPGA IP as one of the initial inputs to your reset circuit.

1.2. Using the Reset Release Intel FPGA IP

The Reset Release Intel FPGA IP holds a control circuit in reset until the device has fully entered user mode. The FPGA asserts the `INIT_DONE` output to signal that the device is fully in user mode. The Reset Release FPGA IP generates an inverted version of the internal `INIT_DONE` signal, `nINIT_DONE` for use in your design.

The `nINIT_DONE` is high until the entire device enters user mode. After `nINIT_DONE` asserts (low), all logic is in user mode and operates normally. You can use the `nINIT_DONE` signal in one of the following ways:

- To gate an external or internal reset.
- To gate the reset input to the transceiver and I/O PLLs.
- To gate the write enable of design blocks such as embedded memory blocks, state machine, and shift registers.
- To synchronously drive register reset input ports in your design.

Attention: When you instantiate Reset Release Intel FPGA IP in your design, the Intel Quartus® Prime Fitter selects one Local Sector Manager (LSM) to output the `nINIT_DONE` signal. A Intel Quartus Prime Pro Edition legality check prevents you from instantiating more than on instance of the Reset Release Intel FPGA IP. Multiple instances would result in some skew between the `nINIT_DONE` signals.

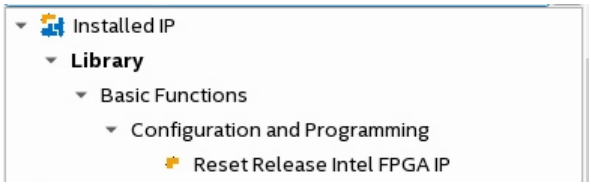
1.2.1. Instantiating the Reset Release IP In Your Design

The Reset Release IP is available in the IP Catalog in the **Basic Functions ➤ Configuration and Programming** category. This IP has no parameters.

Complete the following steps to instantiate the Reset Release IP in your design.

1. In the IP Catalog, type `reset release` in the search window to find the Reset Release Intel FPGA IP.

Figure 1. Locate Reset ReleaseIntel FPGA IP in IP Catalog



2. Double click the **Reset Release Intel FPGA IP** to add the Reset Release IP to your design.
3. In the **New IP Variant** dialog box, browse to your IP directory and specify a file name for the Reset Release IP. Then click **Create**.
   The Reset Release IP is now included in your project.

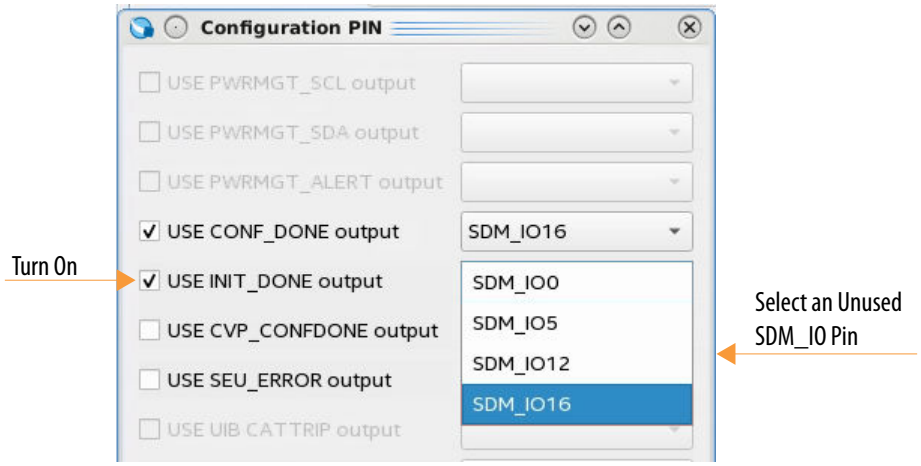
1.2.2. Assigning INIT_DONE To an SDM_IO Pin

If you choose to route `INIT_DONE` to an external pin, you must assign `INIT_DONE` to an SDM_IO pin.

Complete the following steps to make this assignment.

1. On the Intel Quartus Prime Assignments menu, select **Device ➤ Device and Pin Options ➤ Configuration Pin**, turn on the **Use INIT_DONE** output.
2. In the drop-down list, select any SDM_IO pin that is available.

Figure 2. Assigning INIT_DONE to SDM_IO Pin



Note: The Reset Release IP generates the nINIT_DONE internal signal whether or not you choose to assign INIT_DONE to an SDM_IO pin.

1.2.3. Gating an External Reset

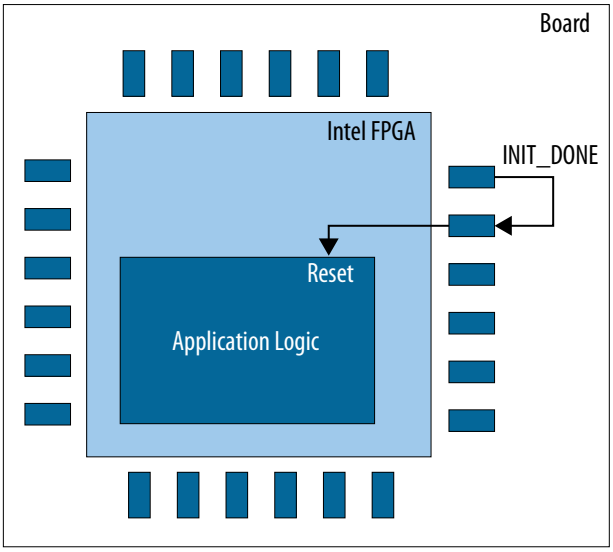
If your design is held in reset using an externally sourced reset signal, you should gate this external reset.

You have two options for gating an external reset:

- Use the INIT_DONE output signal to gate this external reset signal.
- Use nINIT_DONE from the Reset Release IP to gate the external reset source after this reset enters the device.

You can also feed the external INIT_DONE signal directly back into the FPGA through an external pin-to-pin connection in place of the Reset Release IP.

Figure 3. Using INIT_DONE To Hold Your Design in Reset

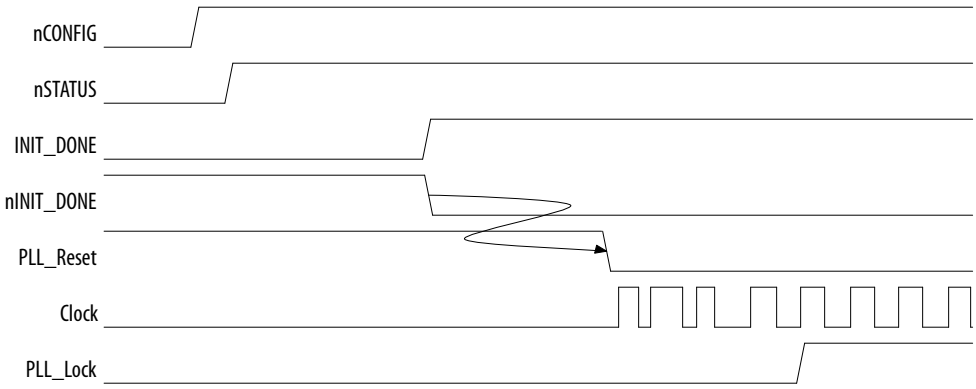


Note that any external signal may not be immediately effective in the fabric while the FPGA is initializing. Furthermore, I/O is frozen as logic 1 until activated. Consequently, Intel recommends using an active high external reset.

1.2.4. Gating the PLL Reset Signal

In older FPGA device families, designs frequently used the PLL lock signal to hold the custom FPGA logic in reset until the PLL locked. In newer Intel device families the lock time of PLLs can be less than the initialization time. In some cases the PLL may lock before the device completes initialization. Consequently, if you use the locked output of the PLL to control resets in Intel Stratix 10 or Intel Agilex™ devices, you should gate the PLL reset input with nINIT_DONE as shown the figure.

Figure 4. Using nINIT_DONE to Gate the PLL_Reset Signal



Another alternative if you are using `PLL_Lock` in your reset sequence is to gate the `PLL_Lock` output with the `nINIT_DONE` signal, `(PLL_Lock && !nINIT_DONE)`.

1.3. Guidance When Using Partial Reconfiguration (PR)

The PR Region Controller IP provides reset logic that ensures that the static region of the device and the PR personas do not interact during PR.

The Reset Release IP is only necessary to manage reset for full FPGA core configuration and subsequent full FPGA core configurations. The Reset Release IP is not necessary to prevent interaction between the static and PR personas during the PR process. For more information about PR refer to the *Intel Quartus Prime Pro Edition User Guide: Partial Reconfiguration*.

Related Information

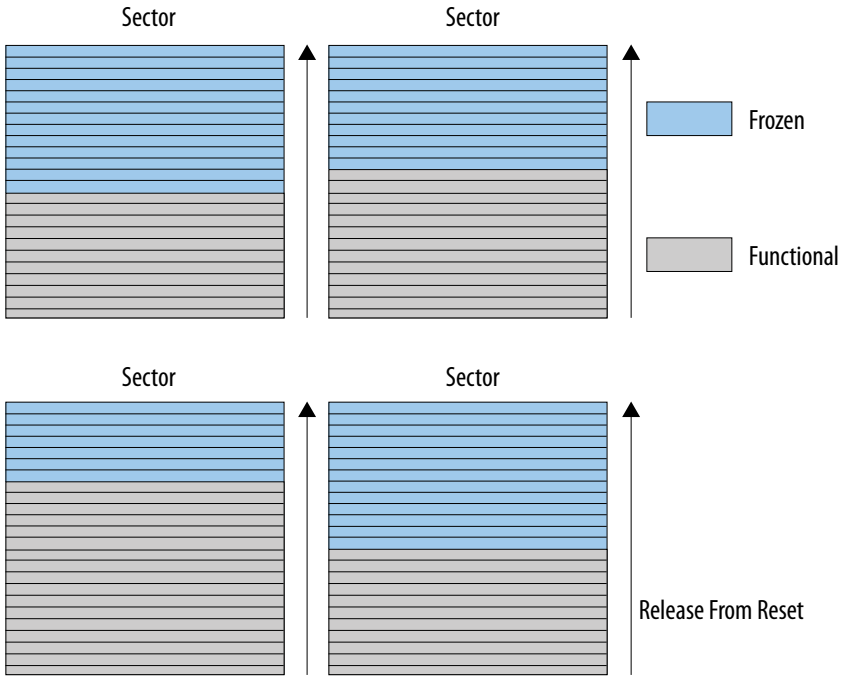
Creating a Partial Reconfiguration Design

1.4. Detailed Description of Device Configuration

Each Local Sector Manager (LSM) configures its own sector. A sector comprises multiple logic array block (LAB) rows. A logical function can span multiple rows and multiple sectors.

During configuration global configuration control signals hold the core fabric in a frozen state to prevent electrical contention. The LSMs work in parallel to asynchronously unfreeze the sectors. Within a sector the LSM unfreezes LAB rows and registers in the LABs sequentially. The LSMs work to unfreeze the fabric in parallel across all sectors without synchronization. Consequently, logic in different sectors or in the same sector but in different rows could begin to operate while other logic is still frozen. The `INIT_DONE` signal asserts when all the LSMs have entered user mode.

Figure 5. Releasing LAB Rows and Registers in the LABs Sequentially and Asynchronously Across Sectors



The following topics provide more detail about device configuration and initialization, and possible consequences if you do not use the Reset Release IP to hold the Intel Stratix 10 or Intel Agilex device in reset until entire fabric enters user mode.

1.4.1. Device Initialization

The following steps summarize device initialization:

1. An external host drives a configuration request to the Secure Device Manager (SDM) by driving nCONFIG high. The SDM exits the IDLE state and signals the beginning of configuration by driving nSTATUS high and driving configuration data.
2. The SDM asserts CONF_DONE indicating that the Intel FPGA has successfully received all the configuration data.
3. For Intel Stratix 10 devices only, the SDM uses the configuration logic to start non-gated clocks in the fabric. Intel Hyperflex™ registers begin shifting data. Consequently, the initial conditions of Intel Hyperflex registers can be random. Use the **Disable Register Power-up Initialization** setting in the Intel Quartus Prime **Configuration** dialog box to disable Intel Hyperflex register initialization during power-on as explained below.
4. The SDM uses the configuration logic to enable and initialize user registers in the LABs, DSP, and embedded memory blocks.
5. The SDM drives INIT_DONE to indicate that the device has fully entered user mode. The Reset Release IP asserts nINIT_DONE. Intel recommends that you use nINIT_DONE to gate your reset logic.
6. The FPGA is now in user mode and ready for operation.

For more detailed information about the configuration flow, refer to the *Intel Stratix 10 Configuration User Guide* or the Intel Agilex Configuration User Guide.

Related Information

- Intel Stratix 10 Configuration User Guide
- Intel Agilex Configuration User Guide

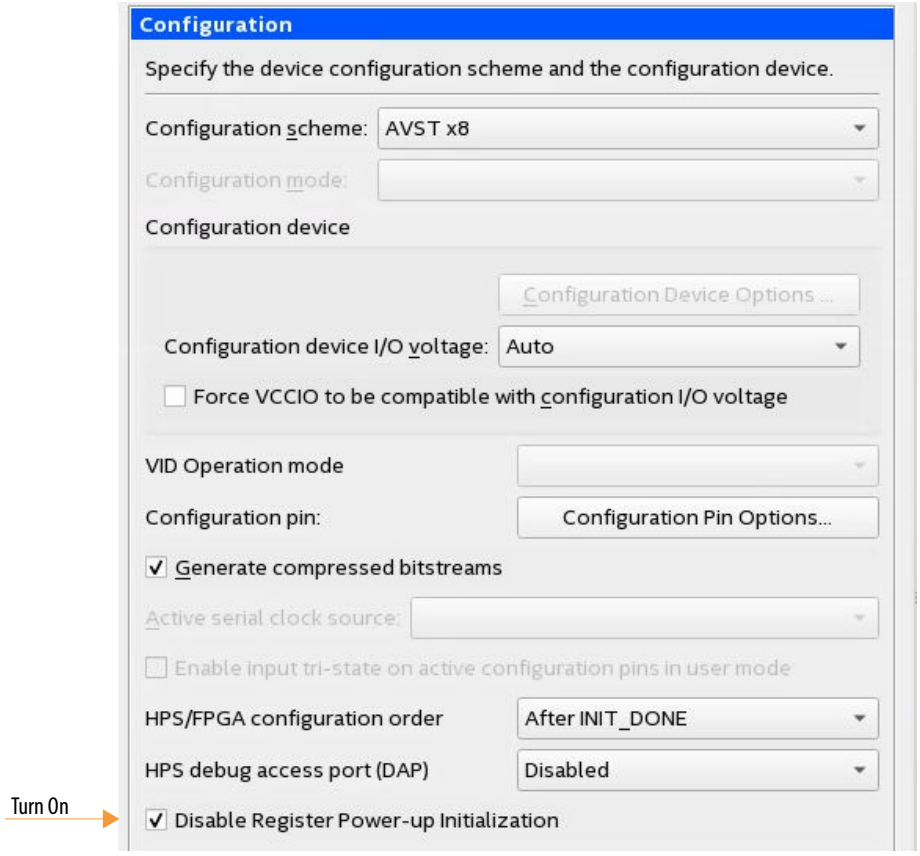
1.4.2. Preventing Register Initialization During Power-On

If not held in reset, both ALM and Intel Hyperflex registers may lose their initial state if they initialize before their respective source.

You can prevent registers from initializing during power-on by enabling an option in the Intel Quartus Prime software. Complete the following steps to turn on this option:

1. On the Assignments menu select **Device ➤ Device and Pin Options ➤ Configuration**.
2. In the **Configuration** dialog box, turn on **Disable Register Power-up Initialization.**

Figure 6. Disabling Register Initialization During Power-On



Note:

Intel also recommends that you not use initial conditions in your register transfer level (RTL) code.

1.4.3. Embedded Memory Block Initial Conditions

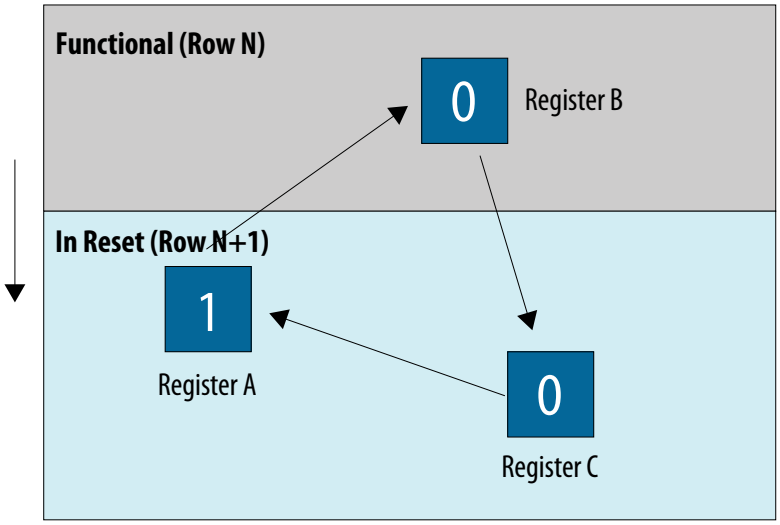
Initialized content of embedded memory blocks is stable during configuration. However, designs that contain logic to modify embedded memory can result in spurious writes. Spurious writes can occur if you fail to gate the write enable with an appropriate reset.

1.4.4. Protecting State Machine Logic

To guarantee correct operation of state machines, your reset logic must hold the FPGA fabric in reset until the entire fabric enters user mode.

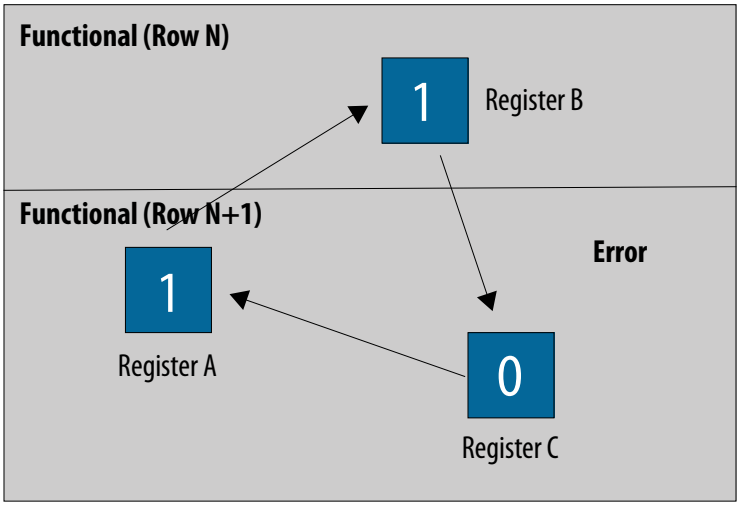
The following example shows how an inadequate reset strategy might result in an illegal state in a one-hot state machine. In this example, the design does not reset any of the state machine registers. The state machine design depends on registers entering an initial state. Without an adequate reset, This state machine begins operating when part of the device is active. Nearby logic included in the state machine remains frozen, before `INIT_DONE` has asserted.

Figure 7. Partially Initialized Design - INIT_DONE = 0



Register B in the active section is operational and takes on the value of Register A in the next clock cycle. Register A is still in the freeze register state and does not respond to the clock edge. Register A remains in the current state.

Figure 8. Advance One Clock Cycle, Device Completely In User Mode - INIT_DONE = 1



The entire fabric is now in user mode. The state machine enters an illegal or unknown state with two ones in a one-hot state machine. To prevent this illegal state, use the Reset Release IP to hold the circuit in reset until INIT_DONE asserts indicating that the entire fabric has entered user mode.

1.5. AN 891: Using the Reset Release Intel FPGA IP Archives

If an IP core version is not listed, the user guide for the previous IP core version applies.

IP Core Version	User Guide
19.1	AN 891: Using the Reset Release Intel Stratix 10 FPGA IP

Related Information

AN 891: Using the Intel FPGA Reset Release FPGA IP on page 3

1.6. Document Revision History for AN 891: Using the Reset Release Intel FPGA IP

Document Version	Intel Quartus Prime Version	Changes
2019.09.30	19.3	Made the following changes: • Added support for Intel Agilex devices. • The Intel Quartus Prime Pro Edition Software now performs a legality check that prevents you from including more than one Reset Release Intel FPGA IP in your design. • Added figure illustrating external loopback of the INIT_DONE signal.
2019.05.10	19.1	Initial release.