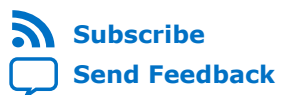




Intel® FPGA OCT IP Core User Guide

Updated for Intel® Quartus® Prime Design Suite: **17.1**



[Subscribe](#)

[Send Feedback](#)

[ug-altera_oct](#) | 2017.11.06

Latest document on the web: [PDF](#) | [HTML](#)



Contents

| | |
|--|----------|
| Intel® FPGA OCT IP Core User Guide..... | 3 |
| Intel FPGA OCT IP Core Features..... | 3 |
| Intel FPGA OCT IP Core Overview..... | 4 |
| RZQ Pin..... | 4 |
| OCT Block..... | 4 |
| OCT Logic..... | 4 |
| Intel FPGA OCT IP Core Functional Description..... | 5 |
| Power-Up Mode Interfaces..... | 7 |
| User Mode OCT..... | 8 |
| Intel FPGA OCT IP Core Design Example..... | 10 |
| Generating the Intel Quartus Prime Design Example..... | 10 |
| Intel FPGA OCT IP Core References..... | 11 |
| Intel FPGA OCT IP Core Parameter Settings..... | 11 |
| Intel FPGA OCT IP Core Signals..... | 11 |
| QSF Assignments..... | 12 |
| IP Migration Flow for Arria V, Cyclone V, and Stratix V Devices..... | 13 |
| Document Revision History for Intel FPGA OCT IP Core User Guide..... | 14 |



Intel® FPGA OCT IP Core User Guide

The Intel FPGA OCT IP core allows you to dynamically calibrate I/O with reference to an external resistor. The Intel FPGA OCT IP core improves signal integrity, reduces board space, and is necessary for communicating with external devices such as memory interfaces.

The Intel FPGA OCT IP core is available for Intel Arria® 10 and Intel Cyclone® 10 GX devices only. For Stratix® V, Arria V, and Cyclone V devices, you need to migrate the IP core. For more details, refer to the related information.

Related Links

- [Migrating Your ALTOCT IP Core to the Intel FPGA OCT IP Core](#) on page 13
Provides steps to migrate your ALTOCT IP core to the Intel FPGA OCT IP core.
- [Dynamic Calibrated On-Chip Termination \(ALTOCT\) IP Core User Guide](#)
Provides information about the ALTOCT IP core.
- [Introduction to Intel FPGA IP Cores](#)
Provides general information about all Intel FPGA IP cores, including parameterizing, generating, upgrading, and simulating IP cores.
- [Creating Version-Independent IP and Platform Designer Simulation Scripts](#)
Create simulation scripts that do not require manual updates for software or IP version upgrades.
- [Project Management Best Practices](#)
Guidelines for efficient management and portability of your project and IP files.

Intel FPGA OCT IP Core Features

The Intel FPGA OCT IP core supports the following features:

- Support for up to 12 on-chip termination (OCT) blocks
- Support for calibrated on-chip series termination (R_S) and calibrated on-chip parallel termination (R_T) on all I/O pins
- Calibrated termination values of 25 Ω and 50 Ω
- Support for OCT calibration in power-up and user modes

Intel FPGA OCT IP Core Overview

Figure 1. Intel FPGA OCT IP Core Top-Level Diagram

This figure shows the top-level diagram of the Intel FPGA OCT IP core.

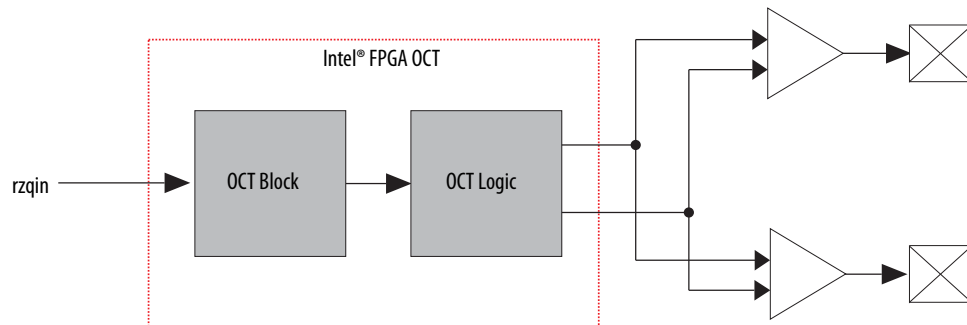


Table 1. Intel FPGA OCT IP Core Components

| Component | Description |
|-----------|---|
| RZQ pin | <ul style="list-style-type: none"> Dual-purpose pin. When used with OCT, the pin connects to an external reference resistor to calculate the calibration codes to implement the required impedance. |
| OCT block | Generates and sends calibration code words to the I/O buffer blocks. |
| OCT logic | Receives the calibration code words serially from the OCT block and sends the calibration code words in parallel to the buffers. |

RZQ Pin

Each OCT block has one RZQ pin.

- RZQ pins are dual-purpose pins. If the pins are not connected to the OCT block, you can use the pins as regular I/O pins.
- Calibrated pins must have the same V_{CCIO} voltage as the OCT block and the RZQ pin. Calibrated pins connected to the same OCT block must have the same series and parallel termination values.
- You can apply location constraints on the RZQ pins to determine the placement of the OCT block because the RZQ pin can only be connected to its corresponding OCT block.

OCT Block

The OCT block is a component that generates calibration codes to terminate the I/Os.

During calibration, the OCT matches the impedance seen on the external resistor through the `rzqin` port. Then, the OCT block generates two 16-bit calibration code words—one word calibrates the series termination and the other word calibrates the parallel termination. A dedicated bus sends the words serially to the OCT logic.

OCT Logic

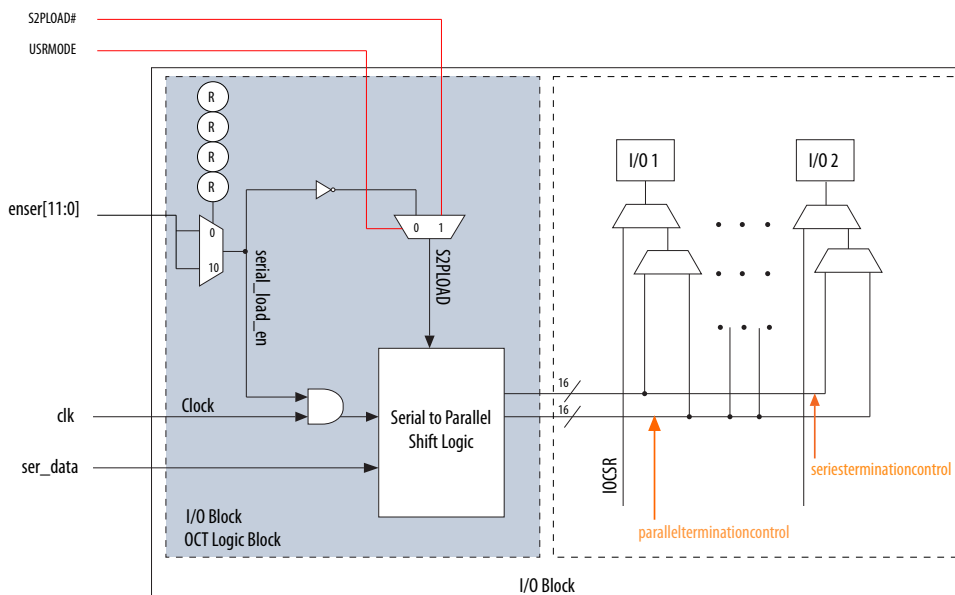
The OCT block sends the calibration code words serially to the OCT logic through the `ser_data` ports.



The `enser` signal, when triggered, specifies from which OCT block to read the calibration code words. The calibration code words are then buffered into the serial-to-parallel shift logic. After that, the Intel FPGA OCT IP core asserts the `S2PLOAD` signal to send the calibration code words in parallel to the I/O buffers.

The calibration code words activate or deactivate the transistors in the I/O block, which will emulate series or parallel resistance to match the impedance.

Figure 2. Internals of OCT Logic



Intel FPGA OCT IP Core Functional Description

To meet DDR memory specification, Intel Arria 10 and Intel Cyclone 10 GX devices support on-chip series termination (R_S OCT) and on-chip parallel termination (R_T OCT) for single-ended I/O standards. OCT can be supported on any I/O bank. The V_{CCIO} must be compatible for all I/Os in a given bank.

In an Intel Arria 10 or Intel Cyclone 10 GX device, there is one OCT block in each I/O bank. Each OCT block requires an association with an external 240 Ω reference resistor through an RZQ pin.

The RZQ pin shares the same V_{CCIO} supply with the I/O bank where the pin is located. An RZQ pin is a dual function I/O pin that you can use as a regular I/O if you do not use OCT calibration. When you use the RZQ pin for OCT calibration, the RZQ pin connects the OCT block to ground through an external 240 Ω resistor.

The following figures show how OCTs are connected in a single I/O column (in a daisy chain). An OCT can calibrate an I/O belonging to any bank, provided that the bank is in the same column and meets the voltage requirements. Because there are no connections between columns, OCT can only be shared if the pins belong to the same I/O column of the OCT.

Figure 3. Intel FPGA OCT Bank-to-Bank Connections

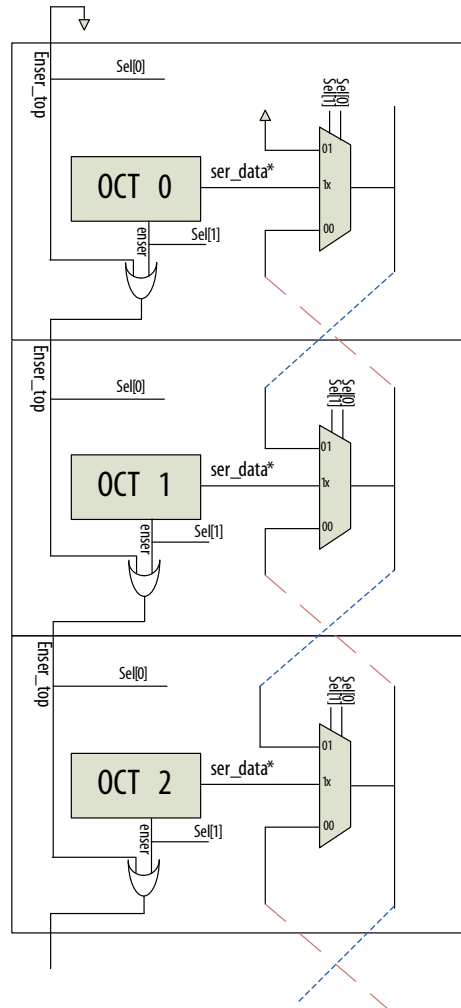
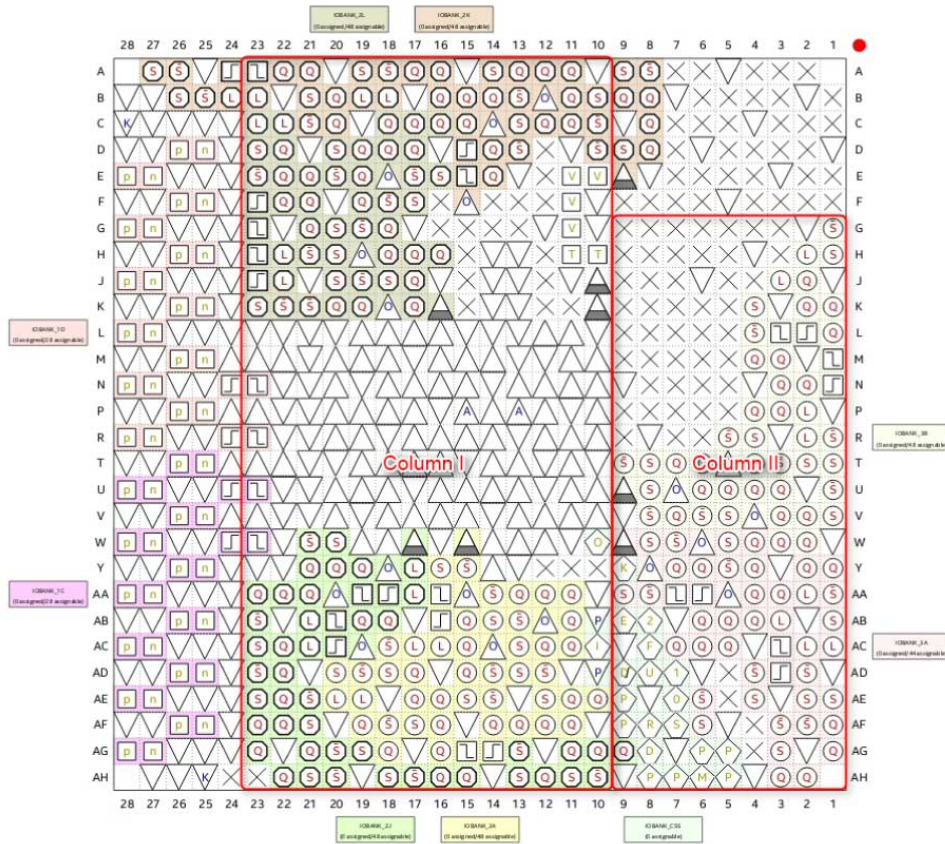




Figure 4. I/O Columns in Intel Quartus® Prime Pin Planner

This figure is an example. The layout varies between different Intel Arria 10 or Intel Cyclone 10 GX devices.

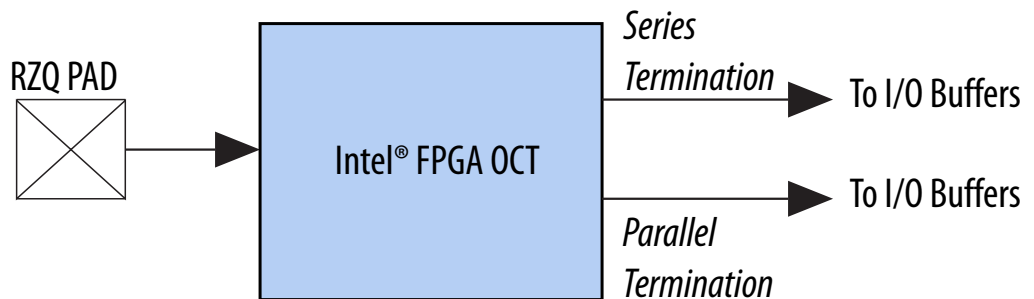


Power-Up Mode Interfaces

The Intel FPGA OCT IP core in power-up mode has two main interfaces:

- One input interface connecting the FPGA RZQ pad to the Intel FPGA OCT block
- Two 16-bit words output which connect to I/O buffers

Figure 5. Intel FPGA OCT Interfaces

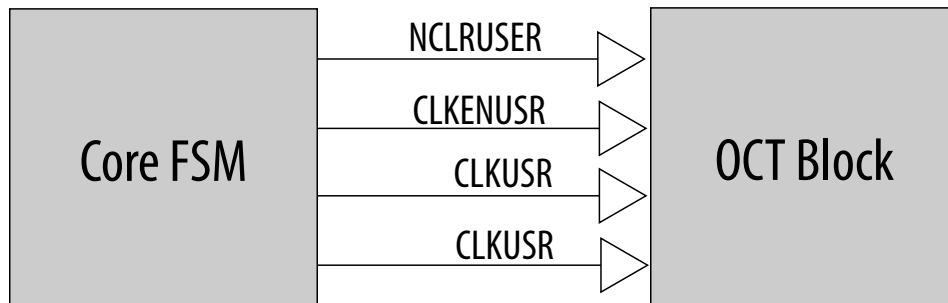


User Mode OCT

User mode OCT operates the same way as power-up OCT mode, with the addition of user controllability.

Figure 6. FSM Signals

This figure shows a finite state machine (FSM) in the core controls the dedicated user signals on the OCT block. The FSM ensures that the OCT block calibrates or sends controlling code words as per your request.



The Fitter does not infer a user-mode OCT. If you want your OCT block to use the user mode OCT feature, you must generate the Intel FPGA OCT IP core. However, because of hardware limitations, you can only use one Intel FPGA OCT IP core in user mode OCT in your design.

Note: A single Intel FPGA OCT IP core can control up to 12 OCT blocks.

The FSM provides the following signals:

- clock
- reset
- s2pload
- calibration_busy
- calibration_shift_busy
- calibration_request

Note: These signals are only available in user-mode and not power-up mode.

Related Links

[Intel FPGA OCT IP Core Signals](#) on page 11
 Provides more information about the FSM signals.



Core FSM

Figure 7. FSM Flow

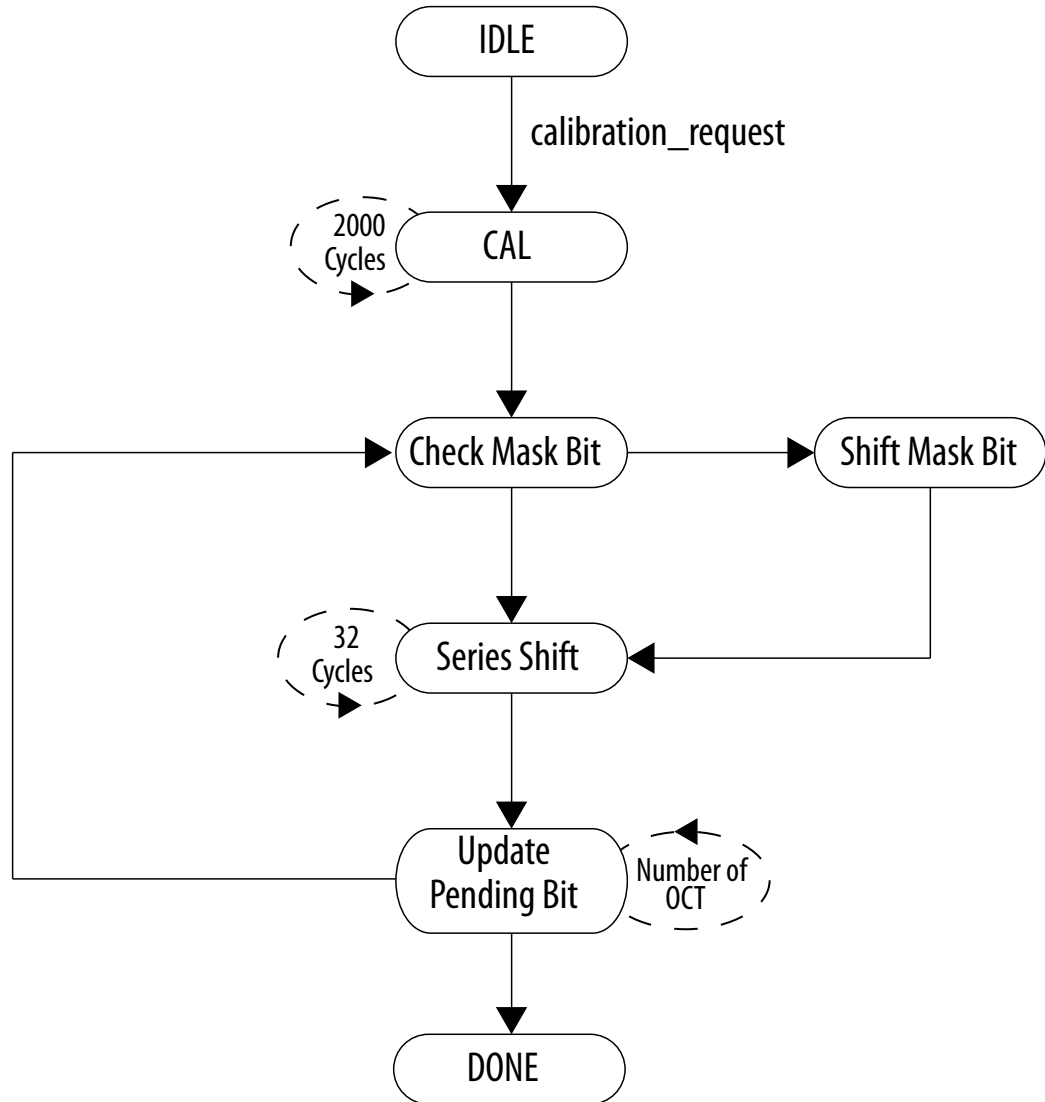


Table 2. FSM States

| State | Description |
|----------------|---|
| IDLE | When you set the calibration_request vector, the FSM moves from IDLE state to CAL state. Keep the calibration_request vector at its value for two clock cycles. After two clock cycles, the FSM contains a copy of the vector. You must reset the vector to avoid reinitiating the calibration process. |
| CAL | During this state, the FSM checks which bits in the calibration_request vector were asserted and services them. The corresponding OCT blocks starts the calibration process that takes around 2,000 clock cycles to complete. After calibration completes, the calibration_busy signal is released. |
| Check Mask bit | The FSM checks each bit in the vector if the bit is set or not. |

continued...



| State | Description |
|--------------------|---|
| Shift Mask bit | This state simply loops over all the bits in the vector until it hits a 1. |
| Series Shift | This state serially sends the termination code from the OCT block to the termination logic. It takes 32 cycles to complete the transfer. After each transfer, the FSM check for any pending bits in the vector and services them accordingly. |
| Update Pending Bit | The pending register holds bits that corresponds to every OCT block in the Intel FPGA OCT IP core. This state updates the pending register by resetting the serviced request. |
| DONE | When the <code>calibration_shift_busy</code> signal is deasserted, you can assert <code>s2pload</code> to transfer the new termination codes into the buffers. You must assert the <code>s2pload</code> signal for at least 25 ns. Because of hardware limitations, you cannot request another calibration until all bits in <code>calibration_shift_busy</code> vector are low. |

Intel FPGA OCT IP Core Design Example

The Intel FPGA OCT IP core can generate a design example that matches the same configuration chosen for the IP core.

The design example is a simple design that does not target any specific application. You can use the design example as a reference on how to instantiate the IP core.

To generate the design example files, turn on the **Generate Example Design** option in the **Generation** dialog box during IP core generation.

Note: The Intel FPGA OCT IP core does not support VHDL generation.

- The software generates the `<instance>_example_design` directory along with the IP core, where `<instance>` is the name of your IP core.
- The `<instance>_example_design` directory contains the `make_qii_design.tcl` scripts.

Note: The `.qsys` files are for internal use during design example generation only. You cannot edit the files.

Generating the Intel Quartus Prime Design Example

The `make_qii_design.tcl` script generates a synthesizable design example along with an Intel Quartus Prime project, ready for compilation.

To generate synthesizable design example, follow these steps.

1. After generating the IP core together with the design example files, run the following script at the command prompt: `quartus_sh -t make_qii_design.tcl`.
2. If you want to specify an exact device to use, use the following command: `quartus_sh -t make_qii_design.tcl <device_name>`.

The script generates a `qii` directory that contains the `ed_synth.qpf` project file. You can open and compile this project in the Intel Quartus Prime software.



Intel FPGA OCT IP Core References

Intel FPGA OCT IP Core Parameter Settings

Table 3. Intel FPGA OCT IP Core Parameters

| Name | Value | Description |
|--|---|--|
| Number of OCT blocks | 1 to 12 | Specifies the number of OCT blocks to be generated. The default value is 1 . |
| Use backwards-compatible port names | <ul style="list-style-type: none"> On Off | Check this to use legacy top-level names compatible with the ALTOCT IP core. This parameter is disabled by default. |
| OCT mode | <ul style="list-style-type: none"> Power-up User | Specifies whether OCT will be user-controlled or not. The default value is Power-up . |
| OCT block x calibration mode | <ul style="list-style-type: none"> Single Double POD | Specifies the calibration mode for the OCT. X corresponds to the number of the OCT block. The default value is Single . |

Intel FPGA OCT IP Core Signals

Table 4. Input Interface Signals

| Signal Name | Direction | Description |
|------------------------|-----------|--|
| rzqin | Input | Input connection from RZQ pad to the OCT block. RZQ pad is connected to an external resistance. The OCT block uses impedance connected to the rzqin port as a reference to generate the calibration code. This signal is available for power-up and user modes. |
| s2pload | Input | Asynchronous input signal to transfer the termination code from the termination logic block to the buffers. Assert this signal for at least 25 ns. |
| clock | Input | Input clock for user mode OCT. The clock must be 20 MHz or less. |
| reset | Input | Input reset signal. Reset is synchronous. |
| calibration_request | Input | Input vector for [NUMBER_OF_OCT:0]. Every bit corresponds to an OCT block. When a bit is set to 1, the corresponding OCT will calibrate, then serially shift the code word into the termination logic block. The request has to be held for two clock cycles. Due to hardware limitations, you must wait until the calibration_shift_busy vector to be zero until another request is issued; otherwise your request will not be processed. |
| calibration_shift_busy | Output | Output vector for [NUMBER_OF_OCT:0] indicating which OCT block is currently working on calibration and shifting termination codes to the termination logic block. When a bit is 1, it indicates that an OCT block is calibrating and shifting the code word to the termination logic block. |

continued...



| Signal Name | Direction | Description |
|--|-----------|---|
| calibration_busy | Output | Output vector for [NUMBER_OF_OCT: 0] indicating which OCT block is currently working on calibration. When a bit is 1, it indicates that an OCT block is calibrating |
| oct_<x>_series_termination_control[15:0] | Output | 16-bit output signal, with <x> ranging from 0 to 11. This signal connects to the seriesterminationcontrol port on the input/output buffer. This port sends the series termination code that calibrates R_s . |
| oct_<x>_parallel_termination_control[15:0] | Output | 16-bit output signal, with <x> ranging from 0 to 11. This signal connects to the parallelerminationcontrol port on the input/output buffer. This port sends the parallel termination code that calibrates R_t . |

QSF Assignments

Intel Arria 10 and Intel Cyclone 10 GX devices have the following termination-related Intel Quartus Prime settings file (.qsf) assignments:

- INPUT_TERMINATION
- OUTPUT_TERMINATION
- TERMINATION_CONTROL_BLOCK
- RZQ_GROUP

Table 5. QSF Assignments

| QSF Assignment | Details |
|---|--|
| INPUT_TERMINATION OUTPUT_TERMINATION | <p>The input/output termination assignment specifies the termination value in ohm on the pin in question.</p> <p>Example:</p> <pre>set_instance_assignment -name INPUT_TERMINATION <value> -to <pin name> set_instance_assignment -name OUTPUT_TERMINATION <value> -to <pin name></pre> <p>To enable the series/parallel termination ports, include these assignments, which will specify the series and parallel termination values for the pins.</p> <p>Make sure to connect the seriesterminationcontrol and parallelerminationcontrol ports from the Intel FPGA OCT IP core to the Intel FPGA GPIO core.</p> <p>Example:</p> <pre>set_instance_assignment -name INPUT_TERMINATION "PARALLEL <VALUE> OHM WITH CALIBRATION" -to <pin> set_instance_assignment -name OUTPUT_TERMINATION "SERIES <VALUE> OHM WITH CALIBRATION" -to <pin></pre> |
| TERMINATION_CONTROL_BLOCK | <p>Directs the Fitter to make the proper connection from the desired OCT block to the specified pins. This assignment is useful when I/O buffers are not explicitly instantiated and you need to associate the pins with a specific OCT block.</p> <p>Example:</p> <pre>set_instance_assignment -name TERMINATION_CONTROL_BLOCK <desired OCT BLK> -to <pin name></pre> |
| RZQ_GROUP | <p>This assignment is supported in Intel Arria 10 and Intel Cyclone 10 GX devices only. This assignment creates an Intel FPGA OCT IP core without modifying the RTL.</p> <p>The Fitter searches for the rzq pin name in the netlist. If the pin does not exist, the Fitter creates the pin name along with the Intel FPGA OCT IP core and its corresponding connections. This allows you to create a group of pins to be calibrated by an existing or non-existing OCT and the Fitter ensures the legality of the design.</p> |

continued...



| QSF Assignment | Details |
|----------------|--|
| | Example: <pre>set_instance_assignment -name RZQ_GROUP <rzq pin name> -to <pin name></pre> |

Termination can exist on input and output buffers, and sometimes simultaneously.

There are two methods to associate pin groups with an OCT block:

- Use a `.qsf` assignment to indicate which pin (bus) is associated with which OCT block. You can use the `TERMINATION_CONTROL_BLOCK` or `RZQ_GROUP` assignment. The former assignment associates a pin with an OCT instantiated in the RTL while the latter associates the pin with a newly created OCT without modifying the RTL.
- Instantiate the I/O buffer primitives at the top level and connect them to the appropriate OCT blocks.

Note: All I/O banks with the same V_{CCIO} can share one OCT block even if that particular I/O bank has its own OCT block. You can connect any number of I/O pins that support calibrated termination to an OCT block. Ensure that you connect I/Os with compatible configuration to an OCT block. You must also ensure that the OCT block and its corresponding I/Os have the same V_{CCIO} and series or parallel termination values. With these settings, the Fitter places the I/Os and OCT block in the same column. The Intel Quartus Prime software generates warning messages if there is no pin connected to the block.

IP Migration Flow for Arria V, Cyclone V, and Stratix V Devices

The IP migration flow allows you to migrate the ALTOCT IP core of Arria V, Cyclone V, and Stratix V devices to the Intel FPGA OCT IP core of Intel Arria 10 or Intel Cyclone 10 GX devices.

The IP migration flow configures the Intel FPGA OCT IP core to match the settings of the ALTOCT IP core, allowing you to regenerate the IP core.

Note: This IP core supports the IP migration flow in single OCT calibration mode only. If you are using double or POD calibration mode, you do not need to migrate the IP core.

Migrating Your ALTOCT IP Core to the Intel FPGA OCT IP Core

To migrate your ALTOCT IP core to the Intel FPGA OCT IP core, follow these steps:

1. Open your ALTOCT IP core in the IP Catalog.
2. In **Currently selected device family**, select **Arria 10** or **Cyclone 10 GX**.
3. Click **Finish** to open the Intel FPGA OCT IP core in the parameter editor. The parameter editor configures the Intel FPGA OCT IP core settings similar to the ALTOCT IP core settings.
4. If there are any incompatible settings between the two, select **new supported settings**.
5. Click **Finish** to regenerate the IP core.
6. Replace your ALTOCT IP core instantiation in RTL with the Intel FPGA OCT IP core.



Note: The Intel FPGA OCT IP core port names may not match the ALTOCT IP core port names. Therefore, simply changing the IP core name in the instantiation is not sufficient.

Document Revision History for Intel FPGA OCT IP Core User Guide

| Date | Version | Changes |
|----------------|------------|---|
| November 2017 | 2017.11.06 | <ul style="list-style-type: none">Added support for Intel Cyclone 10 GX devices.Renamed Altera OCT IP core to Intel FPGA OCT IP core.Renamed Qsys to Platform Designer.Updated text for additional Intel rebranding. |
| May 2017 | 2017.05.08 | Rebranded as Intel. |
| December, 2015 | 2015.12.07 | <ul style="list-style-type: none">Changed instances of "megafunction" to "IP core".Changed instances of <i>Quartus II</i> to <i>Quartus Prime</i>.Various edits to contents and links to improve style and clarity. |
| August, 2014 | 2014.08.18 | <ul style="list-style-type: none">Added information about OCT calibration in user mode.Updated the IP core signals and parameters:<ul style="list-style-type: none">– <code>core_rzqin_export</code> changed to <code>rzqin</code>– <code>core_series_termination_control_export</code> changed to <code>oct_<x>_series_termination_control[15:0]</code>– <code>core_parallel_termination_control_export</code> changed to <code>oct_<x>_parallel_termination_control[15:0]</code> |
| November, 2013 | 2013.11.29 | Initial release. |