

# Intel IP Catalog Flow

Thomas Zerrer

V 1.4

**This AppNote describes how to use the IP catalog flow for all supported Intel FPGA devices**

- How to generate the IP Core
- make specific parameter settings so that the generated IP runs with the Smartlogic core
- How to activate MSI and MSI-X interrupts (only available in the HCC version of the Smartlogic IP core)

### **In order to use the IP catalog flow it is necessary to reference to the correct files in the IP Database**

- Make sure to use `pcie_core_connection_ipi.vhd` instead of `pcie_core_connection.vhd`, if you are not working in multifunction mode
- In case you work with multi-function, you have to include `pcie_core_connection_mf.vhd` in the QSF file
- In case you migrate from the old flow to the new IP catalog flow, you have to delete the direct references to the Intel HIP related files. By adding the HIP IP core from the IP catalog, these files will be added automatically again in the background via the `.ip` / `.qip` files.
- A demo project is available that works as an example

For Cyclone V FPGAs two IP catalog cores have to be generated and added to the design:

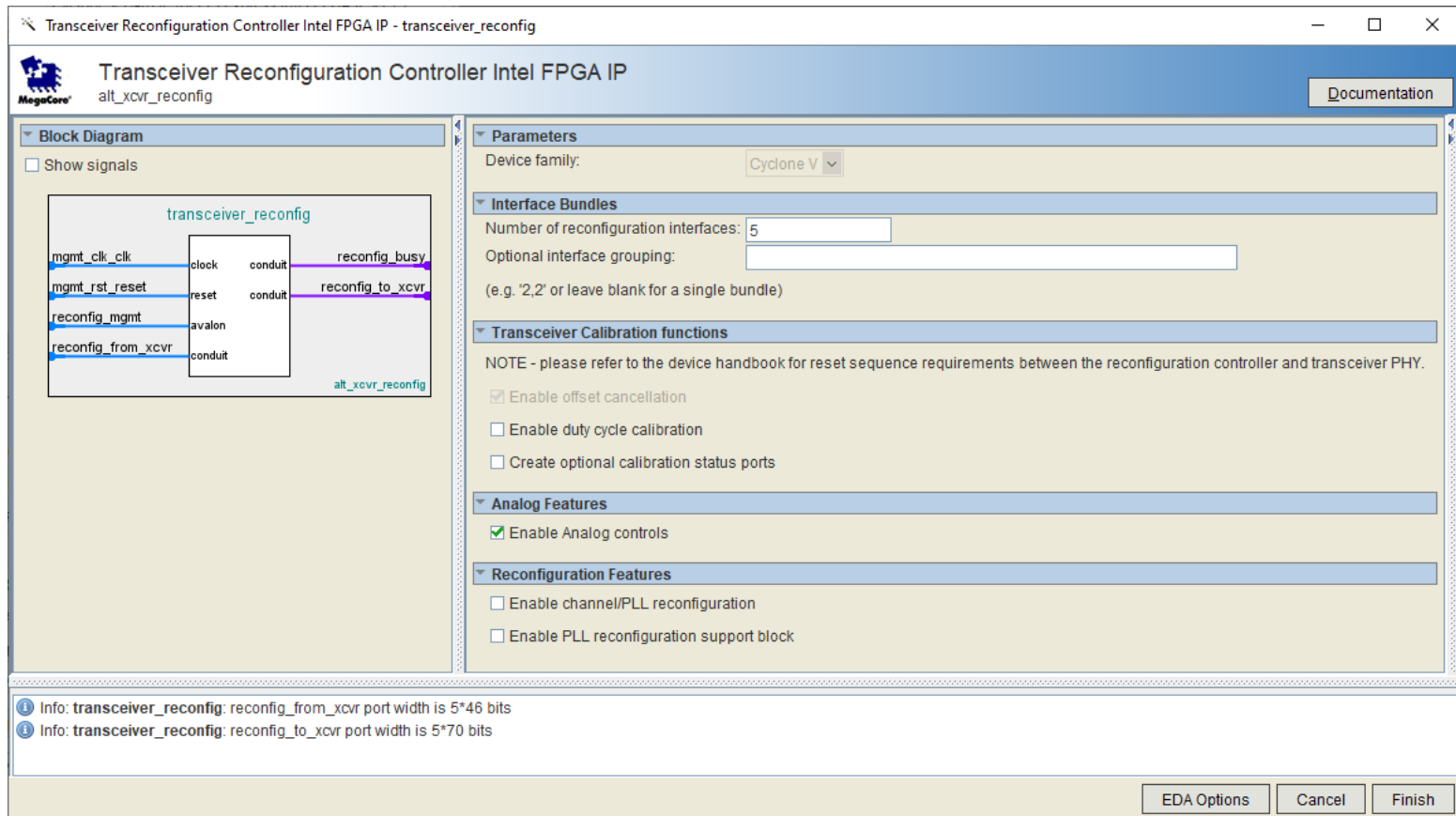
1. The Transceiver Reconfiguration Controller Intel FPGA IP core with the IP variation filename „transceiver\_reconfig.v“
2. The „Cyclone V Hard IP for PCI Express Intel FPGA“ IP core with the IP variation filename „intel\_c5\_pcie2.v“

Select those IPs in the IP catalog and prepare them for configuration by double clicking.

Make sure that your project contains the following flow specific files:

- The file „pcie\_core\_connection\_ipi.vhd“ has to be added to the project instead of pcie\_core\_connection.vhd in non multi function mode and pcie\_core\_connection\_mf.vhd in multi function mode.
- The IP Core wrapper File „altera\_a5\_c5\_pcie2\_wrapper.v“ has to be added to the project instead of the file „altera\_c5\_pcie2.sv“

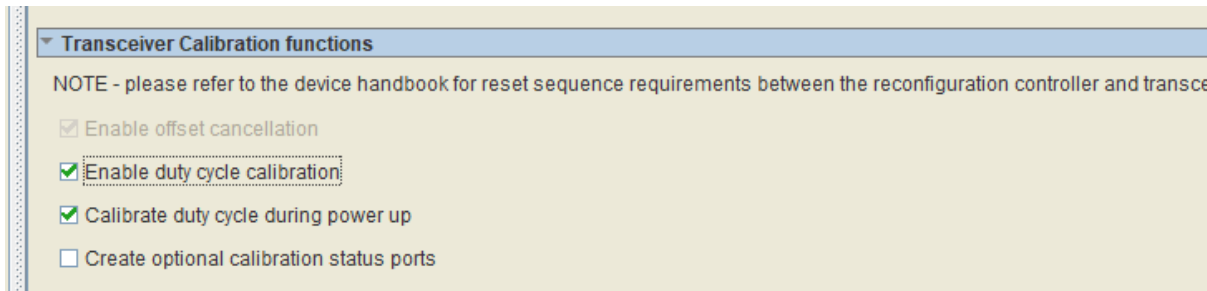
## GUI of the Transceiver Reconfiguration Controller



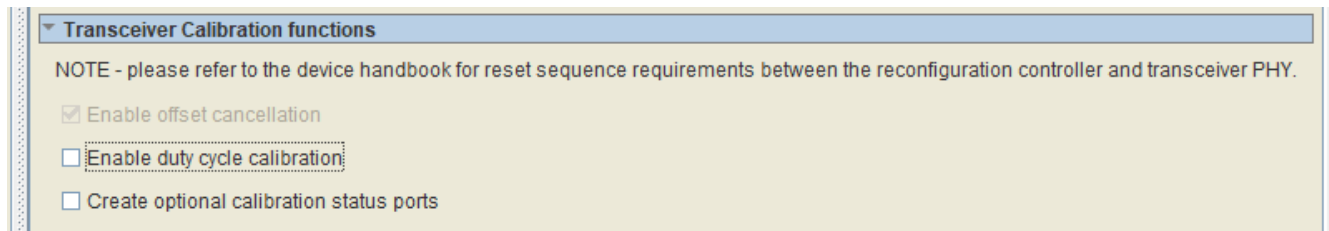
Special care has to be taken regarding the number of reconfiguration interfaces and the duty cycle correction. The next two slides show how to configure them correctly.

Duty cycle correction : Working around a Quartus Prime GUI Problem

First check „Enable duty cycle calibration“ and the GUI should look like this:



Now deselect Enable duty cycle calibration, while leaving „Calibrate duty cycle during power up“ checked and the GUI looks like this :



Although it seems, that no duty cycle calibration is selected, the „Calibrate duty cycle during power up“ is still active. This is the correct setting.

The number of reconfiguration interfaces has to be entered according to the following table:

PCIe Linkwidth	Number of reconfiguration interfaces
X1	2
X2	3
X4	5

```
1 // megafunction wizard: %Transceiver Reconfiguration Controller Intel FPGA IP v19.1%
2 // GENERATION: XML
3 // transceiver_reconfig.v
4
5 // Generated using ACDS version 19.1 670
6
7 `timescale 1 ps / 1 ps
8 module transceiver_reconfig (
9     output wire reconfig_busy, // reconfig_busy.reconfig_busy
10    input wire mgmt_clk_clk, // mgmt_clk_clk.clk
11    input wire mgmt_rst_reset, // mgmt_rst_reset.reset
12    input wire [6:0] reconfig_mgmt_address, // reconfig_mgmt.address
13    input wire reconfig_mgmt_read, // reconfig_mgmt.read
14    output wire [31:0] reconfig_mgmt_readdata, // reconfig_mgmt.readdata
15    output wire reconfig_mgmt_waitrequest, // reconfig_mgmt.waitrequest
16    input wire reconfig_mgmt_write, // reconfig_mgmt.write
17    input wire [31:0] reconfig_mgmt_writedata, // reconfig_mgmt.writedata
18    output wire [349:0] reconfig_to_xcvr, // reconfig_to_xcvr.reconfig_to_xcvr
19    input wire [229:0] reconfig_from_xcvr, // reconfig_from_xcvr.reconfig_from_xcvr
20);
21
22 alt_xcvr_reconfig #(
23     .device_family ("Cyclone-V"),
24     .number_of_reconfig_interfaces (5),
25     .enable_offset (1),
26     .enable_lc (0),
27     .enable_dcd (0),
28     .enable_dcd_power_up (1),
29     .enable_analog (1),
30     .enable_eyemon (0),
31     .enable_ber (0),
32     .enable_dfe (0),
33     .enable_adce (0),
34     .enable_mif (0),
35     .enable_pll (0)
26
```

After you have generated the Transceiver Reconfiguration Controller, open the generated file „transceiver\_reconfig.v“ with a text editor. The parameter settings of alt\_xcvr\_reconfig .enable\_dcd must be 0 and .enable\_dcd\_power\_up, .enable\_offset and .enable\_analog must be 1. If they have a different setting, repeat the configuration process for the IP.



## GUI of the Cyclone V HIP:

**System Settings**

- Number of lanes: x4
- Lane rate: Gen2 (5.0 Gbps)
- Port type: Native endpoint
- Application interface: Avalon-ST 128-bit
- RX buffer credit allocation - performance for received requests: **Low**
- Reference clock frequency: 100 MHz
- Use 62.5 MHz application clock
- Use deprecated RX Avalon-ST data byte enable port (rx\_st\_be)
- Enable configuration via the PCIe link
- Enable Hard IP reconfiguration
- Number of Functions: 1

**Function Capabilities**

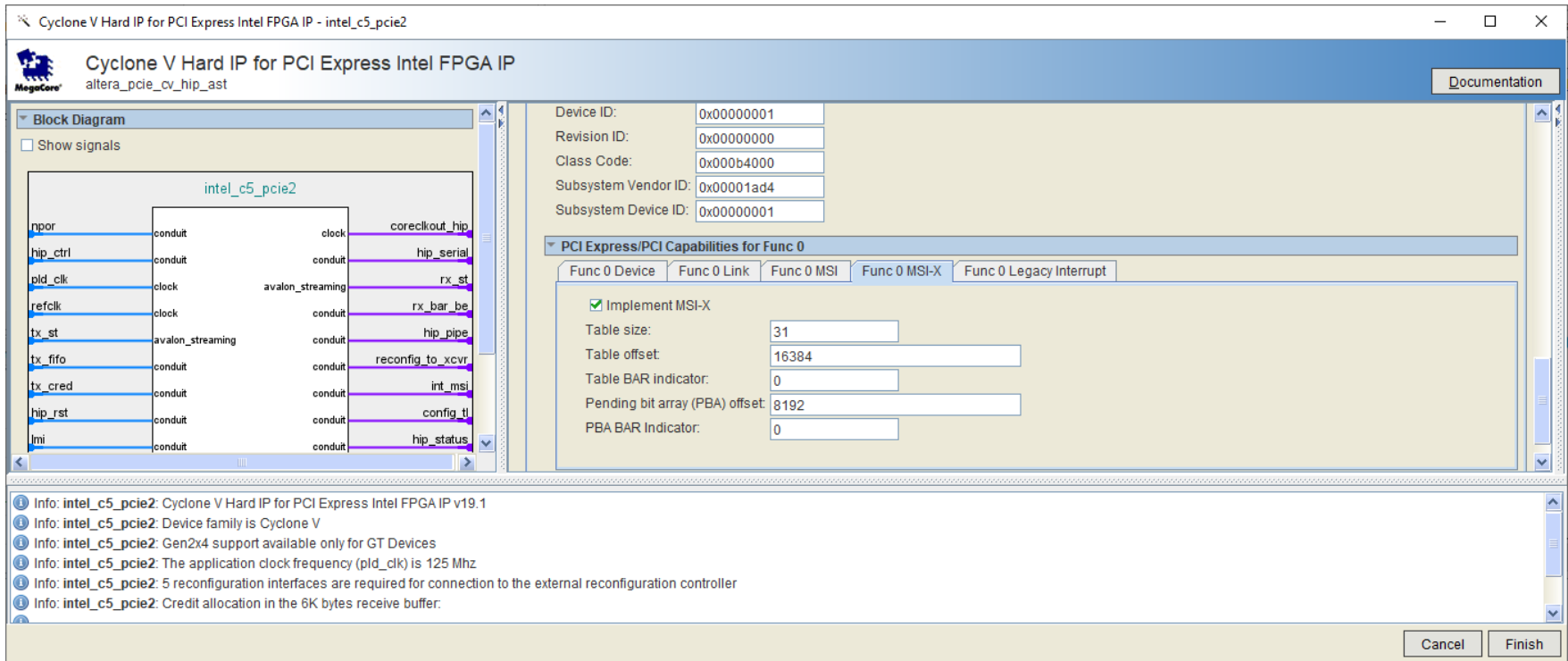
Shared PCI Express/PCI Capabilities Across All Functions

Link port number: 1

Slot clock configuration

Info: intel\_c5\_pcie2: Cyclone V Hard IP for PCI Express Intel FPGA IP v19.1  
Info: intel\_c5\_pcie2: Device family is Cyclone V  
Info: intel\_c5\_pcie2: Gen2x4 support available only for GT Devices  
Info: intel\_c5\_pcie2: The application clock frequency (pld\_clk) is 125 Mhz  
Info: intel\_c5\_pcie2: 5 reconfiguration interfaces are required for connection to the external reconfiguration controller  
Info: intel\_c5\_pcie2: Credit allocation in the 6K bytes receive buffer:  
Info: intel\_c5\_pcie2: Posted : header=16 data=16  
Info: intel\_c5\_pcie2: Non posted: header=16 data=0

GUI of the Cyclone V HIP : Example how to activate 32 MSI-X interrupts :



Currently only 32 MSI-X interrupts are supported, please do not try to enter other values.

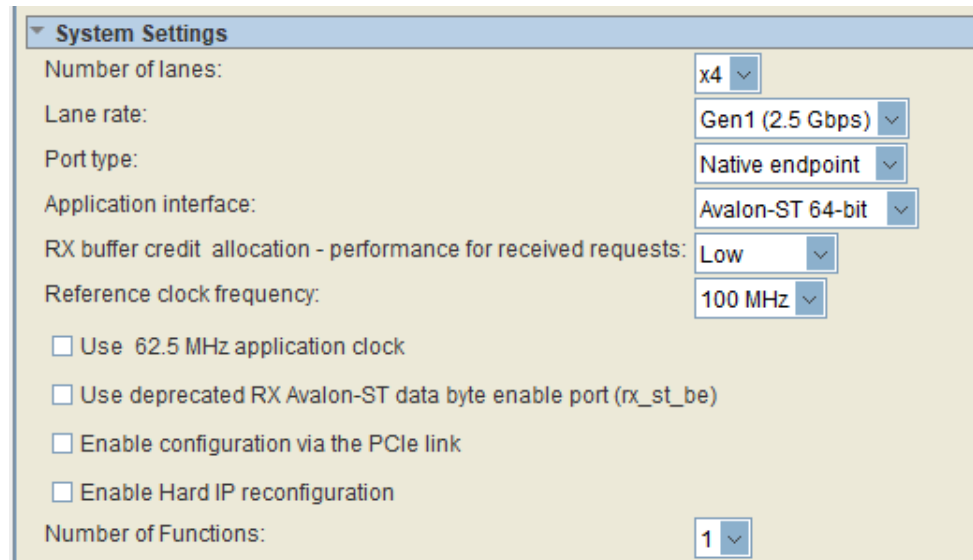
Also make sure to set the generic/parameter use\_msix\_g to 1 !

## Cyclone V – working around a HIP GUI Error

It has been observed for Cyclone V, that the GUI parameters might not be transferred correctly to the IP's HDL parameter section during the IP generation process of Quartus. This problem happens especially when the link speed is changed from Gen2 to Gen1. In this case the GUI looks right, but the 62.5 MHz option is activated in error. This results in an illegal PCIe HIP configuration and the FPGA design does not communicate correctly via PCIe and causes a system crash.

A clear indication that your design is affected is when Quartus reports warning message #332056.

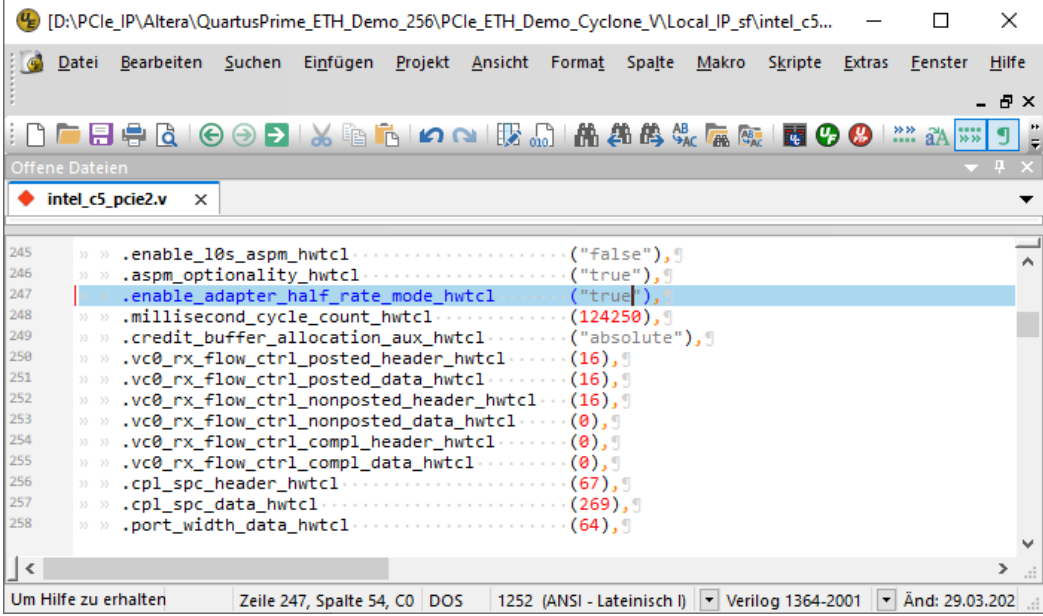
Example:



Although „use 62.5 MHz application clock“ is unchecked, the generated IP might have this option turned on.

## Cyclone V – working around a HIP GUI Error

In order to check, if your design is affected, open the generated file `intel_c5_pcie2.v` and search for the parameter `.enable_adapter_half_rate_mode_hwtcl`:



```
245  >> .enable_10s_aspm_hwtcl ..... ("false"),  
246  >> .aspm_optionality_hwtcl ..... ("true"),  
247  >> .enable_adapter_half_rate_mode_hwtcl ..... ("true"),  
248  >> .millisecond_cycle_count_hwtcl ..... (124250),  
249  >> .credit_buffer_allocation_aux_hwtcl ..... ("absolute"),  
250  >> .vc0_rx_flow_ctrl_posted_header_hwtcl ..... (16),  
251  >> .vc0_rx_flow_ctrl_posted_data_hwtcl ..... (16),  
252  >> .vc0_rx_flow_ctrl_nonposted_header_hwtcl ..... (16),  
253  >> .vc0_rx_flow_ctrl_nonposted_data_hwtcl ..... (0),  
254  >> .vc0_rx_flow_ctrl_compl_header_hwtcl ..... (0),  
255  >> .vc0_rx_flow_ctrl_compl_data_hwtcl ..... (0),  
256  >> .cpl_spc_header_hwtcl ..... (67),  
257  >> .cpl_spc_data_hwtcl ..... (269),  
258  >> .port_width_data_hwtcl ..... (64),
```

If this parameter is set to „true“ and you are not operating in G2-X4 mode, your design is affected.

In this case you have to set all 4 checkboxes in the „System Settings“ Tab (see previous page) and then clear them all and re-generate the IP again. Do not change the verilog file as it will be overwritten when you change other parameters. After re-generating the file, re-check if the setting of `enable_adapter_half_rate_mode_hwtcl` is now correct.

## Cyclone V – working around a HIP GUI Error

---

Summary of the Quartus GUI Error:

- The GUI error seems to affect all IP cores
- Problem is, that the GUI representation of the checkboxes seems not to be in sync with the real parameter values
- This problem leads to illegal designs and even system crashes
- It has been observed in QuartusPrime Lite 19.1 and might occur also in newer versions
- Most likely it affects not only Cyclone V but also all other FPGA design families

## Cyclone V – Configuring the HIP

---

### Important when you configure the Cyclone V HIP core:

- Never check „Use 62.5 MHz application clock“. The Smartlogic IP will not work with this setting
- Never check „use rx\_st\_be“
- Set the RX Buffer credit allocation to „Low“ when the FPGA uses DMA Read. In all other cases, set this to „Balanced“
- You may turn on „Enable configuration via the PCIe link“ also known as CVP  
If you turn on CVP make sure, that the PCIe 100 MHz reference clock is connected to sys\_clk\_ip of the Smartlogic IP core and that a free running 125 MHz clock is connected to sys\_clk\_in of the Smartlogic IP core. If you do not work with CVP, connect the 100 MHz clock to both sys\_clk\_ip and sys\_clk\_in
- Check „slot clock configuration“, if you are operating with a common reference clock between root complex and FPGA endpoint
- If you are running in Gen2-X4 mode, make sure to set the Application interface to Avalon-ST 128-Bit. In all other cases use the 64-Bit version
- Set the Device Identification Registers and the PCI Express Capabilities for Func 0 according to your needs
- You may configure additional PCIe functions in case you have a multi-function license

A demo design, where the Cyclone V IPs are correctly setup, is available from Smartlogic.

**Make sure to configure the following parameters of the Smartlogic IP core in the same way as you entered them into the HIP GUI:**

pcie\_ep\_config\_pkg.vhd

Parameter	Comment
PCle_synth_Core_Type_c	„CV“
Enable_SLOT_CLOCK_C PCIE_MSI_CAP_MULTIMSGCAP_C PCle_BAR<x>_C PCle_CLASS_CODE_C PCle_VEN_ID_C PCle_DEV_ID_C PCle_SUBSYS_VEN_ID_C PCle_SUBSYS_DEV_ID_C PCle_link_cap_max_link_speed_c PCle_link_cap_max_link_width_c	Set these constants in the same way as you entered them in the GUI.

# Cyclone 5 – Selecting the correct Avalon-ST bitwidth

The Avalon ST data bitwidth has to be entered according to the following table:

<b>PCIe Linkspeed / Linkwidth</b>	<b>Avalon ST bitwidth</b>
<b>G2-X4</b>	<b>128</b>
<b>G1-X8</b>	<b>128</b>
<b>All other link width / link speed combinations</b>	<b>64</b>



---

For Arria 10 FPGAs one IP catalog core has to be generated and added to the design:

1. The „Arria 10 Hard IP for PCI Express“ IP core with the IP variation filename „altera\_a10\_pcie3.v“

Make sure that your project contains the following flow specific files:

- The file „pcie\_core\_connection\_ipi.vhd“ has to be added to the project instead of pcie\_core\_connection.vhd. Note that PCIe multi-function is not supported from the PCIe HIP for Arria 10.
- The IP Core wrapper File „altera\_a10\_pcie3\_wrapper.v“ has to be added to the project

# Arria 10 – Configuring the HIP

## GUI of the Arria 10 HIP:

IP Parameter Editor Pro - altera\_a10\_pcie3 (D:\PCIe\_IP\Altera\QuartusPrime\_HCC\_Demo\_256\DMA\_Demo3\_Arria\_X\Local\_IP\altera\_a10\_pcie3.ip)

File Edit System Generate View Tools Help

Parameters

System: altera\_a10\_pcie3 Path: pcie\_a10\_hip\_0

Intel Arria 10/Cyclone 10 Hard IP for PCI Express  
altera\_pcie\_a10\_hip

Design Environment

This component supports multiple interface views:

Standalone

IP Settings Example Designs

PCI Express / PCI Capabilities Configuration, Debug and Extension Options PHY Characteristics

System Settings Avalon-ST Settings Base Address Registers Device Identification Registers

Application interface type: Avalon-ST

Hard IP mode: Gen2:x4, Interface: 128 bit, 125 MHz

Port type: Native endpoint

RX buffer credit allocation for received requests vs completions: **Low**

RX Buffer completion credits: Header:195 Data:773

System Messages

Type	Path	Message
3 Info Messages		
altera_a10_pcie3.pcie_a10_hip_0	device_family is Arria 10	
altera_a10_pcie3.pcie_a10_hip_0	part_trait_device is 10AS048E4F29I3SG	
altera_a10_pcie3.pcie_a10_hip_0	Gen2 (5.0 Gbps) x4 128-bit	

0 Errors, 0 Warnings

Details Block Symbol

Show signals

pcie\_a10\_hip\_0

Inputs: pld\_clk, refclk, npor, pin\_perst, pld\_core\_ready, pld\_clk\_inuse, serdes\_pll\_locked, reset\_status

Outputs: clk, refclk, npor, pin\_perst, hip\_rst, pld\_core\_ready, pld\_clk\_inuse, serdes\_pll\_locked, reset\_status

Other signals: coreclk, rx\_st, startofpacket, endofpacket, error, valid, ready, data, empty, clr\_st, reset

Presets for pcie\_a10\_hip\_0

Clear preset filters

Project

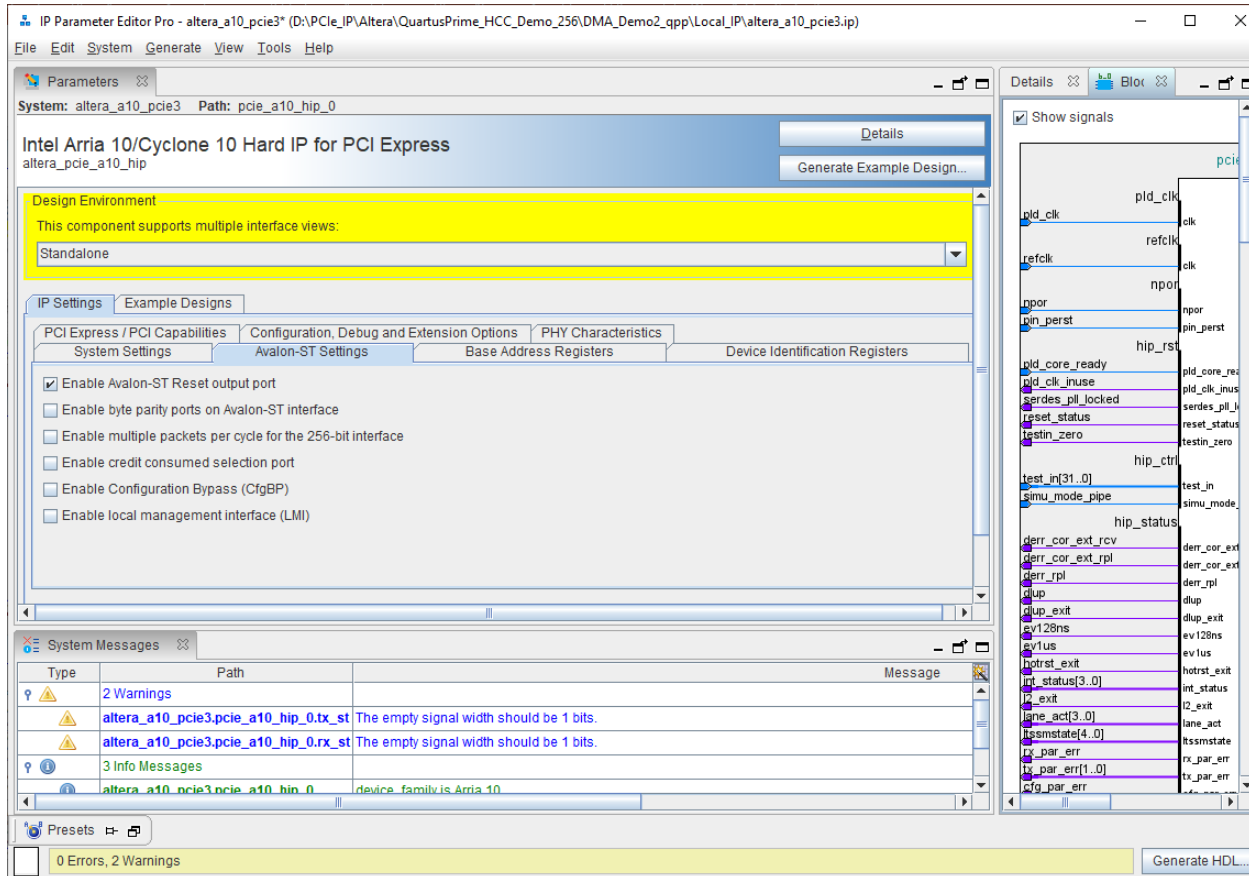
- Click New... to create a preset
- Library
- No presets for Intel Arria 10/Cyclone 10 Hard IP for PCI Express

Apply Update... Delete New...

Generate HDL...

# Arria 10 – Configuring the HIP

GUI of the Arria 10 HIP:



Enable only the Avalon-ST Reset output port. Never turn on „Enable multipackets per cycle for the 256-Bit Interface“

## Arria 10 – Configuring the HIP

---

### Important when you configure the Arria 10 HIP core:

- Set the RX Buffer credit allocation to „Low“ when the FPGA uses DMA Read. In all other cases, set this to „Balanced“
- You may turn on „Enable configuration via the PCIe link“ also known as CVP

If you turn on CVP make sure, that the PCIe 100 MHz reference clock is connected to `sys_clk_ip` of the Smartlogic IP core and that a free running 125 MHz clock is connected to `sys_clk_in` of the Smartlogic IP core. If you do not work with CVP, connect the 100 MHz clock to both `sys_clk_ip` and `sys_clk_in`

- For Signal Integrity measurements, you may turn on ADME to use the transceiver toolkit
- Check „slot clock configuration“, if you are operating with a common reference clock between root complex and FPGA endpoint
- Set the Device „Identification Registers“ and the „Base Address Registers“ according to your needs
- Choose „maximum payload size“ according to the capabilities of your root complex
- The Arria 10 PCIe HardIP does not support PCIe multi-function

A demo design, where the Arria 10 HIP is correctly setup is available from Smartlogic.

**Make sure to configure the following parameters of the Smartlogic IP core in the same way as you entered them into the HIP GUI:**

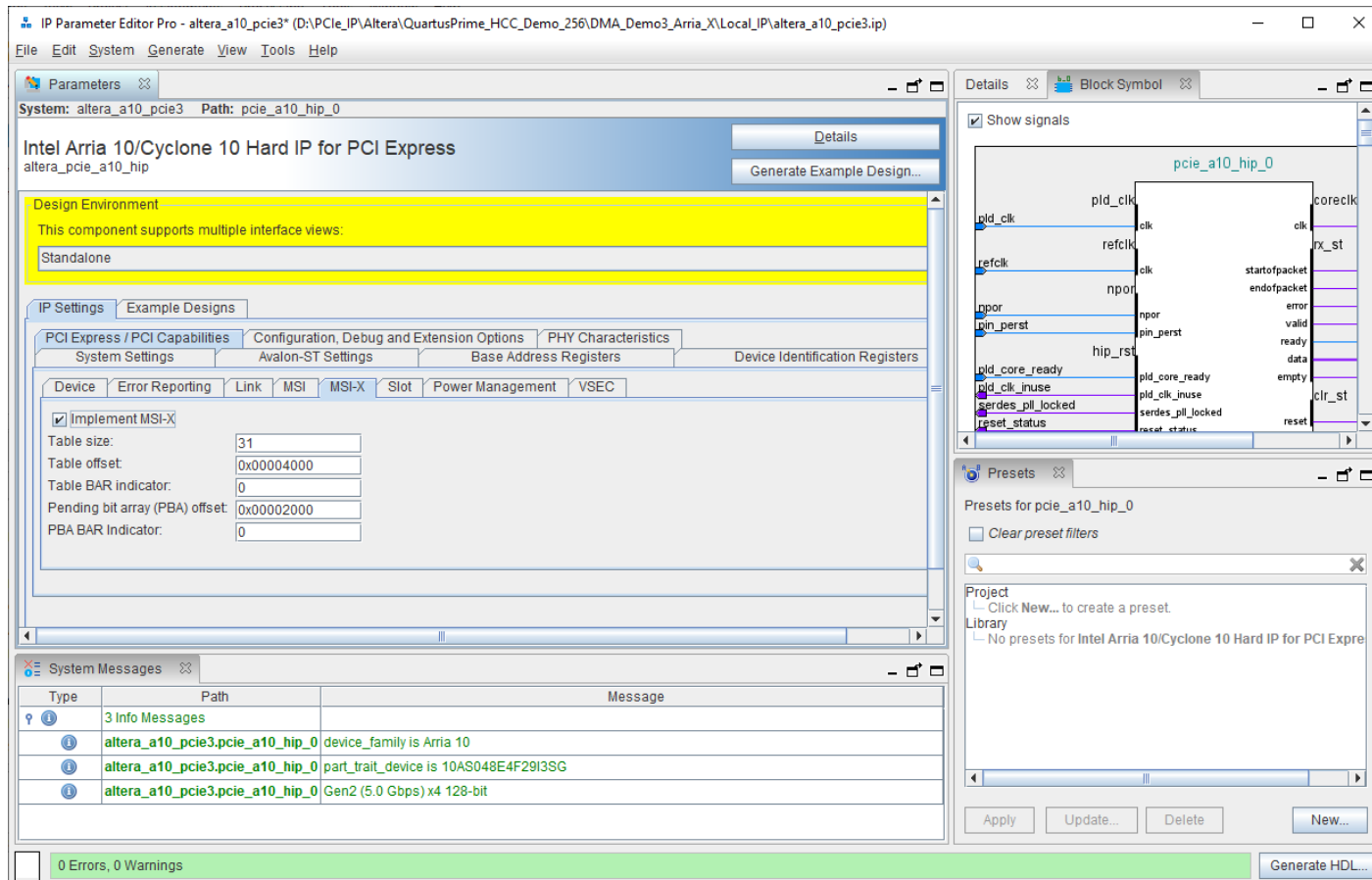
pcie\_ep\_config\_pkg.vhd

Parameter	Comment
PCle_synth_Core_Type_c	„AX“
Enable_SLOT_CLOCK_C PCIE_MSI_CAP_MULTIMSGCAP_C PCle_BAR<x>_C PCle_CLASS_CODE_C PCle_VEN_ID_C PCle_DEV_ID_C PCle_SUBSYS_VEN_ID_C PCle_SUBSYS_DEV_ID_C PCle_link_cap_max_link_speed_c PCle_link_cap_max_link_width_c	Set these constants in the same way as you entered them in the GUI.

The Avalon ST data bitwidth has to be entered according to the following table:

PCIe link speed / link width	Avalon ST bit width
<b>G3-X8, G3-X4, G2-X8</b>	<b>256</b>
<b>G1-X8, G2-X4, G3-X2</b>	<b>128</b>
<b>G1-X1, G1-X2, G1-X4, G2-X1, G2-X2, G3-X1</b>	<b>64</b>

GUI of the Arria 10 HIP : Example how to activate 32 MSI-X interrupts :



Currently only 32 MSI-X interrupts are supported, please do not try to enter other values.

Also make sure to set the generic/parameter use\_msix\_g to 1 !

---

For Stratix 10 FPGAs one IP catalog core has to be generated and added to the design:

1. The „Avalon-ST Intel Stratix 10 Hard IP for PCI Express“ IP core with the IP variation filename „altera\_S10\_pcie3.v“

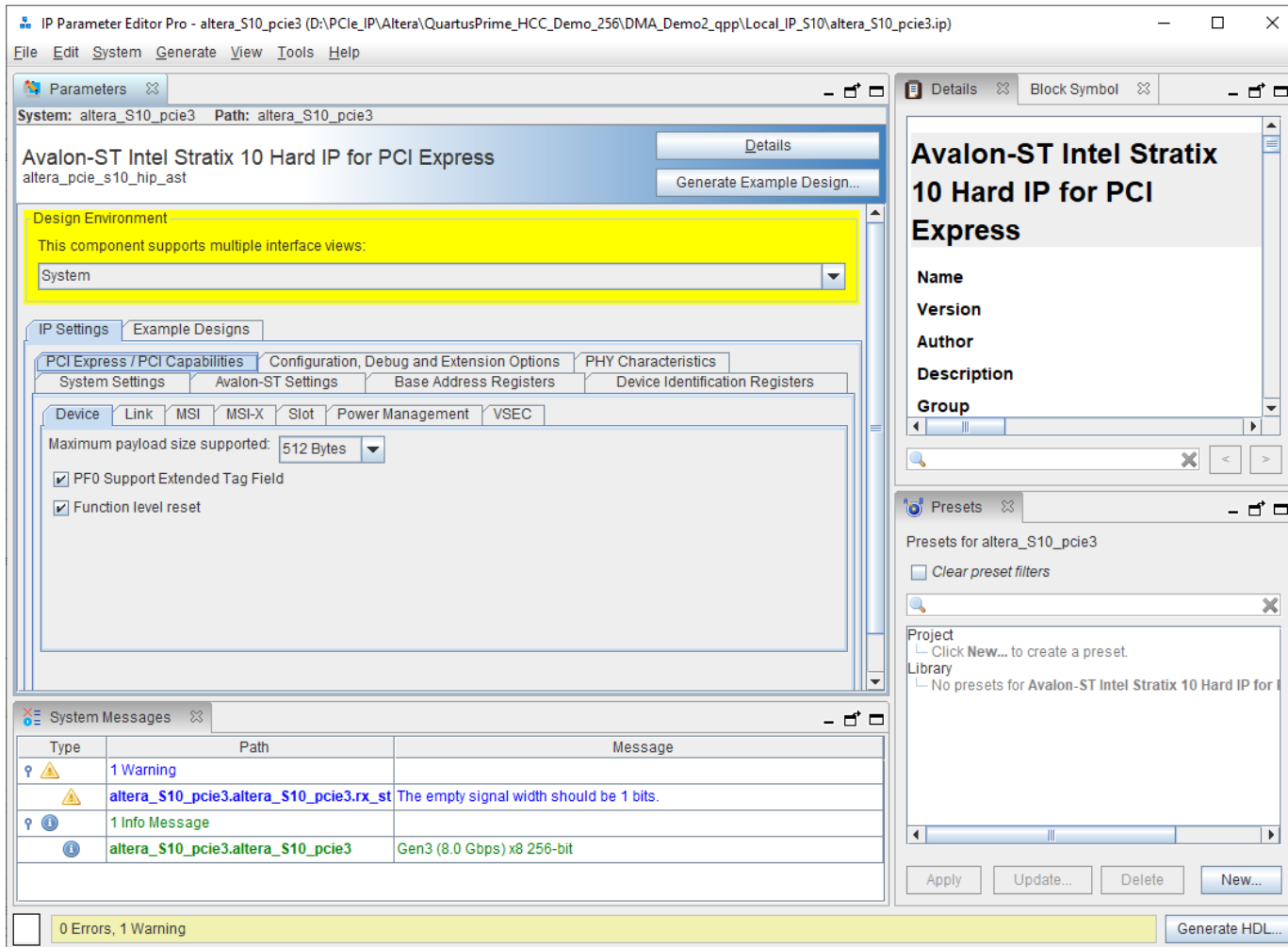
Make sure that your project contains the following flow specific files:

- The file „pcie\_core\_connection\_ipi.vhd“ has to be added to the project instead of pcie\_core\_connection.vhd
- The IP Core wrapper File „altera\_s10\_pcie3\_wrapper.v“ has to be added to the project



# Stratix 10 – Configuring the HIP

## GUI of the Stratix 10 HIP:



## Stratix 10 – Configuring the HIP

---

### Important when you configure the Stratix 10 HIP core:

- You may turn on „Enable configuration via the PCIe link“ also known as CVP

If you turn on CVP make sure, that the PCIe 100 MHz reference clock is connected to `sys_clk_ip` of the Smartlogic IP core and that a free running 125 MHz clock is connected to `sys_clk_in` of the Smartlogic IP core. If you do not work with CVP, connect the 100 MHz clock to both `sys_clk_ip` and `sys_clk_in`

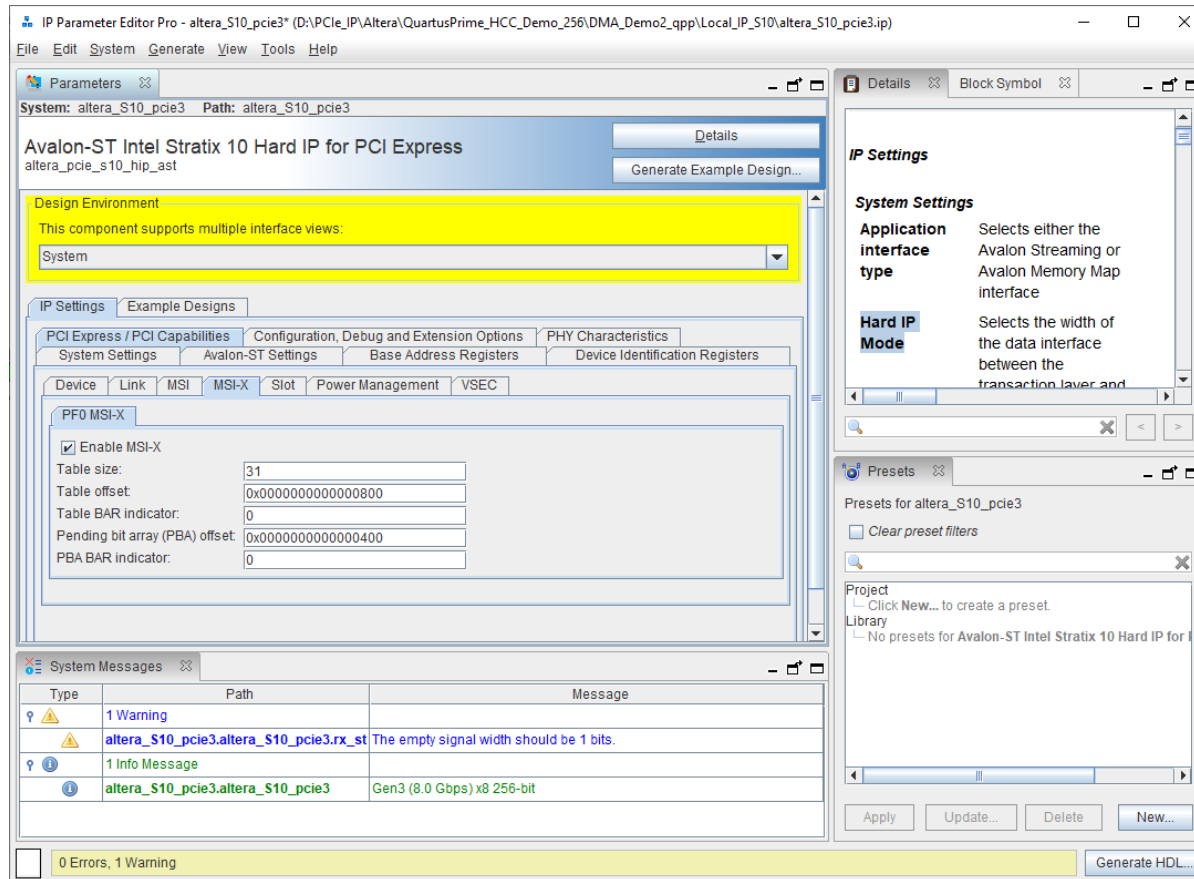
- For Signal Integrity measurements, you may turn on ADME to use the transceiver toolkit
- Check „slot clock configuration“, if you are operating with a common reference clock between root complex and FPGA endpoint
- Set the Device „Identification Registers“ and the „Base Address Registers“ according to your needs
- Choose „maximum payload size“ according to the capabilities of your root complex
- In case you are working with DMA Read you can check the „PF0 Support Extended Tag Field“ to improve performance. However this uses more BRAM resources
- multifunction is only supported for Stratix H-Tile devices

A Demodesign, where the Stratix 10 HIP is correctly setup is available from Smartlogic as a reference.

The Avalon ST data bit width has to be entered according to the following table:

PCIe Linkspeed / Linkwidth	Avalon ST bitwidth
<b>G3-X8, G3-X4, G2-X8, G2-X16</b>	<b>256</b>
<b>G1-X8, G2-X4, G3-X2</b>	<b>128</b>
<b>G1-X1, G1-X2, G1-X4, G2-X1, G2-X2, G3-X1</b>	<b>64</b>

GUI of the Stratix 10 HIP : Example how to activate 32 MSI-X interrupts :



Currently only 32 MSI-X interrupts are supported, please do not try to enter other values.

Also make sure to set the generic/parameter use\_msix\_g to 1 !

**Make sure to configure the following parameters of the Smartlogic IP core in the same way as you entered them into the HIP GUI:**

pcie\_ep\_config\_pkg.vhd

Parameter	Comment
PCle_synth_Core_Type_c	„SX“
Enable_SLOT_CLOCK_C PCIE_MSI_CAP_MULTIMSGCAP_C PCle_BAR<x>_C PCle_CLASS_CODE_C PCle_VEN_ID_C PCle_DEV_ID_C PCle_SUBSYS_VEN_ID_C PCle_SUBSYS_DEV_ID_C PCle_link_cap_max_link_speed_c PCle_link_cap_max_link_width_c	Set these constants in the same way as you entered them in the GUI.