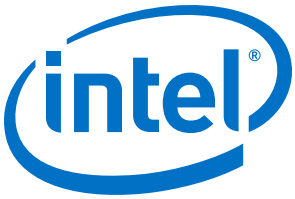


AN 728: I/O PLL Reconfiguration and Dynamic Phase Shift for Intel® Arria® 10 and Intel Cyclone® 10 GX Devices

Subscribe
Send Feedback

AN-728 | 2018.06.15
Latest document on the web: PDF | HTML

Contents

I/O PLL Reconfiguration and Dynamic Phase Shift for Intel® Arria® 10 and Intel® Cyclone® 10 GX Devices

You can use Intel® Arria® 10 and Intel Cyclone® 10 GX devices to implement phase-locked loop (PLL) reconfiguration and dynamic phase shift for I/O PLLs.

Intel Arria 10 and Intel Cyclone 10 GX I/O PLL supports dynamic reconfiguration when the device is in user mode. With the dynamic reconfiguration feature, you can reconfigure I/O PLL settings in real time. You can change the divide settings of the PLL counters and the PLL bandwidth settings (loop filter setting and charge pump setting) through an Avalon® Memory-Mapped (Avalon-MM) interface in the PLL Reconfig Intel FPGA IP core, without the need to reconfigure the entire FPGA. Intel Arria 10 and Intel Cyclone 10 GX I/O PLL uses divide counters (`N`, `M`, and `C` counters) and a voltage-controlled oscillator (VCO) to synthesize the desired phase and frequency output.

You can perform dynamic reconfiguration using one of the following methods:

- Memory Initialization File (`.mif`) streaming reconfiguration
  - Allows I/O PLL reconfiguration using predefined settings saved in an on-chip ROM. You can store many unique PLL configurations in a single ROM.
  - The `.mif` file is generated automatically by the IOPLL Intel FPGA IP core. Using the generated `.mif` file during `.mif` streaming reconfiguration ensures the legality of the new configuration.
  - Intel recommends using this reconfiguration method.
- Reconfiguration of individual PLL settings
  - Supports `N`, `M`, and `C` counters reconfiguration.
  - Supports bandwidth setting changes of the loop filter configuration. You must reconfigure the charge pump current setting and loop filter resistance setting for stable operation of the PLL if the `M` counter value is changed.
  - Supports both read and write operations.
  - This method of reconfiguration is for advanced users. You must ensure the reconfigured PLL settings are within the legal range.

With the dynamic phase shift feature of the I/O PLL, you can modify the phase of the PLL output clocks in real time. You can adjust the phase in increments of 1/8 of the VCO period.

Intel Corporation. All rights reserved. Intel, the Intel logo, Altera, Arria, Cyclone, Enpirion, MAX, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.
*Other names and brands may be claimed as the property of others.

ISO 9001:2008 Registered

You can perform dynamic phase shift using one of the following methods:

- Direct access to the dynamic phase shift ports in the IOPLL IP core
  - Supports both shift up and shift down operations.
  - Supports up to seven phase shift steps in a single operation.
- Dynamic phase shift via the PLL Reconfig IP core
  - Available via `.mif` streaming reconfiguration or via reconfiguration of individual PLL settings.
  - Supports only write operation.

Related Information

- Reconfiguration Interface and Dynamic Reconfiguration chapter, Intel Arria 10 Transceiver PHY User Guide
  Provides more information about Intel Arria 10 fractional PLL (fPLL) reconfiguration and dynamic phase shift. The Intel Arria 10 fPLL reconfiguration is different from I/O PLL.
- Clock Networks and PLLs in Intel Arria 10 Devices chapter, Intel Arria 10 Core Fabric and General Purpose I/Os Handbook
  Provides more information about PLL counters.
- Reconfiguration Interface and Dynamic Reconfiguration chapter, Intel Cyclone 10 GX Transceiver PHY User Guide
  Provides more information about Intel Cyclone 10 GX fPLL reconfiguration and dynamic phase shift. The Intel Cyclone 10 GX fPLL reconfiguration is different from I/O PLL.
- Clock Networks and PLLs in Intel Cyclone 10 GX Devices chapter, Intel Cyclone 10 GX Core Fabric and General Purpose I/Os Handbook
  Provides more information about PLL counters.
- Design Example 1: I/O PLL Reconfiguration
- Design Example 2: .mif Streaming Reconfiguration
- Design Example 3: I/O PLL Dynamic Phase Shift

Implementing I/O PLL Reconfiguration in the PLL Reconfig IP Core

You can enable PLL reconfiguration circuitry for I/O PLL through the Avalon-MM interface in the PLL Reconfig IP core.

Avalon-MM Interface Ports in the PLL Reconfig IP Core

Table 1. Avalon-MM Interface Ports in the PLL Reconfig IP Core

Port	Direction	Description
`mgmt_clk`	Input	Dynamic reconfiguration clock that drives the PLL Reconfig IP core. This port must be connected to a valid clock source. The maximum input clock frequency is 100 MHz. This clock can be an independent clock source.
`mgmt_reset`	Input	Active high signal. This port is synchronous with `mgmt_clk`. Assert this reset input to clear all the data in the PLL Reconfig IP core.

continued...

Port	Direction	Description
mgmt_waitrequest	Output	This port goes high when PLL reconfiguration process started and remains high during PLL reconfiguration. After PLL reconfiguration process completed, this port goes low.
mgmt_read	Input	Active high signal. Asserts to indicate a read operation.
mgmt_write	Input	Active high signal. Asserts to indicate a write operation.
mgmt_readdata[31..0]	Output	Reads data from this port when mgmt_read signal is asserted.
mgmt_address[8..0]	Input	Specifies the address of the data bus for a read or write operation.
mgmt_writedata[31..0]	Input	Writes data to this port when mgmt_write signal is asserted.
mgmt_byteenable[3..0]	Input	Optional. Permits write operation to the PLL Reconfig IP core from an Avalon-MM interface with data bus width wider than 32 bits.
reconfig_from_pll[63..0]	Input	Bus that connects to reconfig_from_pll[63..0] bus in the IOPLL IP core.
reconfig_to_pll[63..0]	Output	Bus that connects to reconfig_to_pll[63..0] bus in the IOPLL IP core.

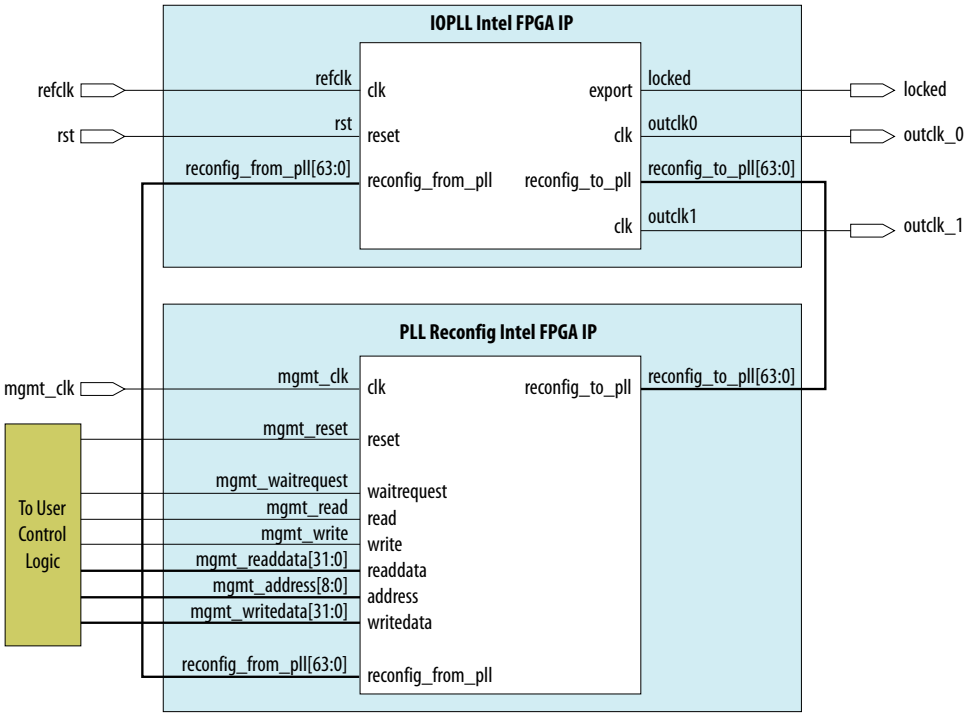
Connecting the IOPLL and PLL Reconfig IP Cores

To connect the IOPLL and PLL Reconfig IP cores in your design, follow these steps:

1. Connect the reconfig_to_pll[63..0] bus on the PLL Reconfig IP core to the reconfig_to_pll[63..0] bus on the IOPLL IP core.
2. Connect the reconfig_from_pll[63..0] bus on the PLL Reconfig IP core to the reconfig_from_pll[63..0] bus on the IOPLL IP core.
3. Connect the mgmt_clk port to a valid clock source.
4. Connect the mgmt_reset port, mgmt_waitrequest port, mgmt_read port, mgmt_write port, mgmt_readdata[31..0] bus, and mgmt_writedata[31..0] bus to user control logic to perform read and write operations.

Connectivity between the IOPLL and PLL Reconfig IP Cores

Figure 1. Connectivity between the IOPLL and PLL Reconfig IP Cores in the Intel Quartus® Prime Software



I/O PLL Reconfiguration Write Operation

To perform a write operation for I/O PLL reconfiguration in the PLL Reconfig IP core, follow these steps:

1. Set the address bus value for `mgmt_address`, and the data bus value for `mgmt_writedata`. To enable write operation for I/O PLL reconfiguration, assert the `mgmt_write` signal for one `mgmt_clk` cycle.
2. Repeat step 1 to set the values for up to eight sets of address bus and data bus.
3. Set the start address (9'b000000000) for `mgmt_address`, and any value for `mgmt_writedata`. To start the write operation for I/O PLL reconfiguration, assert the `mgmt_write` signal for one `mgmt_clk` cycle.
4. After the write operation is complete, the `mgmt_waitrequest` signal is de-asserted.

Each dynamic reconfiguration command (address-data pair) can be of one of the three types:

- Counter setting reconfiguration
- Bandwidth setting reconfiguration
- Dynamic phase shift

You can issue up to eight dynamic reconfiguration commands before triggering reconfiguration by writing to the start address. Issuing more than eight commands causes the internal reconfiguration FIFO to overflow.

Write Operation for .mif Streaming Reconfiguration

The dynamic reconfiguration commands can also be stored in a `.mif` file for `.mif` streaming reconfiguration. To enable `.mif` streaming in the PLL Reconfig IP core, select the appropriate checkbox in the parameter editor and provide the path to the `.mif` file before generating your IP core.

To perform `.mif` streaming reconfiguration, follow these steps:

1. Set the start `.mif` address (9'b000010000) for `mgmt_address` and the base address within the ROM for `mgmt_writedata`. To start the `.mif` streaming operation of I/O PLL reconfiguration, assert the `mgmt_write` signal for one `mgmt_clk` cycle.
2. After the reconfiguration is complete, the `mgmt_waitrequest` signal is de-asserted.

You should not write to the start address (9'b0) in the `.mif` file.

I/O PLL Reconfiguration Read Operation

To perform a read operation for I/O PLL reconfiguration in the PLL Reconfig IP core, follow these steps:

1. Set the address bus value for `mgmt_address`. To enable and start the read operation for I/O PLL reconfiguration, assert `mgmt_read` for one `mgmt_clk` cycle.
2. After the read operation is complete, the `mgmt_waitrequest` signal is de-asserted. At the same time, read the data available from `mgmt_readdata` port.

With I/O PLL reconfiguration read operation, you can read the current I/O PLL settings. This operation is only supported for counter settings and bandwidth settings. Each read operation retrieves data from a single address. An I/O PLL reconfiguration read operation requires at least three `mgmt_clk` cycles of latency.

Address Bus and Data Bus Settings

Assign a value of "0" for all the unused bits in the address bus and the data bus during write and read operations.

Address Bus and Data Bus Setting for Counter Setting Reconfiguration

Table 2. Address Bus and Data Bus Bit Setting for Counter Setting Reconfiguration using the PLL Reconfig IP Core

Counter Name	Address Bus Bit Setting (Binary)	Data bus bit setting (Binary)
M	9'b010010000	• Data[7..0] = low_div • Data[15..8] = high_div • total_div = high_div + low_div • Data[16] = bypass enable — When Data[16] = 1, bypass is enabled. The selected counter is bypassed with counter division value=1. • Data[17] = odd division — When Data[17] = 0, odd division is disabled. The selected counter duty cycle = high_div/total_div. — When Data[17] = 1, odd division is enabled. The selected counter duty cycle = (high_div - 0.5)/ total_div.
N	9'b010100000	
C0	9'b011000000	
C1	9'b011000001	
C2	9'b011000010	
C3	9'b011000011	
C4	9'b011000100	
C5	9'b011000101	
C6	9'b011000110	
C7	9'b011000111	
C8	9'b011001000	

Address Bus and Data Bus Setting for Dynamic Phase Shift

Table 3. Address Bus and Data Bus Bit Setting for Dynamic Phase Shift using the PLL Reconfig IP Core

Counter Name	Address Bus Bit Setting (Binary)	Data Bus Bit Setting (Binary)
C0	9'b100000000	• Data[2..0] = number of phase shift — Maximum of seven phase shifts per operation. Each phase shift step is equal to 1/8 of I/O PLL VCO period. • Data[3] = direction of phase shift — When Data[3] = 0, phase shift is in negative direction (shift down). — When Data[3] = 1, phase shift is in positive direction (shift up).
C1	9'b100000001	
C2	9'b100000010	
C3	9'b100000011	
C4	9'b100000100	
C5	9'b100000101	
C6	9'b100000110	
C7	9'b100000111	
C8	9'b100001000	
All C counters	9'b100001111	

Address Bus and Data Bus Setting for Bandwidth Setting Reconfiguration

Table 4. Address Bus and Data Bus Bit Setting for Bandwidth Setting Reconfiguration using the PLL Reconfig IP Core

Bandwidth Component	Address Bus Bit Setting (Binary)	Data Bus Bit Setting (Binary)
Loop filter setting	9'b001000000	Data[9..6]
Charge pump setting	9'b000100000	Data[5..0]

Data Bus Setting for Loop Filter and Charge Pump Settings

Table 5. Data Bus Setting for Loop Filter and Charge Pump Settings

M Counter Total Division Value	Low Bandwidth		Medium Bandwidth		High Bandwidth	
	Loop Filter Setting	Charge Pump Setting	Loop Filter Setting	Charge Pump Setting	Loop Filter Setting	Charge Pump Setting
	Data[9..6]	Data[5..0]	Data[9..6]	Data[5..0]	Data[9..6]	Data[5..0]
4–5	4'b0010	6'b001011	4'b0010	6'b011000	4'b0010	6'b001100
6–7	4'b0011	6'b010000	4'b0011	6'b001011	4'b0011	6'b011000
8–15	4'b0011	6'b010000	4'b0011	6'b011000	4'b0011	6'b100000
16–23	4'b0011	6'b001011	4'b0011	6'b001100	4'b0011	6'b001101
24–43	4'b0100	6'b010000	4'b0100	6'b011000	4'b0100	6'b100000
44–64	4'b0101	6'b010000	4'b0101	6'b011000	4'b0101	6'b100000
64–124	4'b0101	6'b011000	4'b0101	6'b100000	4'b0101	6'b101000
>125	4'b0110	6'b011000	4'b0110	6'b100000	4'b0110	6'b101000

Address Bus and Data Bus Setting for .mif Streaming Reconfiguration

Table 6. Address Bus and Data Bus Bit Setting for .mif Streaming Reconfiguration using the PLL Reconfig IP Core

Address Bus Bit Setting (Binary)	Data Bus Bit Setting (Binary)
9'b000010000	Data[8..0] .mif base address in ROM that is desired to use for I/O PLL reconfiguration.

.mif Streaming Reconfiguration

The .mif file stores the commands as ROM in M20K embedded memory block. You can send these commands to the PLL Reconfig IP core via .mif streaming reconfiguration.

Each .mif streaming reconfiguration must be indicated with the Start of .mif (SOM) and End of .mif (EOM) operation codes.

The .mif streaming reconfiguration starts at the .mif base address that you specify. The PLL Reconfig IP core reads the .mif file contents until it encounters a SOM operation code (Opcode). From this point, the .mif streaming reconfiguration simulates the commands that the user logic could send to the PLL Reconfig IP core. These commands are address-data pairs in the .mif file specifying the setting to be reconfigured to the new value. The address is stored as an Opcode in the lower nine bits of each entry word, while data is stored in the lower 18 bits of each entry word. The bit settings for address-data pairs in the .mif file is the same as the address bus and data bus bit setting for counter setting reconfiguration and bandwidth setting reconfiguration. The .mif streaming reconfiguration ends when the EOM Opcode is encountered.

You can save multiple I/O PLL configurations in a .mif file if you mark the SOM and EOM Opcodes appropriately. The PLL Reconfig IP core reads the setting from ROM in M20K embedded memory block, which has default address width = 9 bits and data

width = 32 bits with total of 512 words. These sizes can change as parameters are passed to the top-level module. For `.mif` streaming reconfiguration, data width must be 32 bits to match the PLL Reconfig IP core.

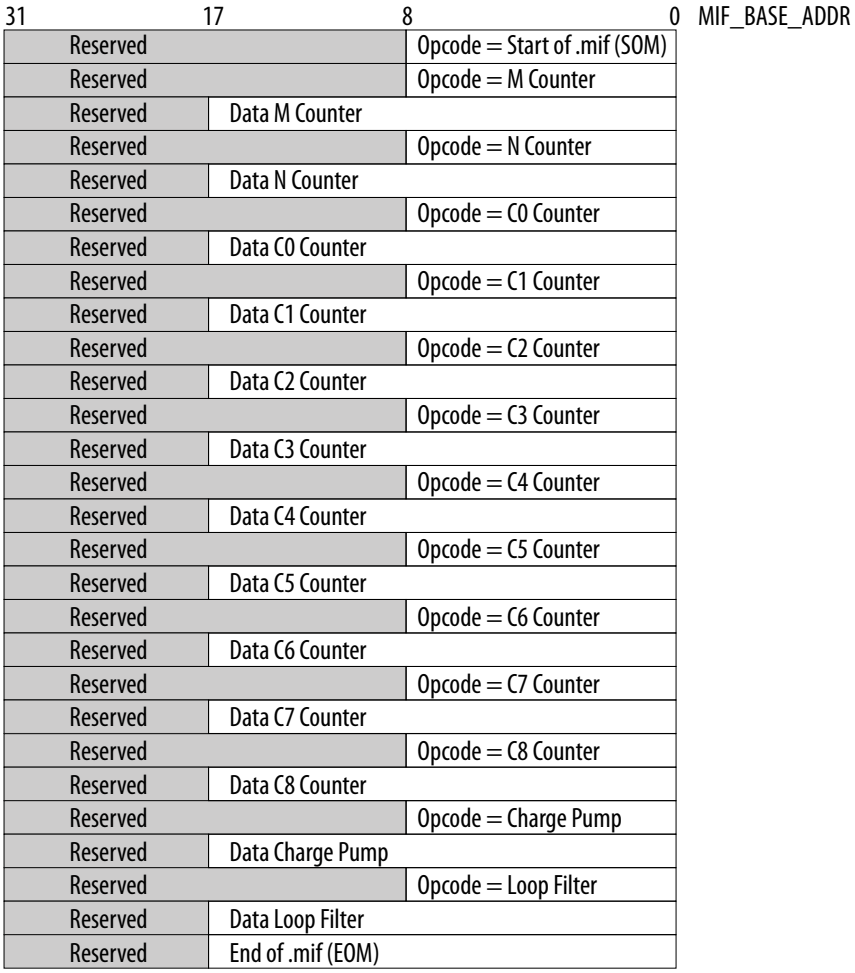
Intel recommends using a `.mif` file generated from the IOPLL IP core with the desired reconfiguration setting. The generated `.mif` file stores the entire I/O PLL profile.

Related Information

- Address Bus and Data Bus Setting for Counter Setting Reconfiguration on page 8
- Address Bus and Data Bus Setting for Bandwidth Setting Reconfiguration on page 8
- Data Bus Setting for Loop Filter and Charge Pump Settings on page 9

.mif File Format

Figure 2. Single .mif File Format for .mif Streaming Reconfiguration



.mif Streaming Reconfiguration Operation Codes

Table 7. .mif Streaming Reconfiguration Operation Codes

Operation Name	Operation Code	Description
Start of `.mif` (SOM)	9'b000011110	Indicates start of I/O PLL reconfiguration.
End of `.mif` (EOM)	9'b000011111	Indicates end of I/O PLL reconfiguration.

Implementing I/O PLL Dynamic Phase Shift in the IOPLL IP Core

You can use the IOPLL IP core to perform phase shifting directly through the dynamic phase shift ports.

Dynamic Phase Shift Ports in the IOPLL IP Core

Figure 3. Dynamic Phase Shift Port Ports in the IOPLL IP Core

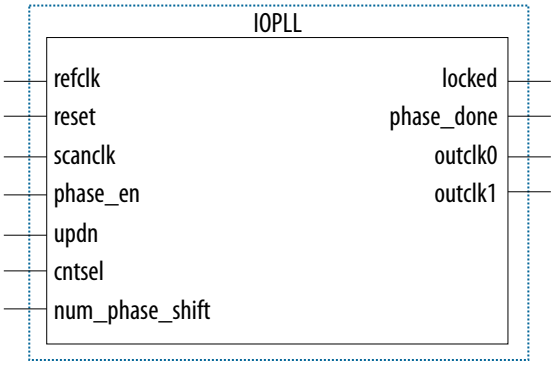


Table 8. Dynamic Phase Shift Ports in the IOPLL IP Core

Port	Direction	Description
`scanclk`	Input	Dynamic phase shift clock that drives the IOPLL IP core dynamic phase shift operation. This port must be connected to a valid clock source. The maximum input clock frequency is 100 MHz.
`phase_en`	Input	Active high signal. Asserts to start the dynamic phase shift operation.
`updn`	Input	Determines the direction of dynamic phase shift. When `updn` = 0, phase shift is in negative direction (shift down). When `updn` = 1, phase shift is in positive direction (shift up).
`cntsel[4..0]`	Input	Determines the counter to be selected to perform dynamic phase shift operation.

Counter Name	`cntsel[4..0]` (Binary)
`C0`	5'b00000
`C1`	5'b00001
`C2`	5'b00010
`C3`	5'b00011
`C4`	5'b00100

continued...

<table>
<tr><th>Port</th><th>Direction</th><th>Description</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><th>Counter Name</th><th>cntsel[4..0] (Binary)</th></tr>
<tr><td>C5</td><td>5'b00101</td></tr>
<tr><td>C6</td><td>5'b00110</td></tr>
<tr><td>C7</td><td>5'b00111</td></tr>
<tr><td>C8</td><td>5'b01000</td></tr>
<tr><td>All C counters</td><td>5'b01111</td></tr>
</table>
</td></tr>
<tr><td>num_phase_shift[2..0]</td><td>Input</td><td>Determines the number of phase shifts per dynamic phase shift operation. Up to seven phase shifts per operation are possible. Each phase shift step is equal to 1/8 of I/O PLL VCO period.</td></tr>
<tr><td>phase_done</td><td>Output</td><td>The IOPLL IP core drives this port high for one scanclk cycle after dynamic phase shift operation is complete.</td></tr>
</table>

I/O PLL Dynamic Phase Shift Operation

To perform dynamic phase shift operation for an I/O PLL in the IOPLL IP core, follow these steps:

1. Set the value for updn, cntsel[4..0], and num_phase_shift[2..0] ports.
2. Assert phase_en port for at least two scanclk cycles.

Each phase_en pulse indicates one dynamic phase shift operation. The phase_done output goes low to indicate that dynamic phase shift is in progress. You can only assert the phase_en signal after the phase_done signal goes from low to high.

The updn, cntsel[4..0], and num_phase_shift[2..0] ports are synchronous to the scanclk cycle.

When the phase_done signal transitions from high to low, the phase_done signal is synchronous to the rising edge of the scanclk signal. The transition from low to high is asynchronous to the scanclk signal.

Depending on the VCO and scanclk frequency, the low time of the phase_done signal may be greater than or less than one scanclk cycle.

Design Considerations

Resetting the PLL

- When changing the M counter, N counter, or loop filter settings, the I/O PLL may lose and regain lock. To maintain the appropriate phase relationship between the reference clock and output clocks, assert the `areset` signal to reset the PLL after reconfiguration is complete. Intel recommends always resetting the PLL after any reconfiguration operation to the M counter, N counter, or loop filter settings.
- When changing the C counter settings, you may lose the expected phase relationship between the C counters. Assert the `areset` signal after reconfiguration is complete to restore the expected phase relationship. Reset is not required if the phase relationships are not important to your application.
- Resetting the PLL does not modify the counter or loop filter settings. However, resetting the PLL undoes any dynamic phase shift operations that were performed. After the PLL is reset, the phase shift on the C counters is restored to the originally programmed settings.

Configuration Constraints

The I/O PLL configuration must obey the following constraints:

- The phase frequency detector (PFD) and VCO each have a legal frequency range of operation.
- The loop filter settings must be appropriate for the M counter value and user-selected bandwidth mode.

If any of these configuration constraints are violated, the I/O PLL may fail to lock or may exhibit poor jitter performance.

If you reconfigure the PLL using `.mif` streaming, the IOPLL IP core always produce legal PLL configurations in the auto-generated `.mif` file.

You must ensure that the PLL settings are legal if you apply the PLL reconfiguration write operations directly. The IOPLL IP core parameter editor provides several methods to identify the legal PLL configurations and to explore the combination of legal configurations.

Timing Closure

- Reconfiguring a PLL's counter and loop filter settings changes both the output frequency and the clock uncertainty of that PLL. Dynamic phase shift only affects the output clock phase.
- The Timing Analyzer in the Intel Quartus® Prime software performs timing analysis for the initial PLL settings only. You must verify that your design closes timing after dynamic reconfiguration or dynamic phase shift.
- Intel recommends compiling PLL designs with each intended configuration setting to determine the variation in the clock with PLL settings.

Other Design Considerations

- I/O PLL reconfiguration interface supports a free running `mgmt_clk` signal. I/O PLL dynamic phase shift interface supports a free running `scanclk` signal. These interfaces eliminate the need to precisely control the start and stop of `mgmt_clk` and `scanclk` signals.
- Reconfiguration commands are queued in the PLL Reconfig IP core, and removed after they are complete. Assert a high pulse on the `mgmt_reset` signal to reset the IP core to its initial state, clearing any queued commands.
- Use caution when reconfiguring a PLL with a non-zero phase shift setting. Modifying the `M` counter or `N` counter settings does not change the relative phase shift (in percent), but alters the absolute phase shift (in picoseconds). Modifying the `C` counter settings does not change the absolute phase shift, but modifies the relative phase shift.
- When writing to the PLL Reconfig IP core using a Nios® processor's Avalon-MM master, use the default Nios word-addressing scheme. The Nios processor produces an 11-bit address. The Platform Designer automatically converts this 11-bit address into the appropriate 9-bit address as required by the PLL Reconfig IP core.

Related Information

I/O Phase-Locked Loop (IOPLL) Intel FPGA IP Core User Guide
Provides more information about PLL configurations methods.

Using the Design Examples

You must install the Intel Quartus Prime software version 15.0 or later. The software must be installed on a Windows* or Linux* computer that meets the Intel Quartus Prime software minimum requirements.

Design Example 1: I/O PLL Reconfiguration

The design example uses a 10AX115R2F40I2SGE2 device to demonstrate the implementation of the I/O PLL dynamic reconfiguration using the PLL Reconfig IP core. This design example consists of the IOPLL IP core, PLL Reconfig IP core, and In-System Sources & Probes Intel FPGA IP core.

The I/O PLL synthesizes two output clocks of 400 MHz with 0 ps phase shift and 200 MHz with 0 ps phase shift on counter `C0` output and counter `C1` output respectively at medium bandwidth. The input reference clock is 50 MHz.

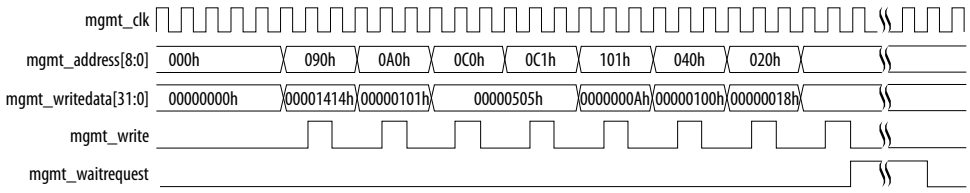
The PLL Reconfig IP core connects to a state machine to perform I/O PLL reconfiguration operation. A low pulse on the `reset_SM` input through the In-System Sources & Probes IP core triggers the I/O PLL reconfiguration operation. After I/O PLL reconfiguration operation is complete, the I/O PLL operates in the following configuration at medium bandwidth:

- 100 MHz with 0 ps phase shift on counter `C0` output
- 100 MHz with 312 ps phase shift on counter `C1` output

To run the test with the design example, perform these steps:

1. Download and restore the `an728-iopll-reconfig-general.qar` file.
2. If necessary, change the device and pin assignments (`refclk`, `c0_out`, `c1_out`, and `locked` pins) of the design example to match your hardware.
3. Recompile the design example. Ensure that the design example does not contain any timing violation after recompilation.
4. Open the `AN.spf` and program the device with `test.sof`.
5. Assert a high pulse on `reset_reconfig` signal to reset the PLL Reconfig IP core. Then assert a high pulse on `reset_SM` signal to start the I/O PLL dynamic reconfiguration operation.

Figure 4. Waveform Example for I/O PLL Reconfiguration Design Example

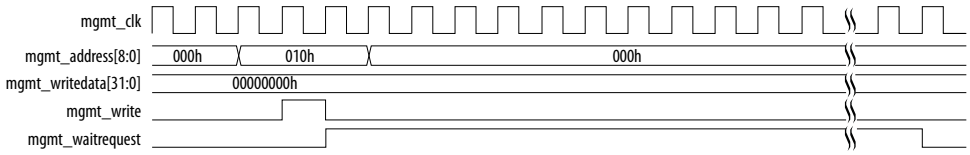


Design Example 2: .mif Streaming Reconfiguration

This design example is similar to Design Example 1, except that the dynamic reconfiguration commands (address-data pair) are stored in the `AN.mif` file.

To run this design example, follow the steps in Design Example 1, except Step 1. Download and restore the `an728-iopll-reconfig-mif-streaming.qar` file for this design example.

Figure 5. Waveform Example for .mif Streaming Reconfiguration Design Example



Related Information

Design Example 1: I/O PLL Reconfiguration on page 14

Design Example 3: I/O PLL Dynamic Phase Shift

This design example uses the same device and pin assignments as in Design Example 1 and Design Example 2. This design example demonstrates the implementation of the I/O PLL dynamic phase shift in the IOPLL IP core.

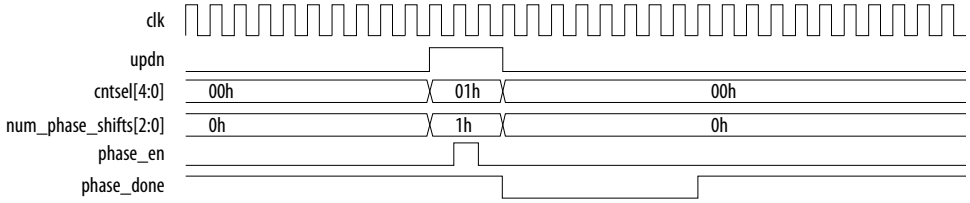
The I/O PLL synthesizes two output clocks of 200 MHz with 0 ps phase shift on counter `C0` output and counter `C1` output at medium bandwidth. The input reference clock is 50 MHz.

The dynamic phase shift ports of the IOPLL IP core connect to a state machine to perform I/O PLL dynamic phase shift operation. A low pulse on the `reset_SM` input through the In-System Sources & Probes IP core triggers the I/O PLL dynamic phase shift operation. After I/O PLL dynamic phase shift operation is complete, counter `C1` is phase shifted 208 ps for one positive phase shift step.

To run the test with the design example, perform these steps:

1. Download and restore the `an728-iopll-dynamic_phase_shift.qar` file.
2. If necessary, change the device and pin assignments (`refclk`, `c0_out`, `c1_out`, and `locked` pins) of the design example to match your hardware.
3. Recompile the design example. Ensure that the design example does not contain any timing violation after recompilation.
4. Open the `AN.spf` and program the device with `test.sof`.
5. Assert a high pulse on `reset_SM` signal to start the I/O PLL dynamic phase shift operation.

Figure 6. Waveform Example for I/O PLL Dynamic Phase Shift Design Example



Related Information

- Design Example 1: I/O PLL Reconfiguration on page 14
- Design Example 2: .mif Streaming Reconfiguration on page 15

Document Revision History for AN 728: I/O PLL Reconfiguration and Dynamic Phase Shift for Intel Arria 10 and Intel Cyclone 10 GX Devices

Document Version	Changes
2018.06.15	• Added support for Intel Cyclone 10 GX devices. • Updated the description for the `mgmt_reset` port in the *Avalon-MM Interface Ports in the PLL Reconfig IP Core* table. • Renamed the following IP cores as per Intel rebranding: — Renamed Altera PLL Reconfig IP core to PLL Reconfig Intel FPGA IP core. — Renamed Altera IOPLL IP core to IOPLL Intel FPGA IP core. — Renamed Altera In-System Sources & Probe IP core to In-System Sources & Probes Intel FPGA IP core. • Updated the following terms: — Updated Quartus Prime to Intel Quartus Prime. — Updated TimeQuest Timing Analyzer to Timing Analyzer. — Updated Qsys to Platform Designer.

Date	Version	Changes
March 2017	2017.03.15	Rebranded as Intel.
May 2016	2016.05.05	• Updated the definition for `mgmt_reset` in Avalon-MM Interface Ports in Altera PLL Reconfig IP Core table. • Changed instances of *Quartus II* to *Quartus Prime*.
June 2015	2015.06.12	• Added description to the introduction section that you must ensure the reconfigured PLL settings are within the legal range. • Added description to the I/O PLL Reconfiguration Read Operation topic: An I/O PLL reconfiguration read operation requires at least three `mgmt_clk` cycles of latency. • Restructured and updated Design Considerations section. • Added Design Examples section.
January 2015	2015.01.23	Initial release.