Enhanced Constraint Flow for SmartFusion® 2, IGLOO® 2, and RTG4™ Devices

Introduction

Block Flow is a bottom-up design methodology that enables you to use design blocks (“components” in generic terms) as building blocks for your top-level design. These building blocks may have already completed layout and been optimized for timing and power performance for a specific Microchip device. Using these blocks as part of your top-level design can cut down design time as well as improve timing and power performance. Using blocks has the following advantages:

- Help you focus on the timing of critical blocks and ensure the timing across the blocks meets requirements before proceeding to integrate your blocks at the top-level.
- Helps re-using blocks without re-optimization for timing closure; changes in other blocks have no impacts.
- Helps re-using blocks in multiple designs.
- Allows shorter verification time.

Blocks can be used in the following scenarios:

- You have multiple team members working on different parts of the same design.
- The design is congested (uses 90% or more of the resources on a given die).
- You have difficulty in meeting timing by doing the design in its entirety. Blocks enable you to compartmentalize the design and optimize sections before you optimize the entire design.
- You want to re-use some elements of your design.
- You want to use the identical elements multiple times in a single design.
- You want to make small changes in your design and expect to keep most of the design unchanged with ensured performance.

You cannot use blocks with all families, they are family and die specific. If your block has I/Os, it is also package specific.

Important: The enhanced block flow user guide is applicable for v 11.8.

Features

The key features of blocks are:

- A block can be synthesized, simulated, and place-and-route the same way as a regular design.
- The place-and-route of the block can be locked to ensure repeatable performance.
- Performance, placement, and routing can be fixed absolutely; however these rules can be relaxed gradually, if necessary, to ensure that you can integrate the block into your top-level project.
- The block flow supports the following HDLs:
  - Verilog
  - VHDL

- The block flow supports SynplifyPro synthesis tool.
- Nested blocks (blocks instantiated inside other blocks) are supported. When publishing, only one file will be published that contains all the required information (including the nested block).

Table of Contents

1. Creating Blocks - Options and Settings

Blocks can be created for a new as well as for the existing projects-

New Project
To enable **Block Creation** for a new project, go to **Project > New Project**. Check the **Enable Block Creation** check box. Select the **Enhanced Constraint Flow** for the new project.

Existing Project
To enable **Block Creation** for an existing project, go to **Project > Project Settings**. Click **Design Flow** and check the **Enable Block Creation** check box. The existing project must use the **Enhanced Constraint Flow**.

1.1 Synthesis Tool Settings

In SynplifyPro, the **I/O Insertion** option is disabled when the block is synthesized.

Important: Libero® automatically disables **I/O Insertion** for you before invoking SynplifyPro.

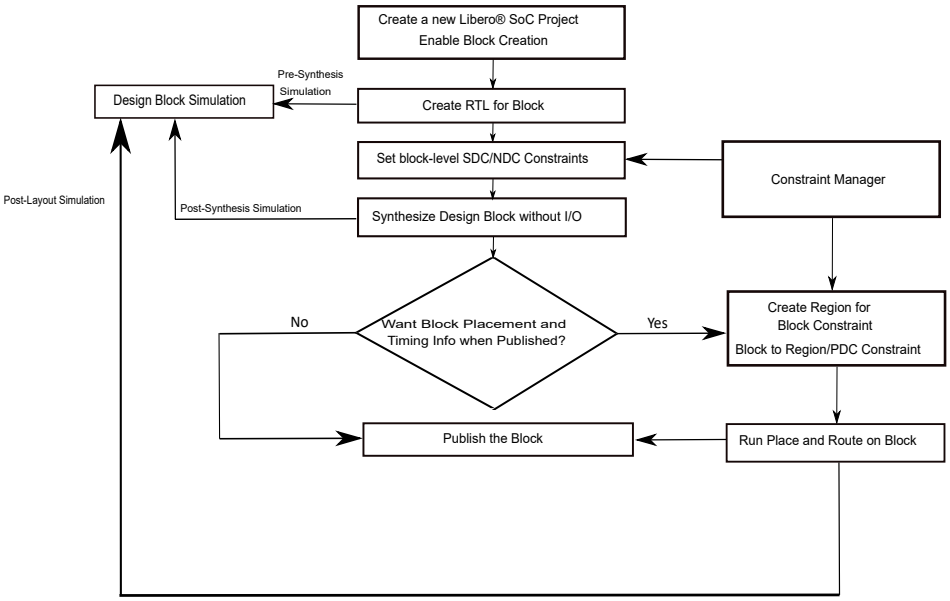
1.2 Synthesis

During synthesis, Libero SoC software adds BLOCK_INTERFACE_I* instances to the block. These instances are virtual buffers added to:

- Improve timing values for the block
- Provide a clear interface to floorplan
- Help with clustering constraints

Important: The BLOCK_INTERFACE_I* instances are removed when the block is published.

Figure 1-1. Creating and Publishing Design Blocks



1.3 Guidelines for Creating Blocks

Few guidelines for creating blocks are to check on the number of globals allowed, DRC rules, floorplanning, and architecture limitations.

1.3.1 Macros/IPs Not Supported in Blocks

The following macros are not allowed when creating a block for instantiation is in a top-level design:

- MSS
- SERDESIF
- FDDR
- FLASH_FREEZE
- SYSRESET
- UJTAG
- µPROM

1.3.2 Synthesis Tool and Global Management

The synthesis tool may promote all clocks to global. The number of globals in your top-level design should be same as number of globals allowed in the device (8 or 16 depending on the device size) you are targeting. You may need to reduce or limit the number of globals in your block by adding row global or plan to share globals in the top-level design with the block. Add a row global, to your HDL (RCLKINT).

1.3.3 Blocks and Design Rule Check (DRC)

Based on the regular Libero design flows, DRC rules are applied to the blocks. For example, some DRC rules assume that some pins must be connected to the power nets. These rules are enforced on the blocks in the block flow just as in the regular design flow.

1.3.4 Blocks and Floorplanning

When creating a block, floorplanning is essential if you plan to publish placement information. Before running layout on the block, you must floorplan the design block. Use Chip Planner or PDC commands for floorplanning.

If you do not create a region and constrain the block to the region (floorplanning) or lock the macros before place and route, a warning message appears when you publish the block. It warns you that not all macros in the block have been constrained to regions or locked and therefore your design netlist is only exported when the block is published.

1.3.4.1 Floorplanning with PDC Commands

Floorplanning reduces the risk of placement conflicts of the blocks at the top-level. If you do not constrain your block placement, its components may be placed anywhere on the die.

You can use the `define_region` PDC command to create a rectangular or rectilinear region, and then use the `assign_region` PDC command to constrain all the macros to that region. For details of command, see Libero SoC v11.8 PDC Commands User Guide. Refer to the Libero Help for the `command syntax`.

It is also important to consider the placement of all block interface instances at the boundaries of block regions. This facilitates the interconnection of the block to the top-level design. If the block is highly optimized (densely packed), there may be no routing channels available to connect to any internal block interface instances. Placing all interfaces at block boundaries helps you eliminate routing congestion and failure.

1.3.4.2 Floorplanning with Chip Planner

Refer to Libero SoC v11.8 Chip Planner User Guide for details on how to use the Chip Planner for floorplanning.

1.3.5 Architecture Limitations - Managing Blocks and Global

Architecturally, the silicon has 8 or 16 globals per device, depending on the device size. If you create a block used in the top level design that uses the maximum number of globals close to the device, then it is a good practice to minimize the number of globals when you create the block.

Examine the global report to see the number of globals that is used for the block. To reduce the number of globals used in the block, you may consider clock sharing and the use of row global for the block.

To add an internal global on a port, use either the Synplify Constraints Editor (SCOPE) or an SDC file.

> To add a CLKINT after a CLK port, the command is: `define_attribute {n:CLK} syn_insert_buffer {CLKINT}`.

2. Instantiating Blocks in the Top-Level Design

You can instantiate multiple instances of the same block or multiple blocks in the top-level design. Microchip recommends that you create a new project for your top-level design. To do so:

1. From the **Project** menu, choose **New Project**.
2. Deselect the **Enable Designer Block Creation** check box.
3. Choose the **Family/Die/Package** for the new project for the top-level as follows:
   - If the block is a netlist only and was not published with place and route information, choose the same **Family** as the block for the new project. Choose any **Die** and **Package**.
   - If the block contains placement information, choose the same **Family** and **Die** as the block for the new project, and choose any **Package**.
   - If the netlist contains I/O and placement information, choose the same **Family**, **Die**, and **Package** as the block for the new project.
4. Choose the **Enhanced Constraint Flow** for the top-level design project.

Important: A top-level project created for the Enhanced Constraint Flow can only import and instantiate blocks created and published from an Enhanced Constraint Flow project. It cannot import or instantiate blocks published from the Classic Constraint Flow project. Likewise, a top-level project created for the Classic Constraint Flow cannot import or instantiate blocks created and published from an Enhanced Constraint Flow project.

2.1 Import the Block

Following are the steps to import a block:

1. From the **File** menu choose **Import > Blocks**.
2. Browse to the directory that contains your `<design_block_name>.cxz` file and select it.
3. Click **Open**.

`<design_block_name>` is imported into the `top_level` project. Version control is not supported for imported blocks. If you import the same block twice, the existing block is overwritten by the new one.

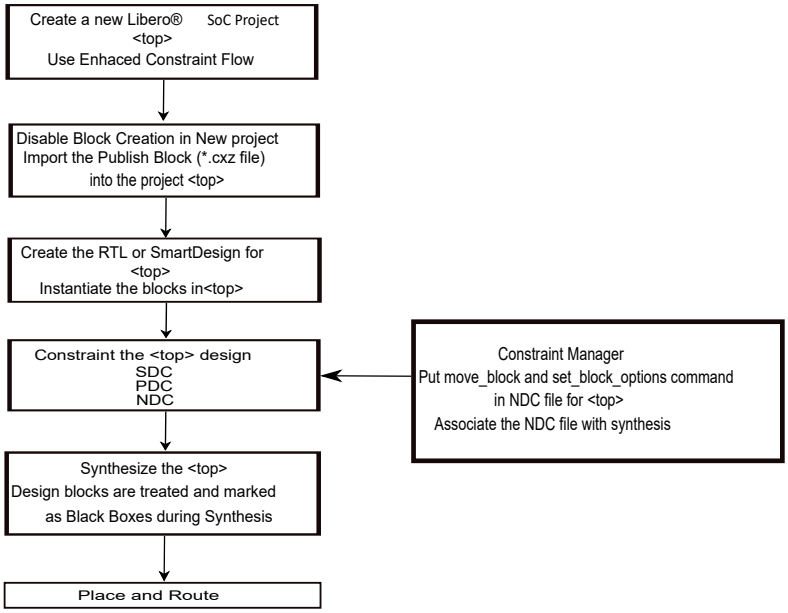
The files will be imported under `<design>\component\work\<design_block_name>`.

Review the files in the preceding directory to view block reports.

2.2 Create a Top-Level Design that Uses Blocks

Use SmartDesign or HDL to create your top-level design. If you use HDL, you can create HDL for the top-level or import a top-level HDL file.

Figure 2-1. Instantiating Blocks in your Top-Level Design



2.3 Constraints Management

When a block with PDC constraints are imported into the top-level design, the block's PDC constraints are captured and stored in two files:

- `<top_level_module>.block.io.pdc` for the IO PDC constraints.
- `<top_level_module>.block.fp.pdc` for the floorplanning PDC constraints.

The `<top_level_module>.block.io.pdc` is displayed in the **I/O Attributes** tab of the **Constraint Manager** on top of any other I/O PDC files.

The `<top_level_module>.block.fp.pdc` is displayed in the **FloorPlanning** tab of the **Constraint Manager** on top of any other floorplanning PDC files.

Important: Do not modify these block PDC files at the top-level. If these PDC files need to be modified, go back to the project where the blocks are created and published. Make the floorplanning modifications and publish the block. Re-import the block into the top-level. You may need to remove any duplicate blocks, if any, at the top-level after the re-import.

3. Hierarchical Structure Resolution in Top-Level Projects

If you import multiple conflicting definitions for your `*.v` files, Libero resolves the conflicts.

3.1 Duplicate Block Definition

If you import two versions of your block file, you must choose which one you want to use. For example:

- Import `top.v` and `block1.v` files as HDL (**File > Import HDL Source Files**) into the top-level project.
- Import `<block1>` (**File > Import > Blocks**).

Libero recognizes a duplicate definition of `<block1>`; one from the HDL and another in the imported block file. The **Design Hierarchy tab** shows a `<block1>.cxf` and `<block1>.v` file under duplicate modules; Libero uses the HDL `<block1>` by default.

To override the default behavior and select the block definition, right click the `<block1>.cxf` file and choose **Use This File**. When you update the behavior, the block icon appears in the **Design Hierarchy**.

3.2 Conflicting Definitions in top.v and Your Imported Block File

You can introduce a conflict if you import a `top.v` file and a block file. Libero does not support HDL definition of low level blocks inside top-level HDL files and subsequent importing of block files. For example, the following scenarios cause an error:

- Import a `top.v` file (**File > Import HDL Source Files**) that contains a definition for `<top>` and a module definition for `<block1>`.
- Import the block `<block1>` (**File > Import > Blocks**).

Libero passes two duplicate files to your synthesis tool because the definition for `<block1>` is duplicated. To continue, you must remove the definition of `<block1>` from `top.v` and then re-import it.

3.3 Resolving top.v and Block Instantiations

Libero integrates your `top.v` file and block file if there is no definition for the block file in `top.v`. For example:

- Import your `top.v` (**File > Import HDL Sources Files**) that contains instantiations but no definition of `<block1>`.
- Import `<block1>` (**File > Import> Blocks**).

Libero resolves the hierarchy for you and puts `<block1>` under `top.v`.

4. EDIF Netlist in the Top-Level Design

If the top-level design is in EDIF, you must convert the EDIF to HDL and then import the HDL into Libero. To convert the top-level EDIF to HDL:

Write a Tcl script as follows:

```
set_device -fam SmartFusion2
read_edif -file {E:\top.edn}
write_verilog -file {E:\top.v} –skip_empty_modules 1
write_vhdl -file {E:\top.vhd}
## -skip_empty_modules 1 is to instruct the tool not to insert module
## definition for the empty modules in the HDL created.
```

From the Windows® Command Prompt or the Linux shell, run `rwnetlist` as follows (this executable is located in the same location as Libero):

`rwnetlist --script "E:/run_export_netlist.tcl”.`

5. Synthesis

Libero passes the block timing to your synthesis tool when the top-level is synthesized. This timing shell enables the synthesis tool to produce more accurate timing numbers for top-level synthesis.

The timing shell also instructs the synthesis tool to treat the design block as a black box; this is done automatically—no action is required.

Use the synthesis tool options (**Design Flow > Synthesize > Configure Options**) for Resolving Place and Route Conflicts of blocks.

6. Resolving Place and Route Conflicts

To resolve place and route conflicts at the top-level:

- Examine the **<design_block_name>_compile_netlist_resources.xml** report. Identify the cause of the problem and manually place and constrain the placement with the floorplanning. For floorplanning details, see Libero SoC v11.8 Chip Planner User Guide.
- If you instantiate a block (published with placement) multiple times, then placement between multiple block instances will overlap. To remove overlapping, move the block placement of one or more instances to another area using the PDC command, `move_block`. Put the `move_block` command inside the NDC file and associate the NDC file with synthesis (**Constraint Manager> Netlist Attributes**).
- The software enforces global sharing. If there is a global driving a CLKINT in the block, it will be deleted. Reduce the number of globals at the top-level by sharing global clock resources. Globals in the blocks may also be re-routed (not preserved).

6.1 Synthesis Options to Resolve Place and Route Conflicts

If there are multiple blocks instantiated in your top-level design, the software uses the synthesis options to resolve the conflicts. These options appear only if there are blocks in your design. Use the synthesis options (**Design Flow > Synthesize > Configure Options**) to resolve placement and/or routing conflicts.

Placement Conflicts

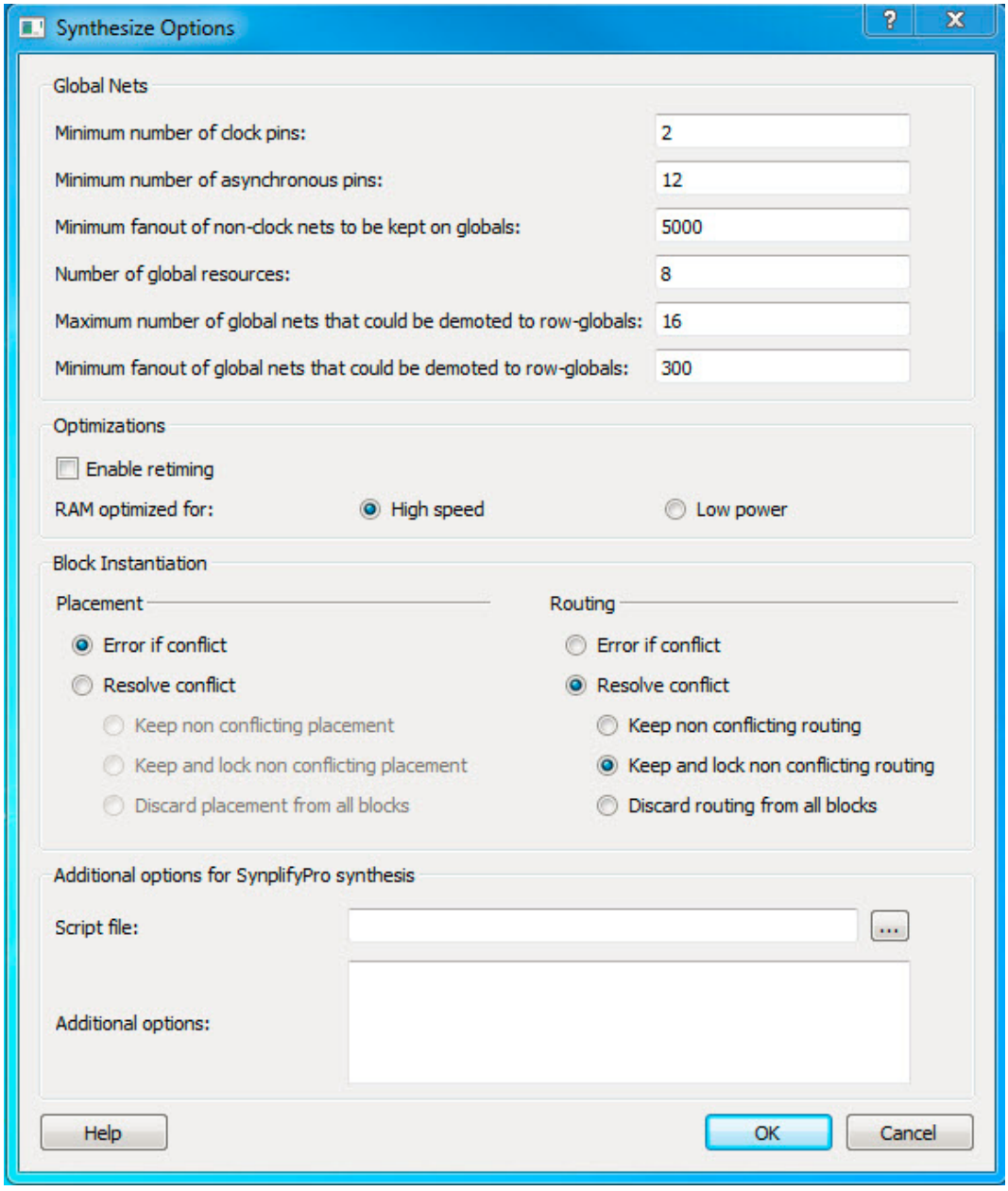
- **Error if conflict**: The layout tool errors out if any instance from a designer block is unplaced. This is the default option.
- **Resolve conflict**: Further categorized into three parts:
  - **Keep non-conflicting placement**: If some instances get unplaced for any reason, the remaining non-conflicting elements are preserved but not locked (you can move them).
  - **Keep and lock non-conflicting placement**: If some instances get unplaced for any reason, the remaining non-conflicting elements are preserved and locked.
  - **Discard placement from all blocks**: Placement information will be discarded from all blocks even if there is no conflict.

Routing Conflicts

There are three methods to resolve a conflict:

- **Error if conflict**: The layout tool errors out if any preserved net routing in a designer block is deleted.
- **Resolve conflict**: Further categorized into three parts:
  - **Keep non-conflicting routing**: If a nets' routing is removed for any reason, the routing for the non-conflicting nets is preserved but not locked (so that they can be rerouted). This is the default option.
  - **Keep and lock non-conflicting routing**: If the routing is removed for any reason, the remaining non-conflicting nets are preserved and locked; they cannot be rerouted. This is the default option.
  - **Discard routing from all blocks**: Routing information will be discarded from all blocks even if there is no conflict.

Figure 6-1. Synthesis Options Dialog Box



7. Block PDC Commands

`move_block` and `set_block_options` are two PDC commands available specifically for working with design blocks at the top-level.

Use the `move_block` and `set_block_options` commands to make changes in your top-level design.

In the top-level design, put the `move_block` and `set_block_options` commands in an NDC file **(Design Flow Window > Manage Constraints > Open Manage Constraints View > Netlist Attributes > New > Create New Compile Netlist Constraints NDC)** and associate the NDC file with Synthesis.

7.1 move_block

PDC command; moves a block from its original, locked placement by preserving the relative placement between the instances. You can move the block to the left, right, up, or down.

Important: Routing is preserved when you move the blocks for IGLOO, SmartFusion, Fusion and ProASCI3 families.

Argument

-inst_name instance_name Specifies the name of the instance to move. If you do not know the name of the instance, run a Compile report or look at the names shown in the **Block** tab of the Libero SoC v11.8 Chip Planner User Guide.

-up y Moves the block up the specified number of rows. The value must be a positive integer.

-down y Moves the block down the specified number of rows. The value must be a positive integer.

-left x Moves the block left the specified number of columns. The value must be a positive integer.

-right x Moves the block right the specified number of columns. The value must be a positive integer.

-non_logic value Specifies what to do with the non-logic part of the block if one exists.

The following table lists the acceptable values for this argument:

Table 7-1. Acceptable Values

Value	Description
move	Moves the entire block.
keep	Moves only the logic portion of the block (COMB/SEQ) and keeps the rest locked in the same previous location, if there is no conflict with other blocks.
unplace	Moves only the logic portion of the block (COMB/SEQ) and unplace the rest of it, such as I/Os and RAM.

Supported Families

The `move_block` PDC command supports the following families:

- SmartFusion2
- IGLOO2
- RTG4
- SmartFusion
- IGLOO

- ProASIC3
- Fusion

Description

This command moves a block from its original, locked position to a new position.

You can move the entire block or just the logic part of it. You must use the `-non_logic` argument to specify what to do with the non-logic part of the block. You can find placement information about the block in the block report.

The `-up`, `-down`, `-left`, and `-right` arguments enable you to specify how to move the block from its original placement. You cannot rotate the block, but the relative placement of macros within the block will be preserved and the placement will be locked. However, routing will be lost. You can either use the **ChipPlanner** tool or run a block report to determine the location of the block.

The `-non_logic` argument enables you to move a block that includes non-logic instances, such as RAM or I/Os that are difficult to move. Once you have moved a part of a block, you can unplace the remaining parts of the block and then place them manually as necessary.

Important: Microchip recommends to move the block left or right by increments of 12, and move the blocks up and down by increments of three. If not, placement may fail because it violates clustering constraints.

Exceptions

Following are the exceptions:

- You must import this PDC command as a source file, not as an auxiliary file.
- You must use this PDC command if you want to preserve the relative placement and routing (if possible) of a block you are instantiating many times in your design as only one instance will be preserved by default. To preserve other instances, move them using PDC command.

  The following example moves the entire block (instance name instA) 12 columns to the right and 3 rows up: `move_block -inst_name instA -right 12 -up 3 -non_logic move.`

  The following example moves only the logic portion of the block and unplaces the rest by 24 columns to the right and 6 rows up: `move_block -inst_name instA -right 24 -up 6 -non_logic unplace.`

7.2 set_block_options

PDC command; overrides the compile option for placement or routing conflicts for an instance of a block.

```
set_block_options -inst_name instance_name -placement_conflicts value -
routing_conflicts value.
```

Argument

- `-inst_name instance_name`
  Specifies the block instance name. If you do not know the name of the instance, run a block report (**Design > Reports > Blocks > Interface**) or look at the names shown in the **Block** tab of the Chip Planner.
- `-placement_conflicts value`
  Specifies what to do when the software encounters a placement conflict.

The following table lists the acceptable values for this argument:

Table 7-2. Acceptable Values

Value	Description
error	Compiles error out if any instance from a block becomes unplaced or its routing is deleted. This is the default compile option.
resolve	If some instances get unplaced for any reason, the non-conflicting elements remaining are also unplaced. Basically, if there are any conflicts, nothing from the block is kept.
keep	If some instances get unplaced for any reason, the non-conflicting elements remaining are preserved but not locked. Therefore, the placer can move them into another location if necessary.
lock	If some instances get unplaced for any reason, the non-conflicting elements remaining are preserved and locked.
discard	Discards any placement from the block, even if there are no conflicts.

- `-routing_conflicts value`
  Specifies what to do when the software encounters a routing conflict.

Supported Families
The `set_block_options` PDC command supports the following families:

- SmartFusion 2
- IGLOO 2
- RTG4
- SmartFusion
- IGLOO
- ProASIC3
- Fusion

Description
This command enables you to override the compile option for placement or routing conflicts for an instance of a block.

Exceptions
Following are the exceptions:

- You must import this PDC command as a source file, not as an auxiliary file.
- If placement is discarded, the routing is automatically discarded too.

This example makes the designer software display an error if any instance from a block becomes unplaced or the routing is deleted: `set_block_options -inst_name instA -placement_conflicts ERROR -routing_conflicts ERROR`.

8. Publish Block - Configuration Options

To view this dialog box you must first enable block creation in the **Libero SoC Project Settings** or **New > Project Creation Wizard**. After **Block Creation** is enabled, **Publish Block** appears in the **Design Flow** window.

Expand **Publish Design**, right click **Publish Block** , and choose **Configure Options**.

8.1 Publish Block Configuration

Publish Placement: Check this box to publish the placement information for the block.

Important: Assign all macros to regions or lock them to publish placement.

If checked, the published block can be instantiated and used in a top-level design only with the same family and device. If the block contains I/Os, the published block can only be instantiated and used in a top-level design with the same family, device and package.

If unchecked, only a netlist is published for the block. The published block can be instantiated and used in a top-level design for any device and package in the same device family as the block.

Publish Routing : Check this box to retain the routing information with the block when published.

Publish Region : Check this box to retain the region constraint information with the block when published.

8.1.1 Language

Select your **Block Hardware Description Language**. The default is the **Preferred HDL** type set in your **Project Settings**.

8.2 Publish Options

Use the 8. Publish Block - Configuration Options to configure the block for publication.

8.3 Publishing Blocks After Synthesis or Layout

You can publish your block after synthesis or layout.

8.3.1 Publish After Synthesis

If you publish a block after synthesis but before layout, a netlist is exported in the block when published. No place and route information or region constraint information is included in the block when published. A warning message appears when you publish a block before place-and-route.

8.3.2 Publish After Layout

If you publish a block after layout, the placement, routing and/or region constraint information will be published along with the netlist. You can always open the configurator and change the options to publish what you want. All macros must be locked or assigned to regions in order to publish the placement information.

8.3.3 Published Content

Libero exports the `<design>.cxz` file to `<project folder>/designer/<design_block_name>/export` folder when a block is published. The `<design>.cxz` file is a zip file that contains the following files:

- `<design_block_name>_syn.v | <design_block_name>_syn.vhd`: A timing shell file passed to synthesis tool when the top-level design is synthesized. The block is marked and treated as a black box when the top-level design is synthesized.

- `<design_block_name>_sim.v | <design_block_name>_sim.vhd`: A structural HDL netlist for post synthesis simulation of the block.
- `header_report.log`: A log file that contains header information on what and how a block is published, including the options you selected to configure the publication.
- `<design_block_name>_compile_netlist_resources.xml`: An XML file that contains Compile Report detailing resource usage, device info, and a list of high-fanout nets.
- `<design_block_name>_gp_report.xml`: Global Placement and Routing Report.
- `<design_block_name>_compile_netlist_combinational_loops.xml`: Combinational Loops Report.
- `<design>.cdb`: Internal proprietary file containing the optimized netlist, placement, routing or timing constraint information.
- `<design_block_name>.sdc`: Contains the SDC constraints for the block to be used for timing verifications.

The `<design_block_name>.cxz` file is your published block. You can move it to another folder, transfer it to other team members, and so on. When you want to instantiate the block, you need to**Import > design_block_name> .cxz file** in your top level design

9. Revision History

The revision history describes the changes that were implemented in the document. The changes are listed by revision, starting with the most current publication.

Revision	Date	Description
A	08/2022	The following is the list of changes in revision A of the document: • The document was migrated to the Microchip template. • The document number was updated to DS50003360 from 502007420. • Updated: 1.3.1. Macros/IPs Not Supported in Blocks.

Microchip FPGA Support

Microchip FPGA products group backs its products with various support services, including Customer Service, Customer Technical Support Center, a website, and worldwide sales offices. Customers are suggested to visit Microchip online resources prior to contacting support as it is very likely that their queries have been already answered.

Contact Technical Support Center through the website at www.microchip.com/support. Mention the FPGA Device Part number, select appropriate case category, and upload design files while creating a technical support case.

Contact Customer Service for non-technical product support, such as product pricing, product upgrades, update information, order status, and authorization.

- From North America, call **800.262.1060**
- From the rest of the world, call **650.318.4460**
- Fax, from anywhere in the world, **650.318.8044**

Microchip Information

The Microchip Website

Microchip provides online support via our website at www.microchip.com/. This website is used to make files and information easily available to customers. Some of the content available includes:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQs), technical support requests, online discussion groups, Microchip design partner program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Product Change Notification Service

Microchip's product change notification service helps keep customers current on Microchip products. Subscribers will receive email notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, go to www.microchip.com/pcn and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Embedded Solutions Engineer (ESE)
- Technical Support

Customers should contact their distributor, representative or ESE for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in this document.

Technical support is available through the website at: www.microchip.com/support

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip products:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner, within operating specifications, and under normal conditions.
- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip product is strictly prohibited and may violate the Digital Millennium Copyright Act.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is “unbreakable”. Code protection is constantly evolving. Microchip is committed to continuously improving the code protection features of our products.

Legal Notice

This publication and the information herein may be used only with Microchip products, including to design, test, and integrate Microchip products with your application. Use of this information in any other manner violates these terms. Information regarding device applications is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. Contact your local Microchip sales office for additional support or, obtain additional support at www.microchip.com/en-us/support/design-help/client-support-services.

THIS INFORMATION IS PROVIDED BY MICROCHIP "AS IS". MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE, OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL LOSS, DAMAGE, COST, OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP'S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION.

Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks

The Microchip name and logo, the Microchip logo, Adaptec, AVR, AVR logo, AVR Freaks, BesTime, BitCloud, CryptoMemory, CryptoRF, dsPIC, flexPWR, HELDO, IGLOO, JukeBlox, KeeLoq, Kleer, LANCheck, LinkMD, maXStylus, maXTouch, MediaLB, megaAVR, Microsemi, Microsemi logo, MOST, MOST logo, MPLAB, OptoLyzer, PIC, picoPower, PICSTART, PIC32 logo, PolarFire, Prochip Designer, QTouch, SAM-BA, SenGenuity, SpyNIC, SST, SST Logo, SuperFlash, Symmetricom, SyncServer, Tachyon, TimeSource, tinyAVR, UNI/O, Vectron, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AgileSwitch, APT, ClockWorks, The Embedded Control Solutions Company, EtherSynch, Flashtec, Hyper Speed Control, HyperLight Load, Libero, motorBench, mTouch, Powermite 3, Precision Edge, ProASIC, ProASIC Plus, ProASIC Plus logo, Quiet- Wire, SmartFusion, SyncWorld, Temux, TimeCesium, TimeHub, TimePictra, TimeProvider, TrueTime, and ZL are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, Augmented Switching, BlueSky, BodyCom, Clockstudio, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, Espresso T1S, EtherGREEN, GridTime, IdealBridge, In-Circuit Serial Programming, ICSP, INICnet, Intelligent Paralleling, IntelliMOS, Inter-Chip Connectivity, JitterBlocker, Knob-on-Display, KoD, maxCrypto, maxView, memBrain, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PowerSmart, PureSilicon, QMatrix, REAL ICE, Ripple Blocker, RTAX, RTG4, SAM-

ISBN: 978-1-6683-1100-4

Quality Management System

For information regarding Microchip's Quality Management Systems, please visit www.microchip.com/quality.

Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
Corporate Office 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 Technical Support: www.microchip.com/support Web Address: www.microchip.com **Atlanta** Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455 **Austin, TX** Tel: 512-257-3370 **Boston** Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088 **Chicago** Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075 **Dallas** Addison, TX Tel: 972-818-7423 Fax: 972-818-2924 **Detroit** Novi, MI Tel: 248-848-4000 **Houston, TX** Tel: 281-894-5983 **Indianapolis** Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380 **Los Angeles** Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800 **Raleigh, NC** Tel: 919-844-7510 **New York, NY** Tel: 631-435-6000 **San Jose, CA** Tel: 408-735-9110 Tel: 408-436-4270 **Canada - Toronto** Tel: 905-695-1980 Fax: 905-695-2078	**Australia - Sydney** Tel: 61-2-9868-6733 **China - Beijing** Tel: 86-10-8569-7000 **China - Chengdu** Tel: 86-28-8665-5511 **China - Chongqing** Tel: 86-23-8980-9588 **China - Dongguan** Tel: 86-769-8702-9880 **China - Guangzhou** Tel: 86-20-8755-8029 **China - Hangzhou** Tel: 86-571-8792-8115 **China - Hong Kong SAR** Tel: 852-2943-5100 **China - Nanjing** Tel: 86-25-8473-2460 **China - Qingdao** Tel: 86-532-8502-7355 **China - Shanghai** Tel: 86-21-3326-8000 **China - Shenyang** Tel: 86-24-2334-2829 **China - Shenzhen** Tel: 86-755-8864-2200 **China - Suzhou** Tel: 86-186-6233-1526 **China - Wuhan** Tel: 86-27-5980-5300 **China - Xian** Tel: 86-29-8833-7252 **China - Xiamen** Tel: 86-592-2388138 **China - Zhuhai** Tel: 86-756-3210040	**India - Bangalore** Tel: 91-80-3090-4444 **India - New Delhi** Tel: 91-11-4160-8631 **India - Pune** Tel: 91-20-4121-0141 **Japan - Osaka** Tel: 81-6-6152-7160 **Japan - Tokyo** Tel: 81-3-6880- 3770 **Korea - Daegu** Tel: 82-53-744-4301 **Korea - Seoul** Tel: 82-2-554-7200 **Malaysia - Kuala Lumpur** Tel: 60-3-7651-7906 **Malaysia - Penang** Tel: 60-4-227-8870 **Philippines - Manila** Tel: 63-2-634-9065 **Singapore** Tel: 65-6334-8870 **Taiwan - Hsin Chu** Tel: 886-3-577-8366 **Taiwan - Kaohsiung** Tel: 886-7-213-7830 **Taiwan - Taipei** Tel: 886-2-2508-8600 **Thailand - Bangkok** Tel: 66-2-694-1351 **Vietnam - Ho Chi Minh** Tel: 84-28-5448-2100	**Austria - Wels** Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 **Denmark - Copenhagen** Tel: 45-4485-5910 Fax: 45-4485-2829 **Finland - Espoo** Tel: 358-9-4520-820 **France - Paris** Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79 **Germany - Garching** Tel: 49-8931-9700 **Germany - Haan** Tel: 49-2129-3766400 **Germany - Heilbronn** Tel: 49-7131-72400 **Germany - Karlsruhe** Tel: 49-721-625370 **Germany - Munich** Tel: 49-89-627-144-0 Fax: 49-89-627-144-44 **Germany - Rosenheim** Tel: 49-8031-354-560 **Israel - Ra'anana** Tel: 972-9-744-7705 **Italy - Milan** Tel: 39-0331-742611 Fax: 39-0331-466781 **Italy - Padova** Tel: 39-049-7625286 **Netherlands - Drunen** Tel: 31-416-690399 Fax: 31-416-690340 **Norway - Trondheim** Tel: 47-72884388 **Poland - Warsaw** Tel: 48-22-3325737 **Romania - Bucharest** Tel: 40-21-407-87-50 **Spain - Madrid** Tel: 34-91-708-08-90 Fax: 34-91-708-08-91 **Sweden - Gothenberg** Tel: 46-31-704-60-40 **Sweden - Stockholm** Tel: 46-8-5090-4654 **UK - Wokingham** Tel: 44-118-921-5800 Fax: 44-118-921-5820