Lattice Radiant Block-Based Design Tutorial

December 11, 2025

Copyright

Trademarks

Disclaimers

Type Conventions Used in This Document

Convention	Meaning or Use
Bold	Items in the user interface that you select or click. Text that you type into the user interface.
<*Italic*>	Variables in commands, code syntax, and path names.
Ctrl+L	Press the two keys at the same time.
`Courier`	Code examples. Messages, reports, and prompts from the software.
`...`	Omitted material in a line of code.
`.` `.` `.`	Omitted lines in code and report examples.
`[ ]`	Optional items in syntax descriptions. In bus specifications, the brackets are required.
`( )`	Grouped items in syntax descriptions.
`{ }`	Repeatable items in syntax descriptions.
`|`	A choice between items in syntax descriptions.

Contents

Lattice Radiant Block-Based Design Tutorial

The Block-Based Design (Macro) tool is used to create design blocks from implemented module of a device family. This tutorial leads you through all the basic steps of creating, reusing, and exporting a macro block.

A macro block is a portion of an FPGA design that can be preserved for reuse afterwards. Some of the benefits of implementing macros in a design include the ability to reuse blocks in other designs and the ability to design more effectively as a team.

About the Tutorial

When you have completed this tutorial, you should be able to do the following:

- **Macro Creation**
  - Create a Macro for your design.
  - Use Physical Designer to set placement and routing physical constraints (optional).
  - Export Macro using three preservation levels:
    - Logical (Logic Macro)
    - Logical & Physical with Place Info (Firm Macro)
    - Logic & Physical with Placement & Routing Info (Hard Macro)
- **Macro Reuse**
  - Import macro (.ipm) package as a design source into existing or new project for reuse.
  - Loaded macro content into the design:
    - Logic Macro will be included in the post-synthesis netlist.

- Firm/Physical Macro will be included in the post-MAP netlist.
- Run Place & Route to recognize macro physical constraint if it is in the macro package.

Time to Complete

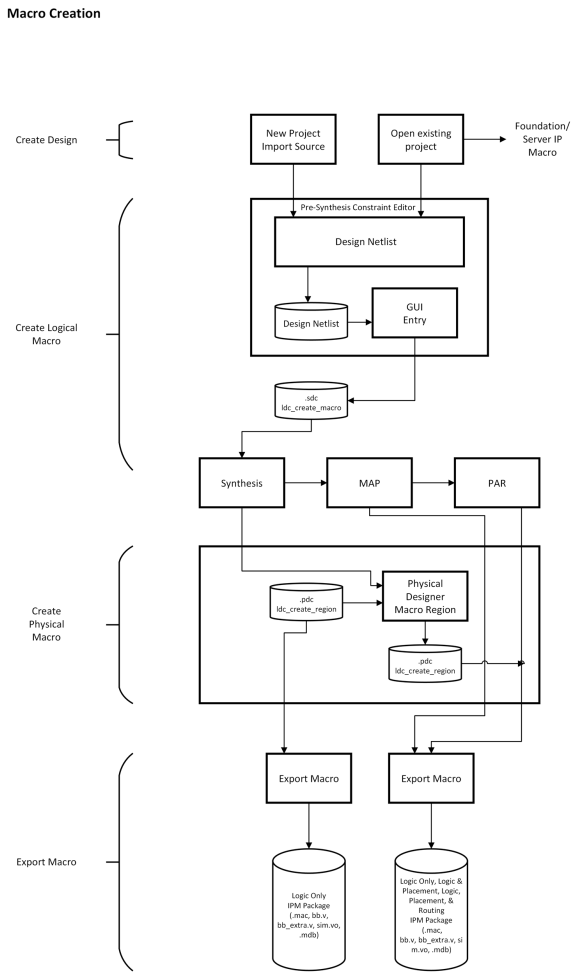
About 45 minutes.

System Requirements

- The Lattice Radiant software is required to complete the tutorial.

About the Tutorial Data Flow

The following figures illustrate the tutorial data flow of macro creation and reuse.

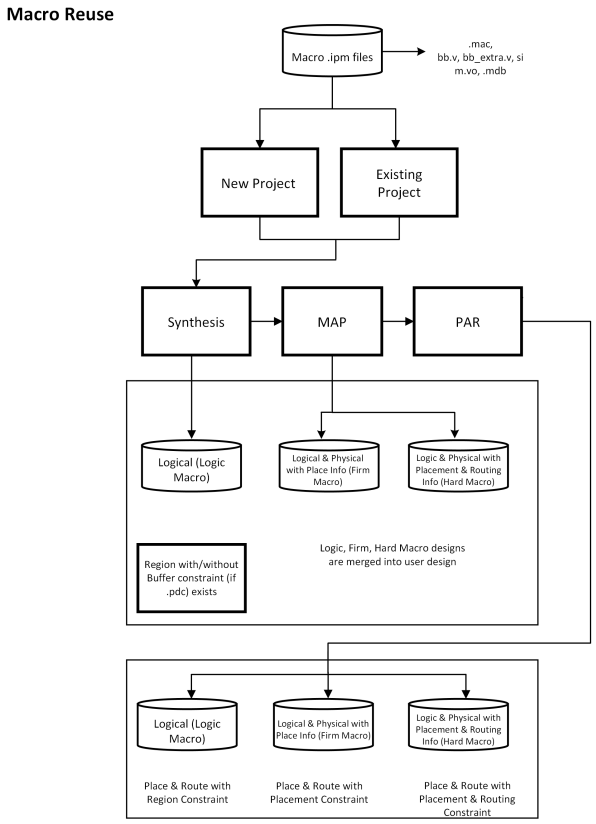


This diagram shows the steps for the macro creation process. First, a design is made using a new or existing project. The Pre-Synthesis Constraint Editor is then used to define a logical macro that is created during synthesis. Once

changes are made in the Pre-Synthesis Constraint Editor, the defined macro is saved in a.sdc file containing the ldc_create_macro constraint.

After synthesis, the next step is creating a macro region in Physical Designer (optional). When you save your changes in Physical Designer, a .pdc file will be generated containing the ldc_create_region constraint.

The next stage is exporting the macro using Logic, Firm, and Hard preservation levels after running the implementation process (Synthesis, MAP, and PAR). The macro package generated by this export contains a number of files, including .mac, bb.v, bb_extra.v, sim.vo, and .mdb files. These files provide the information you need to use the macro and incorporate it into other designs or projects.



This diagram shows how macro models are incorporated and reused in the flow. First, the macro .ipm files are imported and the content is loaded into the design. By instantiating the macro in a new or existing project, the macro can be reused. The .ipm file provides the macro's content, and the IPM package contains a black_box module that you can access. The name provided in the ldc_create_macro command during the creation phase is the same as the name of the macro module. The ldc_create_macro constraint is not required during reuse.

The Post-synthesis Logic Macro is merged into the design, ensuring that the macro constraint is honored. A region with or without a buffer constraint can be specified if the .pdc file exists.

In Map Design, Firm Macro is merged into the design, incorporating the post-MAP macro design with a placement constraint. Hard Macro is also integrated into the design with both placement & route constraints.

In Place & Route Design stage, the Logic Macro undergoes place & route with a region constraint, while the Firm Macro goes through Place & Route with a placement constraint. Finally, the Hard Macro goes through Place & Route with both placement and routing constraints.

Task 1: Create a New Macro Project

We will start by launching the Radiant software.

Opening the Project

To create a macro instance using a new project:

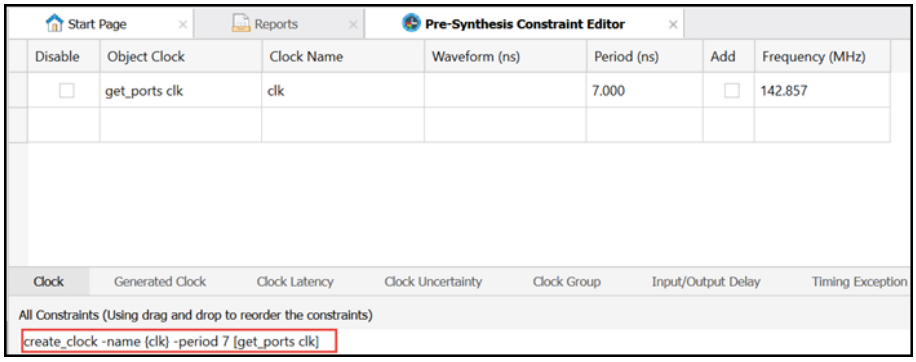
1. In the Radiant software, choose **File** > **New Project**.
   - If you have an existing project, steps 1 through 5 are optional. You can proceed to step 6.
2. In the Add Source dialog box, click **Add Source...**
3. Browse to: *<Radiant_install_path>/docs/tutorial/**block_based_tutorial/macro_create*** and select the **top.v** file.

   Click **Next**.
4. In the **Select Device** dialog box, select a device family and choose the options that you want for that device.
   - The iCE40UP device is not supported.

   Click **Next**.
5. In the **Select Synthesis Tool** dialog box, choose **Synplify Pro** or **Lattice LSE**.
6. Click **Tools** > **Pre-Synthesis Constraint Editor**.

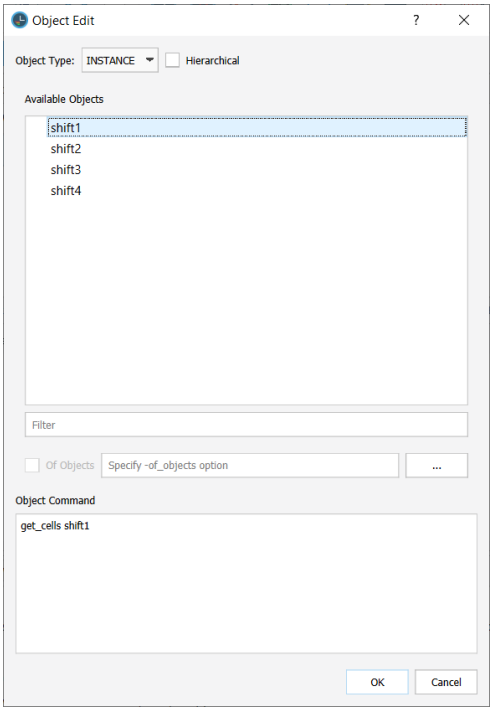
   Synthesize Design runs when you open the Pre-Synthesis Constraint Editor tool.
7. Click the ··· button in the **Object Clock** column to add clock constraints (optional).

   If clock constraints are added, the **create_clock** constraint will be shown in the command line.



8. In **Pre-Synthesis Constraint Editor**, navigate through the list of tabs and select the **Macro** tab at the right-hand corner of the Pre-Synthesis Constraint Editor.

   a. Click the ··· button in the **Instance** column to add shift registers.



   In the example above, **shift1**, **shift2**, **shift3**, and **shift4** are **Instance Names**. The command is **get_cells** as shown in the **Object Command** pane.

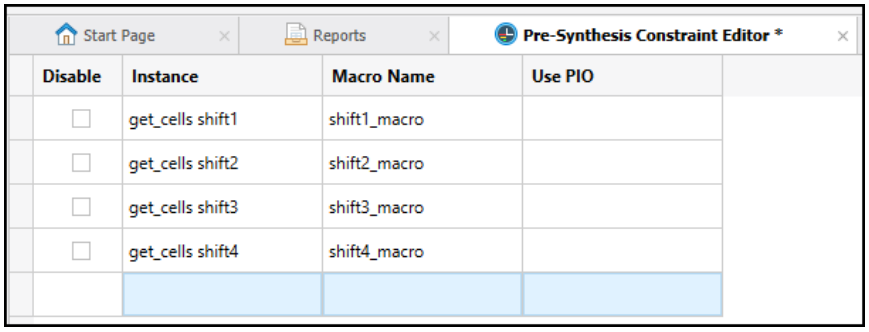
   b. In the **Macro Name** column, the name is automatically set for the macro instance.

      You can also edit the name by double-clicking a cell under the Macro Name column.

   c. In the **Use PIO** column, you can select the ports of the module to include the device pins inside the macro (optional).

If the macro block has padded IO ports, this step is required.

d. In the **Disable** column, you can click on the checkbox if you do not want to use the macro.



e. Click **File > Save** or the **Save** icon.

The defined macro will be saved as a synthesis constraint (.sdc) file.

You can also directly add **ldc_create_macro** constraint to the .sdc file.

Note:

By default, Radiant tools produce hierarchical paths with a "/" (forward slash) as separator. However, Synplify uses a "." (period). In order for the flow to work, any hierarchical path needs to be adjusted for Synplify by either setting **set_hierarchy_separator {/}** to the constraint file or by manually changing any hierarchical path created in Radiant to use "." instead of "/."

9. Run **Synthesize Design**.

Task 2: Set Macro Region

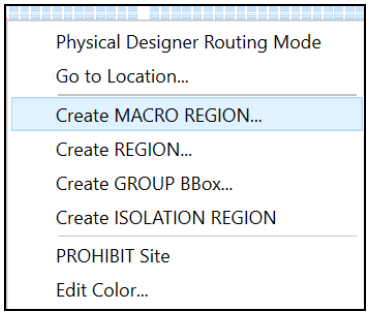
After running Synthesize Design, you can set a macro region in Physical Designer (optional).

To create a Macro Region in Physical Designer:

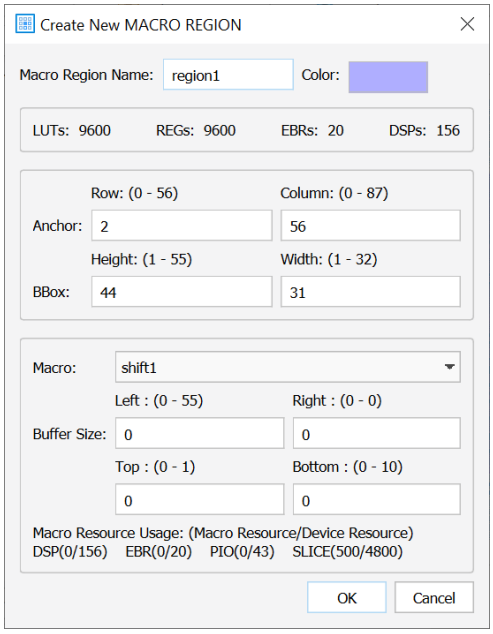
1. Open **Physical Designer**.
2. Right-click on the floorplan layout and select the **Create MACRO REGION…** option.

   The mouse pointer changes from an arrow to a cross symbol.
3. Draw a rectangle around the sites where you want to place the macro.

   Create New MACRO REGION... dialog box opens.

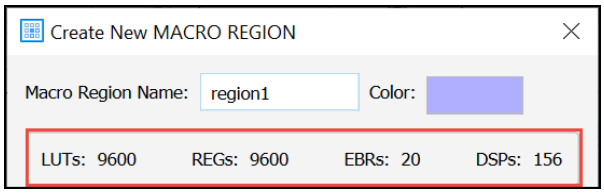


4. In the **Create New MACRO REGION** dialog box:



a. Enter the **Macro Region Name** (mandatory).

   Below the Macro Region Name field, the macro resource is displayed as a guide for the BBox size (LUTs, REGs, EBRs, and DSPs).



   To get an idea of what the minimum macro region size should be, you can check the logic resource usage in the MAP Report (.mrp) file.

   Calculating the region resources to compare macro resource usage is different for each device family.

   For example:

One PFU has 4 SLICEs (=8 LUTs + 8 REGs) for most devices, but 6 SLICEs (=12 LUTs + 12 REGs) per PFU for Avant.

You may also want to check non-SLICE type resource usage (e.g., EBR and DSP). These two blocks only appear at one out of multiple rows, and the resources covered by region also depend on the anchor location.

b. Enter your desired values for the **Anchor** and **BBox** fields.

You must enter a value within range.

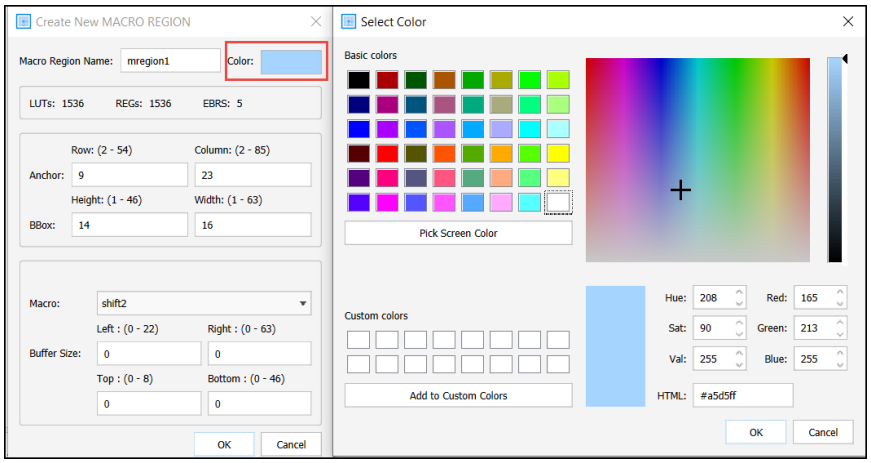
c. In the **Macro** dropdown menu, you can select the predefined macro instance from the pre-synthesis macro creation.

d. Enter your desired value for the **Buffer Size** field (optional).

Buffer Size gives PAR an extra room to do the routing. The routing is confined within the macro region plus its surrounding ring. The top, bottom, right, and left buffer size could be set differently.

The Buffer Size range is device density dependent. You must enter a value within the range.

e. Click the **Color** option to change the color of the macro region (optional).



5. Click **OK**.
6. Click the **Save** icon in **Physical Designer**.

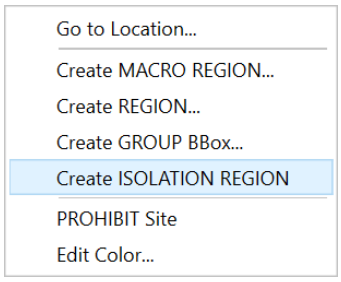
The created macro regions are saved into a post-synthesis constraint (.pdc) file.

The .pdc file contains the following information:

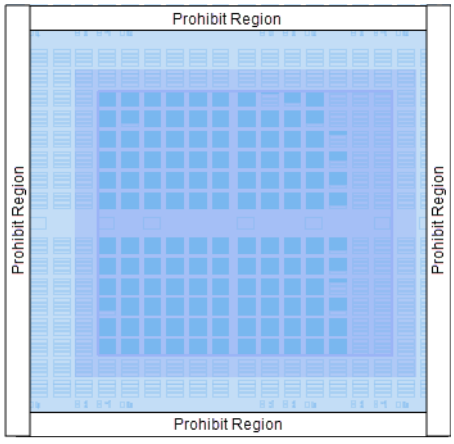
- **ldc_create_region** – Defines the placement region.
- **buffer_left/-buffer_right/-buffer_top/-buffer_bottom** (optional) – Defines routing region.

```
ldc_create_region -name mregion1 -site R4C2D -width 13 -height 13 -buffer_left 1 -buffer_right 1 -buffer_top 1 -buffer_bottom 1
ldc_set_location -region mregion1 [get_cells shift1]

ldc_create_region -name mregion2 -site R22C20D -width 13 -height 13 -buffer_left 1 -buffer_right 1 -buffer_top 1 -buffer_bottom 1
ldc_set_location -region mregion2 [get_cells shift2]

ldc_create_region -name mregion3 -site R39C44D -width 13 -height 13 -buffer_left 1 -buffer_right 1 -buffer_top 1 -buffer_bottom 1
ldc_set_location -region mregion3 [get_cells shift3]
```

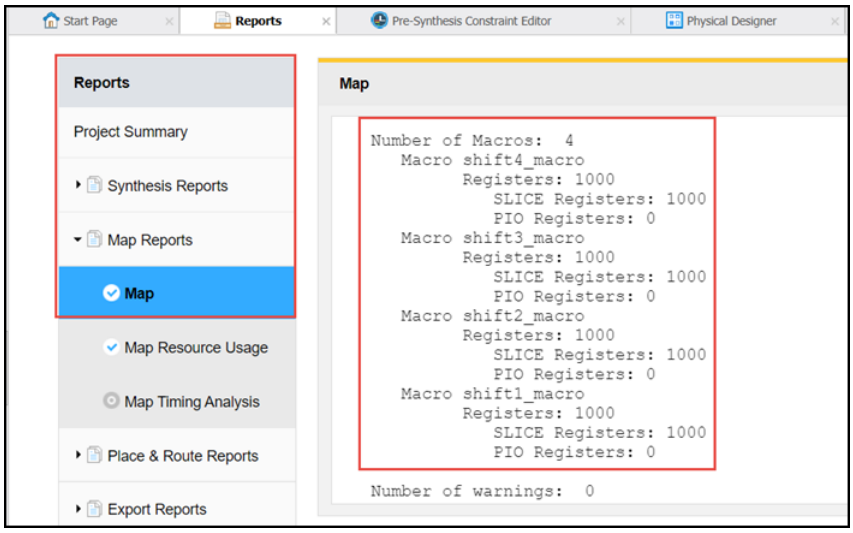
7. You can also create an isolated region by right-clicking on the floorplan layout and selecting **Create ISOLATION REGION.**



   a. Draw a rectangle around the sites where you want to place the isolated region.
      - Physical Designer automatically creates prohibited regions.
      - This feature locks placement and routing, and excludes all other logic from using the region for placement or routing.
   b. If a gap exists between the isolation and secure region, some design instances may be placed within it by PAR. The gap will not be considered as an empty/secure region. You need to manually cover the prohibited region by expanding it. To expand the isolated regions, you can do one of the following:
      - Double-click on the region that you want to expand. The **Edit Region Property** dialog box opens. Change the existing values of the BBox field to resize the region.
      - Select the region that you want to expand and drag it.
      - Open the .pdc file and change the height or width of the isolated region in the ldc_create_region constraint.



8. Run **Map Design**
9. Click the **Reports** tab and expand **Map Reports** in the Project Summary pane.



This section lists the macros you have created and the total number of resources used.

Task 3: Run PAR to Complete Macro Creation

After creating a macro region in the current project, run **Place & Route Design.**

If you want to import an existing .pdc file, do the following:

1. In **File List** view, right-click on **Post-Synthesis Constraints Files** folder.
2. Click **Add** > **Existing File**.

   Browse to where your existing .pdc file is located and click **Add**.
3. Run **Place & Route Design.**

Task 4: Export Macro

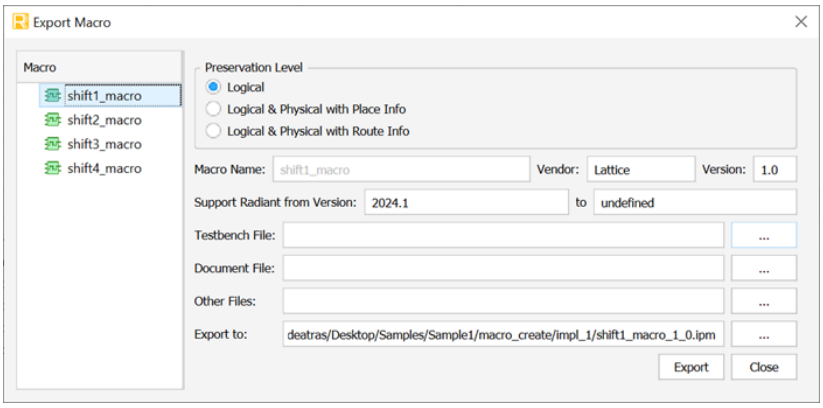
1. Choose **Tools** > **Export Macro**

   The **Export Macro** dialog box opens.

   The macro data can be exported according to a specified preservation level. You can also lower the preservation level of an exported macro after importing it to a design.

- **Logic Macro**
  - After running **Synthesize Design** and **Map Design** (optional), you can only export to **Logical** preservation level.
- **Firm Macro**

- For Firm Macro, you can export to the following preservation levels after running **Place & Route Design.**
  - Logical
  - Logical & Physical with Place Info
- **Hard Macro**
  - For Hard Macro, you can export to the following preservation levels after running **Place & Route Design**.
    - Logical
    - Logical & Physical with Place Info
    - Logic & Physical with Placement & Routing Info



You can also see the following fields in the Export Macro dialog box:

- **Macro Name** – Displays the name of the selected macro from the Macro column in the left-hand pane.
- **Vendor** – Displays the vendor name.
- **Version** – Displays the Macro IP version, the default is always 1.0.
- **Support Radiant from Version to Version** – Displays the supported maximum and minimum Radiant version. The maximum version could be empty, but the minimum is the same as the current Radiant version.
- **Testbench File** – Contains testbenches that allow you to do simulation or evaluate the macro.
- **Document File** – Contains any document such as help, user guide, or introduction file included in the macro.
- **Other Files** – Additional files to be exported with the macro.
- **Export to** – Default path where the macro package will be saved.

2. Choose the preservation level for each instance then click **Export**.

   The exported Macros are saved as .ipm package files.

   The macro .ipm package contains the following:

   - **<macro_name>_.mac** – Contains general information of the macro project.

- **<macro_name>_bb.v** – Synthesis header file of the macro project. It is exported to define module interface for synthesis.
- **<macro_name>_bb_extra.v** – Synthesis header file exported with additional information.
- **<macro_name>_sim.vo** – Exported for macro simulation. The file is generated based on logic macro.
- **<macro_name>.mdb** – Exported Macro in unified constraints database (.udb) format. Modules representing macros will be extracted from design and exported to .mdb files. The macro will be the top module in .mdb file. The name specified in ldc_create_macro constraint will be used as the name of the top module.

Task 5: Reusing an Exported Macro

To add a Macro Block file in a new project:

1. In the Radiant software, choose **File** > **New Project**.
   - If you have an existing project, steps 1 through 4 are optional. You can proceed to step 5.
2. In the Add Source dialog box, click **Add Source...**
   a. Add the following files from the *<Radiant_install_path>/docs/tutorial/**block_based_tutorial/macro_reuse*** directory.
      - top.v
      - macro_reuse.sdc
   b. Click **Next**.
3. In the **Select Device** dialog box, select the following options:
   - **Family**: CrossLink-NX (LIFCL)
   - **Device**: LIFCL-40

   Since LIFCL-40 was used to create the.ipm files in this example, it is recommended to use this device. In this case, other devices will not work.

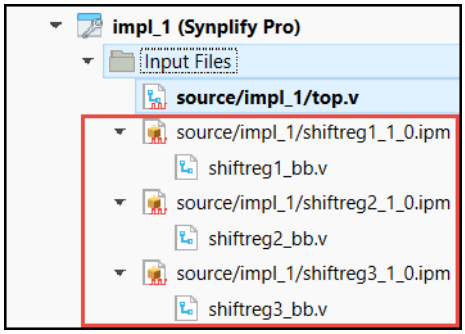
   Click **Next**.
4. Select **Synplify Pro** or **Lattice LSE** as the synthesis tool.
5. In the File List view, right-click **Input Files**.
6. Click **Add** > **Existing File**.

   In the **Add Existing File** dialog box, browse to: *<Radiant_install_path>/docs/tutorial/**block_based_tutorial/macro_reuse/impl_1***
7. Select the following **.ipm files:**
   - shiftreg1_1_0.ipm
   - shiftreg2_1_0.ipm
   - shiftreg3_1_0.ipm

Click **Add**.

8. Enable the **Copy file to directory** check box, then click **Add**.

   You can see the imported .ipm files in the **Input Files** folder.



9. Run **Synthesize Design**.

   The post-synthesis Logic Macro is merged into your design.

   It has <macro_name>_bb.v as the input. The logic synthesis tool treats every module that matches <macro_name>_bb.v as a black box.

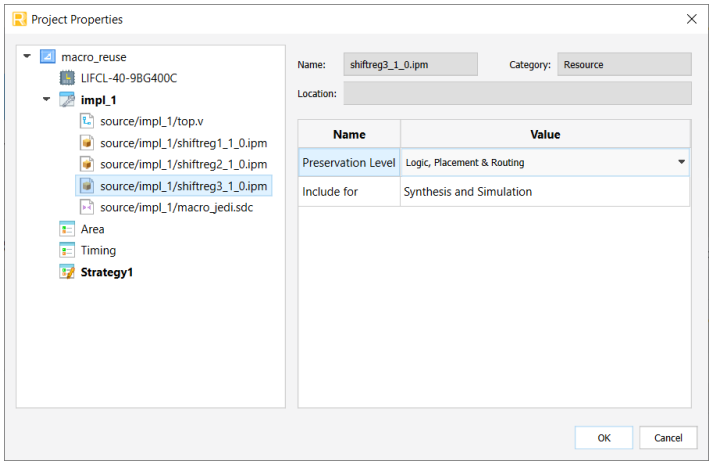
   For example:

```
module shiftreg2 (
    input \clk ,
    input \si ,
    output \so )
/* synthesis TYPE = "MACRO"

            syn_black_box
            syn_resources = "lut=0,ff=1000"
*/;
endmodule
```

10. In the File List View, you can check the **Project Properties** by right-clicking the selected .ipm file to see or change their value.

    You can lower the preservation level value of the imported .ipm files.

    - If the imported macro's preservation level is Firm Macro, you can lower it to Logic Macro.
    - If the imported macro's preservation level is Hard Macro, you can lower it to Logic and Firm Macro.

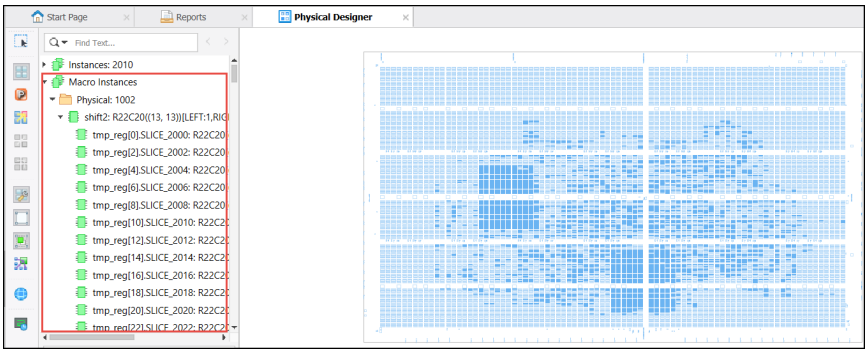


11. Run **Map Design**.

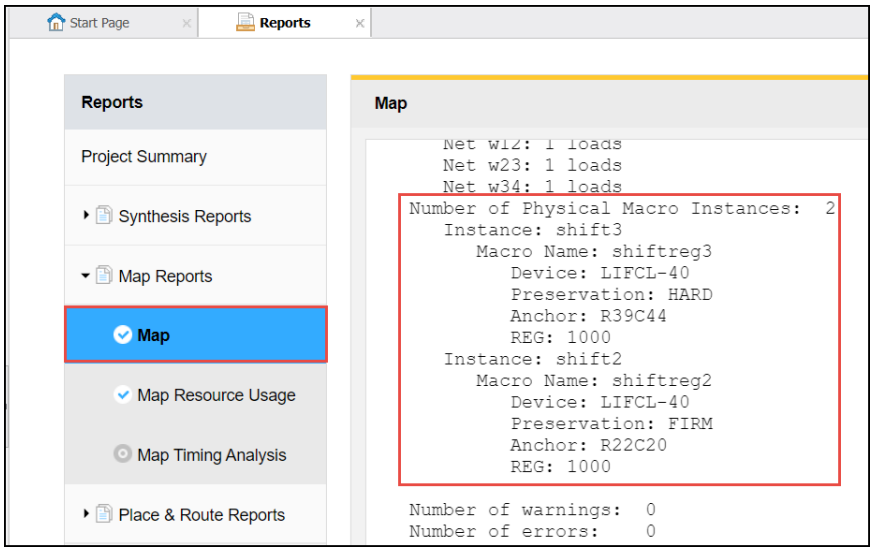
    For Firm Macro, the post-MAP macro design with placement constraint is merged into your design.

    For Hard Macro, the post-MAP macro design with placement and route constraint is merged into your design.

    Open **Physical Designer,** the netlist shows the reused macro instances. They contain information of the original region from the create stage.



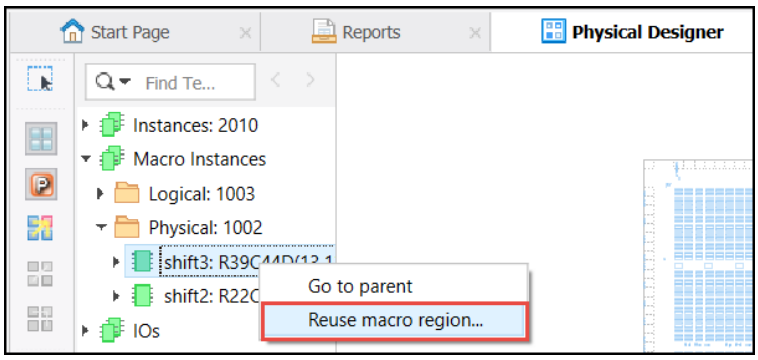
12. Click the **Reports** tab and expand **Map Reports** in the Project Summary pane.



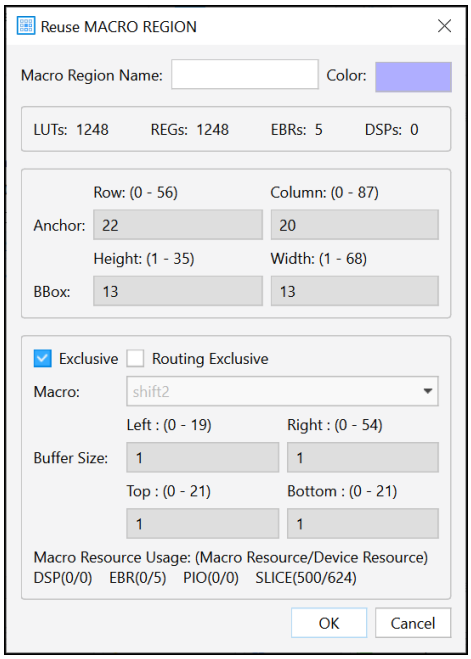
The **Design Summary** section of Map Report displays the following information:

- Total number of reused Firm and Hard macros in the design.
- Device, preservation level, anchor, and main resources.

13. You can create a macro region with the **Exclusive** option in Physical Designer.
    - In **Physical Designer netlist**, right-click on a macro instance under the **Physical** folder.
    - Select the **Reuse macro region...** option.



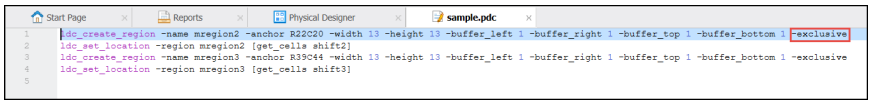
- The **Reuse Macro Region** dialog box opens.



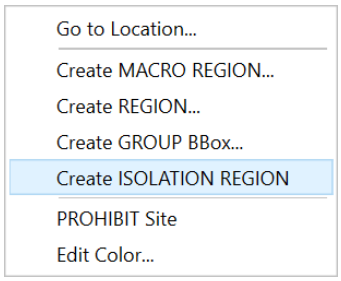
- You can also create a macro region with the Exclusive options in Physical Designer.
  - **Exclusive** – This option excludes other logic from the region for placement and routing when reusing the exported Firm and Hard macro.
  - **Routing Exclusive** – This option excludes other logic routing when reusing the exported Firm and Hard macro.
- Click the **Save** icon in Physical Designer.

  The created macro regions are saved into a post-synthesis constraint (.pdc) file.

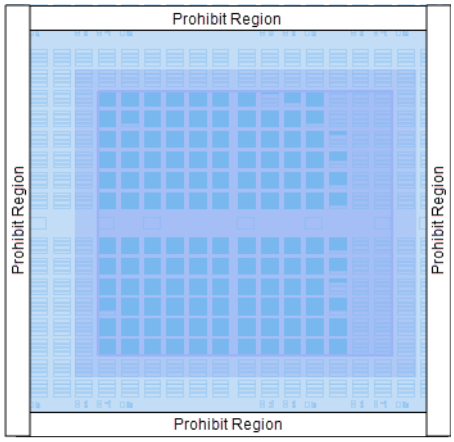
  If the Exclusive checkbox is enabled, the **-exclusive** option is added to the **ldc_create_region** command line in the .pdc file.



14. You can also create an isolated region by right-clicking on the floorplan layout and selecting **Create ISOLATION REGION.**

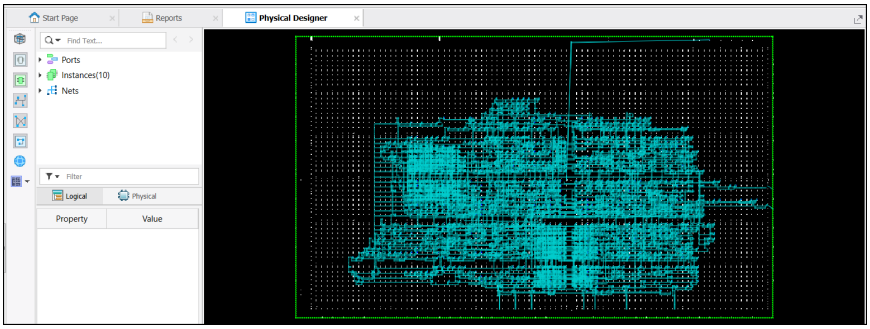


- Draw a rectangle around the sites where you want to place the isolated region.
  - Physical Designer automatically creates prohibited regions.
  - This feature locks placement and routing, and excludes all other logic from using the region for placement or routing.
- If a gap exists between the isolation and secure region, some design instances may be placed within it by PAR. The gap will not be considered as an empty/secure region. You need to manually cover the prohibited region by expanding it. To expand the isolated regions, you can do one of the following:
  - Double-click on the region that you want to expand. The **Edit Region Property** dialog box opens. Change the existing values of the BBox field to resize the region.
  - Select the region that you want to expand and drag it.
  - Open the .pdc file and change the height or width of the isolated region in the ldc_create_region constraint.



15. Run **Place & Route Design**.

    Open **Physical Designer** and select **Routing Mode** to see the macro instances in the physical netlist.



Summary of Accomplishments

You have completed the *Lattice Radiant Block-Based Design Tutorial* . In this tutorial, you have learned how to:

- Create a macro block in Pre-Synthesis Constraint Editor.
- Create a macro region in Physical Designer
- Export Macro using three preservation levels
  - Logical (Logic Macro)
  - Logical & Physical with Place Info (Firm Macro)
  - Logic & Physical with Placement & Routing Info (Hard Macro)
- Import existing .ipm files for macro reuse.
- Change the preservation level value of .ipm files.

Recommended References

You can find additional information on the subjects covered by this tutorial in the Radiant Help:

- Entering the Design > Block-Based Design - Using Macro Blocks > Creating a Macro Block
- Entering the Design > Block-Based Design - Using Macro Blocks > Creating a Macro Region
- Entering the Design > Block-Based Design - Using Macro Blocks > Reusing a Macro Block
- Entering the Design > Block-Based Design - Using Macro Blocks > Exporting Macro
- Entering the Design > Block-Based Design - Using Macro Blocks > Macro Usage Guidelines

Revision History

The following table gives the revision history for this document.

Date	Version	Description
12/11/2025	2025.2	Updated to reflect changes in Radiant 2025.2.
06/26/2025	2025.1	Updated to reflect changes in Radiant 2025.1.
12/20/2024	2024.2	Updated to reflect changes in Radiant 2024.2.
06/28/2024	2024.1	Updated to reflect changes in Radiant 2024.1.
11/20/2023	2023.2	Updated to reflect changes in Radiant 2023.2.
06/22/2023	2023.1	Initial release.