PDC Commands Reference Guide for PolarFire and PolarFire SoC FPGAs

Libero SoC v2025.2

Introduction (Ask a Question)

In the FPGA design world, constraint files are as important as design source files. Physical Design Constraints (PDC) are used to constrain I/O attributes, placement, and routing during the physical layout phase.

You can enter PDC commands manually using the Libero® SoC Text Editor. PDC commands can also be generated by the Libero SoC interactive tools. The I/O Attribute Editor is the interactive tool for making the I/O attributes changes, and the Chip Planner is the interactive tool for making the floor-planning changes. When changes are made in the I/O Attribute Editor or the Chip Planner, the PDC file(s) are updated to reflect these changes. These PDC commands are used as part of a script file to constrain the Place-and-Route step of your design.

Supported Families (Ask a Question)

This user guide describes the I/O and floorplanning PDC commands applicable to PolarFire® and PolarFire SoC FPGA devices.

Table of Contents

1. PDC Syntax Conventions (Ask a Question)

The following table lists the typographical conventions that are used for the PDC command syntax.

Table 1-1. Typographical Conventions Used for the PDC Command Syntax

Syntax Notation	Description
`command -argument`	Commands and arguments appear in Courier New typeface.
[-argument value]	Optional arguments begin and end with a square bracket.

Note: PDC commands and arguments are case sensitive.

1.1. Examples (Ask a Question)

Syntax for the `set_io` command followed by a sample command starting with `set_io EXT_RST_N \`:

Syntax:

```
-port_name <port_name> [-pin_name <package_pin>] [-fixed <true|false>] [-io_std
<io_std_values>] [-OUT_LOAD <value>] [-RES_PULL <value>] [-LOCK_DOWN <value>] [-CLAMP_DIODE
<value>] [-SCHMITT_TRIGGER <value>] [-SLEW <value>] [-VCM_RANGE <value>] [-ODT <value>]
[-ODT_VALUE <value>] [-OUT_DRIVE <value>] [-IMPEDANCE <value>] [-SOURCE_TERM <value>] [-
IN_DELAY <value>] [-OUT_DELAY <value>]
```

Sample command 1:

```
set_io -port_name EXT_RST_N \
-pin_name AD4 \
-fixed true \
-io_std LVCMOS15 \
-IN_DELAY 6 \
-LOCK_DOWN Yes \
-ODT ON \
-ODT_VALUE 240 \
-RES_PULL Down \
-SCHMITT_TRIGGER ON \
-DIRECTION INPUT
```

Sample command 2:

```
set_io -port_name TX \
-io_std LVCMOS12 \
-OUT_DELAY 8 \
-OUT_DRIVE 10 \
-RES_PULL Hold \
-DIRECTION OUTPUT
```

1.2. Wildcard Characters (Ask a Question)

You can use the following wildcard characters in names used in PDC commands.

Table 1-2. Wildcard Characters in Names Used in PDC Commands

Wildcard	What It Does
\	Interprets the next character.
?	Matches any single character.
*	Matches any string.

Note: The matching function requires that you add a slash (\) before each slash in the port, instance, or net name when using a wildcard character in a PDC command.

1.3. Special Characters ([ ], { }, and \) (Ask a Question)

Sometimes square brackets are part of the command syntax. In these cases, you must either enclose the open and closed square brackets characters with curly brackets or precede the open and closed square brackets characters with a backslash (\). Otherwise, you receive an error message.

For example:

```
set_io -port_name {P12}
```

Note: Do not add spaces between {}. For example, {PORT1} succeeds and { PORT1 } does not succeed.

1.4. Entering Arguments on Separate Lines (Ask a Question)

To enter an argument on a separate line, must enter a backslash (\) character at the end of the preceding line of the command, as shown in the following example.

```
set_io ADDOUT2 \
-iostd PCI \
-port_name \
-out_drive 16 \
-slew High \
-out_load 10 \
-pin_name T21 \
-fixed yes
```

2. PDC Naming Conventions (Ask a Question)

Note: The names of ports, instances, and nets in an imported netlist are sometimes referred to as their original names.

2.1. Rules for Displaying Original Names (Ask a Question)

Port names appear exactly as they are defined in a netlist.

Instances and nets display the original names plus an escape character (\) before each backslash (/), and each forward slash (\) is not a hierarchy separator. For example, the instance named A/\B is displayed as A\/\\B.

2.2. Which Name Do I Use in PDC Commands? (Ask a Question)

When writing PDC commands, follow these rules:

- Always use the macro name as it appears in the netlist.
- Names from a netlist:
  - For port names, use the names exactly as they appear in the netlist.
  - For instance and net names, add an escape character (\) before each backslash (\) and each forward slash (/) that is not a hierarchy separator.
- For wildcard names, always add an extra backslash (\) before each backslash.
- Always apply the PDC syntax conventions to any name in a PDC command.

The following table provides examples of names as they appear in an imported netlist and the names as they should appear in a PDC file.

Table 2-1. Sample Names in an Imported Netlist and PDC File

Name and Its Location	Name in the Imported Netlist	Name to Use in PDC File
Port name in a netlist	A/:B1	A/:B1
Instance name in a netlist	A/:B1 A$(1)	A\\/:B1 A$(1)
Instance name in the netlist but using a wildcard character in a PDC file	A/:B1	A\\\\/:B*
Net name in a netlist	Net1/:net1	Net1\\/:net1

When exporting PDC commands, the software always exports names using the PDC rules described in this section.

2.3. Case Sensitivity When Importing PDC Files (Ask a Question)

The following table lists the case sensitivity in the PDC file based on the source netlist.

Table 2-2. Case Sensitivity in the PDC File Based on a Source Netlist

File Type	Case Sensitivity
Verilog	Names in the netlist are case sensitive.
VHDL	Names in the netlist are not case sensitive unless those names appear between slashes (\).

For example, in VHDL, capital **A** and lowercase **a** are the same name, but **\A** and **\a** are two different names. However, in a Verilog netlist, an instance named **A10** fails, if spelled as **a10** in the `set_io` command:

`set_io -port_name A10 -pin_name W5` (This command succeeds).

3. I/O PDC Commands (Ask a Question)

I/O PDC commands are used to set and reset I/O standards, voltage values, and attributes.

For detailed information about I/Os and I/O standards, see PolarFire Family I/O User Guide.

3.1. set_iobank (Ask a Question)

This PDC command sets the input/output supply voltage (vcci) and the input reference voltage (vref) for the specified I/O bank.

All banks have a dedicated vref pin. Do not set any pin on these banks. There are two types of I/O banks:

- General-Purpose IO (GPIO)
- High-Speed IO (HSIO)

Setting the iobanks is recommended to ensure that banks are set as expected. If not set, following are the default bank settings:

- The bank will not have any settings if there is no I/O placed on the bank.
- The first I/O placed on the bank will be used to set the bank. Example: If we place an I/O with IOSTD LVCMOS25, the bank will be set to 2.5V.
- If I/Os are not placed by the user, the tool sets the banks as needed and place the I/Os.

The following table lists the `set_iobank` standards. Each bank type supports a different set of I/O standards.

Table 3-1. set_iobank Standards

I/O Types	Supported I/O Standards
HSIO	LVCMOS12, LVCMOS15, LVCMOS18, SSTL18I, SSTL18II, HSUL18I, HSUL18II, SSTL15I, SSTL15II, HSTL15I, HSTL15II, SSTL135I, SSTL135II, HSTL135I, HSTL135II, HSTL12I, HSTL12II, HSUL12I, SLVSE15, POD12I, POD12II, SLVS18, HCSL18, LVDS18, RSDS18, MINILVDS18, SUBLVDS18, PPDS18, SHIELD18, SHIELD15, SHIELD135, SHIELD12
GPIO	LVTTL, LVCMOS33, PCI, LVCMOS12, LVCMOS15, LVCMOS18, LVCMOS25, SSTL25I, SSTL25II, SSTL18I, SSTL18II, HSUL18I, HSUL18II, SSTL15I, SSTL15II, HSTL15I, HSTL15II, SLVS33, SLVS25, HCSL33, HCSL25, MIPI25, MIPIE25, LVPECL33, LVPECLE33, LVDS18G, LVDS25, LVDS33, RSDS25, RSDS33, MINILVDS25, MINILVDS33, SUBLVDS25, SUBLVDS33, PPDS25, PPDS33, SLVSE15, MLVDSE25, BUSLVDSE25, LCMDS33, LCMDS25, SHIELD33, SHIELD25, SHIELD18, SHIELD15, SHIELD12

The following example provides the usages of the `set_iobank` command with the supported arguments described.

```
set_iobank -bank_name <bank_name>\
[-vcci <vcci_voltage>]\
[-vref <vref_voltage>]\
[-fixed <value>]\
[-update_iostd <value>]\
[-auto_calib <value>]\
[-auto_calib_ramp_time <value>]
```

Arguments

-bank_name	Specifies the name of the bank. I/O banks are numbered 0 through N (Bank 0, Bank 1, ..., Bank N). The number of I/O banks varies with the device. See the data sheet for your device to determine how many banks it has.
-vcci <vcci_voltage>	Sets the input/output supply voltage. Enter one of the values in the following table.

Table 3-2. -vcci Values

Vcci Voltage	Compatible Standards
3.3V	LVTTL, LVCMOS33, PCI, LVDS33, LVPECL33, LVPECLE33, SLVS33, HCSL33, RSDS33, MINILVDS33, SUBLVDS33
2.5V	LVCMOS25, SSTL25I, SSTL25II, PPDS25, SLVS25, HCSL25, MLVDSE25, MINILVDS25, RSDS25, SUBLVDS25, LVDS25, MLVDSE25, BUSLVDSE25
1.8V	LVCMOS18, SSTL18I, SSTL18II, HSUL18I, HSUL18II, SLVS18, HCSL18, LVDS18, LVDS18G, RSDS18, MINILVDS18, SUBLVDS18, PPDS18
1.5V	LVCMOS15, SSTL15I, SSTL15II, HSTL15I, HSTL15II, SLVSE15
1.35V	HSTL135I, HSTL135II, SSTL135I, SSTL135II
1.2V	LVCMOS12, HSUL12I, HSTL12I, POD12I, MIPI12

-vref <vref_voltage> Sets the input reference voltage. Enter one of the values in the following table.

Table 3-3. -vref Values

Vref Voltage	Compatible Standards
1.25V	SSTL25I
1.0V	SSTL18I, HSUL18I
0.75V	POD12I, HSTL15I, SSTL15I, HSUL12I, HSTL12I
0.67V	SSTL135I, HSTL135I

-fixed <value> Specifies whether the I/O technologies (vcci and vccr voltages) assigned to the bank are locked. Enter one of the values in the following table.

Table 3-4. -fixed Values

Value	Description
True	The technologies are locked.
False	The technologies are not locked.

-update_iostd <value> For I/Os placed on the bank, this command specifies whether placement is kept, and the host is changed to one that is compatible with this bank setting.

Table 3-5. -update_iostd Values

Value	Description
True	If there are I/Os placed on the bank, you keep the placement and change the host to one that is compatible with this bank setting. Check the I/O attributes to see the one used by the tool.
False	If I/Os are placed and locked on the bank, the command fails. If they are placed I/Os, they are unplaced.

-auto_calib <value> Specifies whether the I/O bank is auto-calibrated at power-up or not. Values are true or false. The default value is true.

Note: This argument is not supported for MPF300TS_ES, MPF300T_ES, and MPF300XT devices.

-auto_calib_ramp_time <value> Specifies the I/O bank VDDI supply ramp time (in ms), if the I/O bank is auto-calibrated. Values are 1–50. The default value is 50.

Note: Not supported for MPF300TS_ES, MPF300T_ES, and MPF300XT devices.

Exceptions

Any pins assigned to the specified I/O bank that are incompatible with the default technology are unassigned.

Examples

The following example assigns 3.3V to the input/output supply voltage (vcci) for I/O bank 0.

```
set_iobank -bank_name bank0 -vcci 3.3
```

3.2. reserve (Ask a Question)

This PDC command reserves the named pins in the current device package.

```
reserve -pin_name "list of package pins"
```

Arguments

-pin_name list of package pins	Specifies the package pin name(s) to reserve. You can reserve one or more pins.

Exceptions

None

Examples

```
reserve -pin_name "F2"
reserve -pin_name "F2 B4 B3"
reserve -pin_name "124 17"
```

3.3. set_io (Ask a Question)

The `set_io` command assigns an I/O technology, places, or locks the I/O at a given pin location. Two I/O types are available for PolarFire:

- GPIO
- HSIO

Each I/O type supports different I/O standards.

Table 3-6. set_io Standards

I/O Types	Supported I/O Standards
HSIO	LVCMOS12, LVCMOS15, LVCMOS18, SSTL18I, SSTL18II, HSUL18I, HSUL18II, SSTL15I, SSTL15II, HSTL15I, HSTL15II, SSTL135I, SSTL135II, HSTL135I, HSTL135II, HSTL12I, HSTL12II, HSUL12I, SLVSE15, POD12I, POD12II, SLVS18, HCSL18, LVDS18, RSDS18, MINILVDS18, SUBLVDS18, PPDS18, LCMDS18, SHIELD18, SHIELD15, SHIELD135, SHIELD12
GPIO	LVTTL, LVCMOS33, PCI, LVCMOS12, LVCMOS15, LVCMOS18, LVCMOS25, SSTL25I, SSTL25II, SSTL18I, SSTL18II, HSUL18I, HSUL18II, SSTL15I, SSTL15II, HSTL15I, HSTL15II, SLVS33, HCSL33, HCSL25, MIPI25, MIPIE25, LVPECL33, LVPECL25, LVPECLE33, LVDS18G, LVDS25, LVDS33, RSDS25, RSDS33, MINILVDS25, MINILVDS33, SUBLVDS25, SUBLVDS33, PPDS25, PPDS33, SLVSE15, MLVDSE25, BUSLVDSE25, LCMDS33, LCMDS25, SHIELD33, SHIELD25, SHIELD18, SHIELD15, SHIELD12

Note: LCMDS18 IOSTD is not supported for MPF300XT devices.

The following is the syntax for this command:

```
set_io -port_name <port_name> \
       [-pin_name <package_pin>] \
       [-fixed <true|false>] \
       [-io_std <io_std_values>] \
       [-OUT_LOAD <value>] \
       [-RES_PULL <value>] \
       [-LOCK_DOWN <value>] \
       [-CLAMP_DIODE <value>] \
       [-SCHMITT_TRIGGER <value>] \
       [-SLEW <value>] \
       [-VCM_RANGE <value>] \
       [-ODT <value>] \
       [-ODT_VALUE] \
       [-OUT_DRIVE <value>] \
       [-IMPEDANCE <value>] \
       [-SOURCE_TERM <value>] \
```

```
        [-IN_DELAY <value>] \
        [-OUT_DELAY <value>]
```

Arguments

-port_name <port_name>

Specifies the port name of the I/O macro.

-pin_name <package_pin>

Specifies the package pin name(s) on which to place the I/O.

-io_std <value>

Sets the I/O standard for this macro. If the voltage standard used with the I/O is not compatible with other I/Os in the I/O bank, assigning an I/O standard to a port invalidates its location and automatically unassigns the I/O.

The following table lists the supported values for `-io_std` and their corresponding I/O standards. Some I/O standards support only single I/O or differential I/Os, while others support both single and differential I/Os.

Table 3-7. -io_std Values and Corresponding I/O Standards

-io_std Value	I/O Standard	
	Single	Differential
LVTTL	YES	NO
LVCMOS33	YES	NO
LVCMOS25	YES	NO
LVCMOS18	YES	NO
LVCMOS15	YES	NO
LVCMOS12	YES	NO
PCI	YES	NO
POD12I	YES	YES
POD12II	YES	YES
PPDS33	NO	YES
PPDS25	NO	YES
PPDS18	NO	YES
SLVS33	NO	YES
SLVS25	NO	YES
SLVS18	NO	YES
HCSL33	NO	YES
HCSL25	NO	YES
HCSL18	NO	YES
SLVSE15	NO	YES
BUSLVDSE	NO	YES
BUSLVDSE25	NO	YES
MLVDSE	NO	YES
MLVDSE25	NO	YES
LVDS	NO	YES
LVDS25	NO	YES
LVDS18	NO	YES
LVDS18G	NO	YES
BUSLVDS	NO	YES
BUSLVDSE25	NO	YES
MLVDS	NO	YES
MIPI25	NO	YES

Table 3-7. -io_std Values and Corresponding I/O Standards (continued)

-io_std Value	I/O Standard	
	Single	Differential
MIPIE25	NO	YES
MIPIE33	NO	YES
MINILVDS	NO	YES
MINILVDS33	NO	YES
MINILVDS25	NO	YES
MINILVDS18	NO	YES
RSDS	NO	YES
RSDS33	NO	YES
RSDS25	NO	YES
RSDS18	NO	YES
LVPECL (only for inputs)	NO	YES
LVPECL33	NO	YES
LVPECLE33	NO	YES
HSTL15I	YES	YES
HSTL15II	YES	YES
HSTL135I	YES	YES
HSTL135II	YES	YES
HSTLI2I	YES	YES
HSTL12II	YES	YES
SSTL18I	YES	YES
SSTL18II	YES	NO
SSTL15I	YES	YES
SSTL15II	YES	NO
SSTL135I	YES	YES
SSTL135II	YES	YES
SSTL25I	YES	YES
SSTL25II	YES	YES
HSUL18I	YES	YES
HSUL18II	YES	YES
HSUL12I	YES	YES
HSUL12II	YES	YES
SUBLVDS33	NO	YES
SUBLVDS25	NO	YES
SUBLVDS18	NO	YES
LCMDS33	NO	YES
LCMDS25	NO	YES
LCMDS18	NO	YES
SHIELD33	YES	NO
SHIELD25	YES	NO
SHIELD18	YES	NO
SHIELD15	YES	NO
SHIELD135	YES	NO
SHIELD12	YES	NO

-fixed <value> Specifies whether the location of this port is fixed (that is, locked). Locked ports are not moved during layout. The default value is true. You can enter one of the following values.

Table 3-8. -fixed Values

Value	Description
True	The location of this port is locked.
False	The location of this port is unlocked.

-OUT_LOAD <value> Sets the output load (in pF) of output signals. The default is 5 pF.

Direction: Output

-RES_PULL <value> Allows you to include a weak resistor for either pull-up or pull-down of the input and output buffers. Not all I/O standards have a selectable resistor pull option.

The following table lists the acceptable values for the `-RES_PULL` attribute for the input buffer.

Table 3-9. -RES_PULL Values (Input Buffer)

I/O Standards	Value	Description
Single I/Os: LVTTL, LVCMOS33, LVCMOS25, LVCMOS18, LVCMOS15, LVCMOS12, PCI	Up	Includes a weak resistor for pull-up of the input buffer.
	Down	Includes a weak resistor for pull-down of the input buffer.
	Hold	Holds the last value.
	None	Does not include a weak resistor.
Differential I/Os: PPDS25, PPDS33, HCSL33, HCSL25, BUSLVDSE25, LVDS33, LVDS25, LVDS18G, MINILVDS33, MINILVDS25, RSDS33, RSDS25, LVPECL33, SUBLVDS33, SUBLVDS25, LCMDS33, LCMDS25	Up	Includes a weak resistor for pull-up of the input buffer.
	Down	Includes a weak resistor for pull-down of the input buffer.

For I/O standards in the preceding table, the default is Up. For all other I/O standards, the value is None.

The following table lists the acceptable values for the `-RES_PULL` attribute for the output buffer.

Table 3-10. -RES_PULL Values (Output Buffer)

I/O Standards	Value	Description
Single I/Os: LVTTL, LVCMOS33, LVCMOS25, LVCMOS18, LVCMOS15, LVCMOS12, PCI	Up	Includes a weak resistor for pull-up of the output buffer.
	Down	Includes a weak resistor for pull-down of the output buffer.
	None	Does not include a weak resistor.

Notes:

- For all I/O standards, the default value for output buffer is None.
- For output differential I/Os, `res_pull` is not supported.

Direction: Inout

-LOCK_DOWN <value> Security feature that locks down the I/Os, if tampering is detected. Values are ON and OFF. The default is OFF.

Direction: Inout

-CLAMP_DIODE <value>

Specifies whether to add a power clamp diode to the I/O buffer. This attribute option is available to all I/O buffers with I/O technology set to LVTTL. A clamp diode provides circuit protection from voltage spikes, surges, electrostatic discharge, and other overvoltage conditions.

Values are OFF and ON.

The following table lists the values for GPIO standards. For HSIO standards, the value is always ON.

Table 3-11. -CLAMP_DIODE Values

I/O Standards	Values
LVCMOS12, LVCMOS15, LVCMOS18, SSTL18I, SSTL18II, SSTL15I, SSTL15II, HSTL15I, HSTL15II, LVTTL, LVCMOS33, LVCMOS25, SSTL25I, SSTL25II, SLVS25, HCSL25, LVDS25, RSDS25, MINILVDS25, SUBLVDS25, PPDS25, LCMDS25	OFF, ON. The default is ON.
MIPI25, LVDS18G	OFF, ON. The default is OFF.
HSUL18I, HSUL18II, SLVSE15, PCI, SLVS25, SLVS33, HCSL33, HCSL25, MIPIE33, MIPIE25, LVPECL33, LVPECL25, LVPECLE33, LVDS25, LVDS33, RSDS25, RSDS33, MINILVDS25, MINILVDS33, SUBLVDS25, SUBLVDS33, PPDS25, PPDS33, MLVDSE25, BUSLVDSE25, LCMDS25, LCMDS33	ON

Direction: Inout

Note: For input LVDS18G, the only supported value for Clamp Diode is OFF.

-SCHMITT_TRIGGER <value>

Specifies whether this I/O has an input Schmitt Trigger. The Schmitt Trigger introduces hysteresis on the I/O input. This allows very slow moving or noisy input signals to be used with the part without false or multiple I/O transitions taking place in the I/O.

For the following I/O standards, the values are OFF and ON. The default is OFF.

Table 3-12. -SCHMITT_TRIGGER Values

I/O Standards	Values
GPIO	
LVCMOS25, LVCMOS33, LVTTL, PCI	OFF, ON
HSIO	
LVCMOS18, LVCMOS15	OFF, ON

For all other I/O standards, the value is OFF.

Direction: Input

-SLEW <value>

Sets the output slew rate. Slew control affects only the falling edges for some families. Slew control affects both rising and falling edges. Not all I/O standards have a selectable slew. Whether you can use the slew attribute depends on which I/O standard you have specified for this command.

The following I/O standards have values OFF and ON. The default is OFF.

Table 3-13. -SLEW Values

I/O Standards	Values
LVCMOS25, LVCMOS33, LVTTL, PCI	OFF, ON

For all other I/O standards, the value is OFF.

Direction: Output

-VCM_RANGE <value>

Sets the VCM input range.

The following table lists the supported values and I/O standards.

Table 3-14. -VCM_RANGE Values

I/O Standards	Values
GPIO	
HSTL15I, HSTL15II, HSUL18I, HSUL18II, SSTL15I, SSTL15II, SSTL18I, SSTL18II, SSTL25I, SSTL25II	MID
HCSL33, HCSL25, LVDS18G, LVDS33, LVDS25, LVPECL33, LVPECLE33, MINILVDS33, MINILVDS25, MIPI25, MIPIE25, MLVDSE25, PPDS33, PPDS25, RSDS33, RSDS25, SLVS33, SLVS25, SLVSE15, BUSLVDSE25, SUBLVDS33, SUBLVDS25 **Note:** While assigning VCM input range for true differential I/Os in the same bank, a mix of MID and LOW values cannot be assigned for the I/Os. You can assign only MID or only LOW values for all differential I/Os in the same bank. If a mixture of VCM input ranges is needed for true differential inputs within the same GPIO bank, to optimize or tune the interface performance in hardware, the I/O attribute can be set using the Post Layout I/O Editing flow in the Libero Design Flow User Guide, and the `edit_io` command. For more information, see edit_io. In that scenario, the internal 100 Ohm on-die differential termination resistor accuracy percentage tolerance will follow the maximum percentage tolerance of the two ranges per the device datasheet specifications.	MID, LOW. The default is MID.
LCMDS33, LCMDS25	LOW
HSIO	
HSTL12I, HSTL12II, HSTL135I, HSTL135II, HSTL15I, HSTL15II, HSUL12I, HSUL18I, HSUL18II, LVSTL11I, LVSTL11II, POD12I, POD12II, SSTL135I, SSTL135II, SSTL15I, SSTL15II, SSTL18I, SSTL18II	MID
SLVSE15, LVDS18, HCSL18, MINILVDS18, PPDS18, RSDS18, SLVS18, SUBLVDS18	MID, LOW. The default is MID.
LCMDS18	LOW

Direction: Input

-ODT <value>

On-die Termination (ODT) is the technology where the termination resistor for impedance matching in transmission lines is located inside a semiconductor chip instead of on a printed circuit board.

In case of LVDS fail-safe mode, use the DYNAMIC value for ODT.

Values are OFF and ON.

The following table lists acceptable values.

Table 3-15. -ODT Values

I/O Standards	Values
LVCMOS12, LVCMOS15, LVCMOS18, LVCMOS25	OFF, ON. The default is OFF.
HSUL18I, HSUL18II	OFF, ON, DYNAMIC. The default is OFF.
SSTL15I, SSTL15II, SSTL18I, SSTL18II, HSUL12I, LVSTL11I, LVSTL11II, POD12I, POD12II, SSTL135I, SSTL135II, HSTL15I, HSTL15II, LVDS18G, LVDS33, LVDS25, LVPECL33, LVPECLE33, LVPECL25, MINILVDS33, MINILVDS25, RSDS33, RSDS25, SUBLVDS33, SUBLVDS25, HSTL12I, HSTL12II, HSTL135I, HSTL135II, LCMDS33, LCMDS25	OFF, ON, DYNAMIC. The default is ON.

Direction: Input

-ODT_VALUE

Sets the ODT value in Ohms.

Values vary depending on the I/O standard. The following table lists acceptable values.

Table 3-16. -ODT_VALUE Values

I/O Standards	Values
LVCMOS12, LVCMOS15, LVCMOS18	120, 240. The default is 120.
LVCMOS25	120
HSUL12I	60, 120, 240. The default is 120.
SSTL15I, SSTL15II	20, 30, 40, 60, 120. The default is 30.
SSTL135I, SSTL135II	20, 30, 40, 60, 120. The default is 40.
SSTL18I, SSTL18II	50, 75, 150. The default is 50.
POD12I, POD12II	34, 40, 48, 60, 80, 120, 240. The default is 60.
LVDS18G, LVDS33, LVDS25, LVPECL33, LVPECL25, MINILVDS33, MINILVDS25, RSDS33, RSDS25, SLVSE15, SUBLVDS33, SUBLVDS25, LCMDS33, LCMDS25	100
HSTL15I, HSTL15II, HSUL18I, HSUL18II, HSTL12I, HSTL12II, HSTL135I, HSTL135II	50

Direction: Inout

-OUT_DRIVE <value>

Sets the strength of the output buffer to 1.5, 2, 3.5, 4, 6, 8, 10, 12, 16, or 20 in mA, weakest to strongest. The list of I/O standards for which you can change the output drive and the list of values you can assign for each I/O standard is family-specific. Not all I/O standards have a selectable output drive strength.

Each I/O standard has a different range of legal output drive strength values. The values you can choose depend on which I/O standard you specified for this command. The following table lists the acceptable values.

Table 3-17. -OUT_DRIVE Values

I/O Standard	Values
GPIO	
LVCMOS12	2, 4, 6, 8. Default is 8.
LVCMOS15	2, 4, 6, 8, 10. Default is 8.
LVCMOS18	2, 4, 6, 8, 10, 12. Default is 8.
LVCMOS25	2, 4, 6, 8, 12, 16. Default is 8.
LVCMOS33, LVTTL	2, 4, 8, 12, 16, 20. Default is 8.
LVDS18G, LVDS25, LVDS33, MINILVDS25, MINILVDS33, LCMDS33, LCMDS25	3, 3.5, 4, 6. Default is 6.
PPDS25, PPDS33, RSDS25, RSDS33	1.5, 2, 3. Default is 3.
SUBLVDS25, SUBLVDS33	1, 1.5, 2. Default is 2.
BUSLVDSE25, MLVDSE25, LVPECLE33	16
MIPIE25, SLVSE15	8
PCI	20
HSIO	
LVCMOS12, LVCMOS15	2, 4, 6, 8, 10. Default is 8.
LVCMOS18	2, 4, 6, 8, 10, 12. Default is 8.
SLVSE15	8

Direction: Output

-IMPEDANCE Sets the impedance value in Ohms.

Values vary depending on the I/O standards.

Direction: Output

-SOURCE_TERM Near-end termination for a differential output I/O. The following table lists the acceptable values.

Table 3-18. -SOURCE_TERM Values

I/O Standards	Values
LVDS18G, LVDS25, LVDS33, MINILVDS25, MINILVDS33, LCMDS33, LCMDS25, PPDS25, PPDS33, RSDS25, RSDS33, SUBLVDS25, SUBLVDS331	OFF, 100. The default is OFF.

Direction: Output

-IN_DELAY Sets the static input delay.

Input delay applies to all I/O standards. The values are OFF and 0–127, 128, 130, 132,...254. The default value is OFF.

Direction: Input

Notes:

- This attribute does not appear in the I/O attributes and cannot be used in the PDC for some I/Os with dynamic delays, such as DDR, LPDDR, QDR memory interfaces and I/O CDR interfaces.
- For the dynamically delayed I/O interfaces that also support setting a static delay value, this attribute will be listed as **OFF** if a static delay has not been added. You can still adjust the dynamic delays during normal operation, even if the static delay is listed as **OFF**.

-OUT_DELAY Sets the static output delay.

Output delay applies to all I/O standards. The values are OFF and 0–127. The default value is OFF.

Direction: Output

Notes:

- This attribute does not appear in the I/O attributes and cannot be used in the PDC for some I/Os with dynamic delays, such as DDR, LPDDR, QDR memory interfaces and I/O CDR interfaces.
- In case of LVDS fail-safe mode, you can use the following PDC commands:
  - Weak pull-up/pull-down resistor for differential inputs.

    ```
    set_io -RES_PULL <value>
    ```

  - Dynamic ODT access per I/O.

    ```
    set_io -ODT DYNAMIC
    ```

- For the dynamically delayed I/O interfaces that also support setting a static delay value, this attribute will be listed as **OFF** if a static delay has not been added. You can still adjust the dynamic delays during normal operation, even if the static delay is listed as **OFF**.

Examples

```
set_io -port_name IO_in\[2\]
-io_std LVCMOS25 \
-fixed true\
```

I/O Directions Not Supported

The following table lists the I/O directions that are not supported for I/O standards.

Table 3-19. I/O Directions That are Not Supported (set_io)

I/O Direction	IO_STD Value
Input	SLVSE15, MLVDSE25, BUSLVDSE25, MIPIE33, LVPECLE33, SHIELD33, SHIELD25, SHIELD18, SHIELD15, SHIELD135, SHIELD12
Output	SLVS33, SLVS25, HCSL33, HCSL25, LVPECL33, LVPECL25, MIPI25, LVDS18, RSDS18, MINILVDS18, SUBLVDS18, PPDS18, SLVS18, HCSL18, LCMDS18
Tribuff	SLVS33, SLVS25, HCSL33, HCSL25, LVPECL33, LVPECL25, MIPI25, LVDS18, RSDS18, MINILVDS18, SUBLVDS18, PPDS18, SLVS18, HCSL18, LVDS18G, LVDS25, LVDS33, RSDS25, RSDS33, MINILVDS25, MINILVDS33, SUBLVDS25, SUBLVDS33, PPDS25, PPDS33, LCMDS25, LCMDS33, LCMDS18
Inout	LVDS18G, LVDS33, LVDS18, LVDS25, RSDS18, RSDS33, RSDS25, MINILVDS18, MINILVDS33, MINILVDS25, SUBLVDS18, SUBLVDS33, SUBLVDS25, PPDS18, PPDS33, PPDS25, SLVS33, SLVS25, HCSL33, HCSL25, LVPECL33, LVPECL25, MIPI25, MIPIE25, SLVS18, HCSL18, SHIELD33, SHIELD25, SHIELD18, SHIELD15, SHIELD135, SHIELD12, LCMDS25, LCMDS33, LCMDS18

3.4. set_location (Ask a Question)

This PDC command assigns the specified macro to a particular location on the chip.

```
set_location -inst_name <macro_inst_name> -fixed <true|false> -x <integer> -y <integer>
```

Note: This command may not honor placing globals to physical locations on the die. Instead of placing globals on die locations, let the Libero Design Suite decide where to place the global buffers.

Arguments

-inst_name Specifies the instance name of the macro in the netlist to assign to a particular location on the chip.

-fixed <true | false> Sets whether the location of this instance is fixed (that is, locked). Locked instances are not moved during layout. The default is YES.

The following table lists the acceptable values for this argument.

Table 3-20. -inst_name Values

Value	Description
True	The location of this instance is locked.
False	The location of this instance is unlocked.

-x -y The x and y coordinates specify where to place the macro on the chip. Use the Chip Planner tool to determine the x and y coordinates of the location.

Exceptions

None

Example

This example assigns and locks the macro with the name `mem_data_in\[57\]` at the location x = 7, y = 2:

```
set_location -inst_name mem_data_in\[57\] -fixed true -x 7 -y 2
```

DDR3 Memory Placement

DDR3 memory must be placed in specific locations on the PolarFire chip to meet timing requirements. For DDR3 memory placement, the `set_location` command has the following syntax:

```
set_location -inst_name <hierarchical path to DDR instance> -location <edge>_<anchor>
```

-inst_name <hierarchical path to DDR instance>	Specifies the hierarchical path to the DDR instance.
-location <edge>_<anchor>	Specifies the `edge_anchor` location.

Example

```
set_location -inst_name {DDR3_TOP/DDR3_0}\ -location {NORTH_NE}
```

The maximum DDR width varies with the die/package combinations and the location they are placed in. See the following table for the correct location to place the DDR3 memory. The numbers in the table refer to the maximum DDR3 width.

Table 3-21. Locations for Placing DDR3 Memory

Die/Package	Location (Edge_Anchor} Edge={NORTH/SOUTH/WEST}, Anchor={NE/NW/SE/SW}					
	NORTH_NE	NORTH_NW	SOUTH_SE	SOUTH_SW	WEST_NW	WEST_SW
MPF200/FULLPKGE	16	16	Invalid Loc	40	64	40
MPF300/FCG1152	64	72	16	40	72	64
MPF300/FCG484	8	8	Invalid Loc	32	Invalid Loc	16
MPF300/FCVG484	16	16	Invalid Loc	40	16	16

PLL Placement

For PLL placement, the `set_location` command has the following syntax:

```
set_location -inst_name <hierarchical inst name> -location <PLL location>
```

-inst_name <hierarchical inst name>	Specifies the hierarchical instance name.
-location <PLL location>	Specifies the PLL location. Location can be one of the following: • PLL0_NW • PLL1_NW • PLL0_NE • PLL1_NE • PLL0_SW • PLL1_SW • PLL0_SE • PLL1_SE For more information, see Placement Rules for PLLs and DLLs.

DLL Placement

For DLL placement, the `set_location` command has the following syntax:

```
set_location -inst_name <hierarchical inst name> -location <DLL location>
```

-inst_name <hierarchical inst name>	Specifies the hierarchical instance name.
-location <DLL location>	Specifies the DLL location. Location can be one of the following: • DLL0_NW • DLL1_NW • DLL0_NE • DLL1_NE • DLL0_SW • DLL1_SW • DLL0_SE • DLL1_SE For more information, see Placement Rules for PLLs and DLLs.

TxPLL Placement

For TxPLL placement, the `set_location` command has the following syntax:

```
set_location -inst_name <hierarchical inst name> -location <TxPLL location>
```

-inst_name <hierarchical inst name>	Specifies the hierarchical instance name.
-location <TxPLL location>	Specifies the TxPLL location. Location can be one of the following: • Q2_TXPLL0 • Q2_TXPLL_SSC • Q2_TXPLL1 • Q0_TXPLL0 • Q0_TXPLL_SSC • Q0_TXPLL1 • Q1_TXPLL0 • Q1_TXPLL_SSC • Q1_TXPLL1 • Q3_TXPLL_SSC • Q3_TXPLL1 For more information, see Placement Rules for Transceivers.

Placement Rules for PLLs and DLLs

The following table lists the error messages that indicate non-compliance with placement rules for PLL and DLL.

Table 3-22. Error Messages

Error Code	Error Message	Description
PRPF_010	There can be a maximum of 6 PLL/DLL reference and/or fabric clocks coming driven by the FPGA fabric in the <NW|SW|NE|SE> location.	There are four corners (NW, SW, NE, SE) that PLL and DLL instances can be placed in on each MPF300 or MPF200 FPGA device. You can place multiple PLL/DLL instances in each corner. However, for each corner, the total of PLL/DLL reference clocks and fabric clocks that the fabric drives must be six or less.

Table 3-22. Error Messages (continued)

Error Code	Error Message	Description
PRPF_011	There can be a maximum of 2 PLL/DLL reference clocks coming driven by the FPGA fabric in the <NW|SW|NE|SE> location.	For each corner, only two PLL/DLL reference clocks can be driven by the fabric.

Placement Rules for RGMII, SGMII, and IOG CDR Interfaces

Placement rules must be adhered to for RGMII, SGMII, and IOG CDR interfaces. Non-compliance with these rules may result in the following errors.

Table 3-23. Error Messages

Error Code	Error Message	Description
PRPF_001	Port <port name> for Interface <inst name> must be placed before running Place-and-Route.	All PADs must be placed using the `set_io` command.
PRPF_002	Interface <inst name> has ports that must be assigned to the same physical lane. The current port assignment for this interface does not meet this requirement.	For the SGMII interface and IOG CDR, all RX_ and TX_ PADs must be placed in the same lane. For the RGMII interface, all RX [ ] PADs and the RXCLK PAD must be placed in the same lane.
PRPF_003	The current Interface <inst name> port assignment requires that pin <pin name (functional pin name)> is reserved. You must not assign any port to that package pin.	For the SGMII interface and IOG CDR, the `DQS_N` pin of the lane is reserved for internal use. It must be left unused.
PRPF_004	You must not assign <inst name> to any location. Use the set_io command to assign any interface port to package pins. This instance is automatically placed.	IOD instances with TRAINING/OVERLAY must not be placed. These are internal instances and will be handled by the tool.
PRPF_005	Port <port name> for Interface <inst name> must be assigned to <pin name (functional pin name)>.	For the RGMII interface, `RX_CLK` must be assigned to the DQS (P pad) of the lane.

Placement Rules for Transceivers

For PolarFire designs with the transceiver (XCVR) interface, some placement rules apply. Non-compliance with these rules may result in the following errors. For more information about rules for transceivers, see PolarFire Family Transceiver User Guide.

Table 3-24. Error Messages

Error Code	Error Message	Description
PRPF_007	TxPLL <inst name> must be placed before running Place-and-Route.	Transceiver Tx PLLs must be placed by the user with the `set_location` command before running Place-and-Route.
PRPF_008	Dedicated XCVR ports <port name>* must be placed before running Place-and-Route.	The transceiver interface has dedicated ports. These must be placed using the `set_io` command.
PRPF_009	Dedicated XCVR reference clock port <port name> must be placed before running Place -and-Route.	All transceiver reference clock PADs must be placed using the `set_io` command before running layout.
PRPF_008	Dedicated XCVR ports <port name>* must be placed before running Place & Route.	The transceiver interface has dedicated ports. These must be placed using the `set_io` command.
PRPF_009	Dedicated XCVR reference clock port <port name> must be placed before running Place & Route.	All transceiver reference clock PADs must be placed using the `set_io` command before running layout.
PRPF_009	Dedicated XCVR reference clock port <port name> must be placed before running Place & Route.	The transceiver interface has dedicated ports. These must be placed using the `set_io` command.

4. Netlist Attributes NDC Commands (Ask a Question)

To set netlist-specific constraints, use Netlist attributes Netlist Design Constraint (NDC) commands. These commands are placed in a Compile Netlist Constraint (`*.ndc`) file and used by the Libero SoC Compile engine to optimize the post-synthesis netlist.

4.1. set_ioff (Ask a Question)

This command specifies whether or not a register is combined with an I/O during Synthesis. I/Os are combined with a register to achieve better clock-to-out and input-to-clock timing. This command is placed in a Compile Netlist Constraint (`*.ndc`) file that the Constraint Manager passes to Synthesis as a constraint in the Libero SoC Constraint Flow.

```
set_ioff -port_name <portname> \ [-IN_REG true/1|false/0] \
[-OUT_REG true/1|false/0] \ [-EN_REG true/1|false/0]
```

Important: This command is supported only as an NDC file to be read before Compile/Synthesis. At least one of the preceding options must be specified to use this command. Only one option can be enabled at a time. Microchip supports combining only one Flip-Flop with the I/O, and the fanout must be 1.

Arguments

-port_name <portname>	Specifies the name of the I/O port to be combined with a register. The port can be an input, output, or inout port. Wildcard characters are supported.
-IN_REG	Specifies whether the input register is combined into the port <portname>. Valid values are true/1 or false/0.
-OUT_REG	Specifies whether the output register is combined into the port <portname>. Valid values are true/1 or false/0.
-EN_REG	Specifies whether the enable register is combined into the port <portname>. Valid values are true/1 or false/0.

Example

The following example is applicable for the PolarFire, RT PolarFire, and PolarFire SoC family of devices:

```
set_ioff -port_name {D} -IN_REG <value> -OUT_REG <value> -EN_REG <value>
set_ioff -port_name {Q} -IN_REG <value>-OUT_REG <value>-EN_REG <value>

<value> can be either true/false, or 1/0.
```

The following example is applicable for the SmartFusion® 2, IGLOO® 2, and RTG4™ family of devices:

```
set_ioff {D} -IN_REG <value> _ -OUT_REG_ <value> _ -EN_REG_ <value>
set_ioff {Q} -IN_REG <value> -OUT_REG <value> -EN_REG <value>

<value> can be yes/no.
```

The `set_ioff` command applies to scalar I/Os only. For an I/O bus, use the 'for' loop available in Tcl. The following command combines each scalar member of the 32-bit I/O bus DataA with input registers:

```
for { set i 0 } { i &lt; 32 } { incr i } { set_ioff -port_name "DataA\[$i\]" -IN_REG 1 }
```

Alternatively, you can use a wildcard to include all scalar signals of an I/O bus:

```
set_ioff -port_name {DataA[*]} -IN_REG 1
```

Return Value

The command returns 0 on success and 1 on failure.

Error Messages

The following error messages are related to this command:

```
Error: [19138170]: PDCPF-426: IN_REG: Invalid argument value: 'yes' (expecting TRUE, 1,
true, FALSE, 0 or false).[set_ioff -port_name PAD -IN_REG yes][[D:/designs/ test_ioff_ioedit/
constraint/test.ndc]
Error: [19137989]: PDCPF-01: Port name doesn't exist in the netlist or is not connected to an
IoCell macro. [set_ioff -port_name PAD253 -IN_REG 1]
Error: [19138170]: PDCPF-426: Required parameter 'port_name' is missing. [set_ioff][[D:/
designs/test_ioff_ioedit/constraint/test.ndc]
```

4.2. set_preserve (Ask a Question)

This command sets a preserve property on instances before compile. Therefore, compile preserves these instances and does not combine them.

```
set_preserve -inst_name <hier_inst_name>
```

Arguments

-inst_name Specifies the full hierarchical name of the macro in the netlist to preserve.

Example

```
set_preserve -inst_name “test1/AND2_0”
```

5. Floorplanning FDC Commands (Ask a Question)

Floorplanning (FDC) commands are used to create and edit user regions and to assign/unassign logic to these regions.

5.1. assign_region (Ask a Question)

This PDC command constrains a set of macros to a specified region.

```
assign_region -region_name <region_name> -inst_name <macro_name>+
```

Arguments

-region_name Specifies the region to which the macros are assigned. The macros are constrained to this region. Because the `define_region` command returns a region object, you can write a simpler command such as `assign_region [define_region]+ [macro_name]+`.

-inst_name Specifies the macro(s) to assign to the region. You must specify at least one macro name. The following table lists the wildcard characters you can use in macro names.

Table 5-1. Supported Wildcard Characters in Macro Names

Wildcard	What It Does
\	Interprets the next character as a non-special character
?	Matches any single character
*	Matches any string

Important:

- The region must be created before you can assign macros to it. If the region creation PDC command and the macro assignment command are in different PDC files, the order of the PDC files is important.
- You can assign only hard macros or their instances to a region. You cannot assign a group name. A hard macro is a logic cell consisting of one or more silicon modules with locked relative placement.
- The macro name must be a name with full hierarchical path.

Examples

In the following example, two macros are assigned to a region:

```
assign_region -region_name UserRegion1 -inst_name "test_0/AND2_0 test_0/AND2_1"
```

In the following example, all macros whose names have the prefix des01/Counter_1 (or all macros whose names match the expression des01/Counter_1/*) are assigned to a region:

```
assign_region -region_name User_region2 -inst_name des01/Counter_1/*
```

5.2. assign_net_macros (Ask a Question)

This PDC command assigns to a user-defined region all the macros that are connected to a net.

```
assign_net_macros -region_name <region_name> -net_name <net_name> -include_driver <true|false>
```

Arguments

-region_name Specifies the name of the region to which you assign macros. The region must exist before you use this command. See `define_region` (rectangular) or `define_region`

(rectilinear). Because the `define_region` command returns a region object, you can write a simple command such as `assign_net_macros [define_region]+ [net]+`.

-net_name

You must specify at least one net name. Net names are AFL-level (flattened netlist) names. These names match your netlist names most of the time. When they do not, you must export AFL and use the AFL names. Net names are case insensitive. Hierarchical net names from ADL are not allowed.

The following table lists the wildcard characters you can use in net names.

Table 5-2. Supported Wildcard Characters in Net Names

Wildcard	What It Does
\	Interprets the next character as a non-special character
?	Matches any single character
*	Matches any string

-include_driver

Specifies whether to add the driver of the net(s) to the region. Enter one of the values in the following table.

Table 5-3. -include_driver Values

Value	Descriptions
True	Includes the driver in the list of macros assigned to the region (default).
False	Do not assign the driver to the region.

Observe the following guidelines and see define_region for more information:

- Placed macros (not connected to the net) that are inside the area occupied by the net region are automatically unplaced.
- Net region constraints are internally converted into constraints on macros. PDC export results as a series of assign_region <region_name> macro1 statements for all the connected macros.
- If the region does not have enough space for all of the macros, or if the region constraint is impossible, the constraint is rejected and a warning message appears in the Log window.
- For overlapping regions, the intersection must be at least as big as the overlapping macro count.
- If a macro on the net cannot legally be placed in the region, it is not placed and a warning message appears in the Log window.
- Net region constraints may result in a single macro being assigned to multiple regions. These net region constraints result in constraining the macro to the intersection of all the regions affected by the constraint.

5.3. define_region (Ask a Question)

This PDC command defines either a rectangular region or a rectilinear region.

```
define_region -region_name <region_name> -type <inclusive|exclusive|empty> -x1 <integer> -y1
<integer> -x2 <integer> -y2 <integer> [-route <true|false>]
```

Note: The -route parameters are optional. To define region colors, use the Display option in the Chip Planner.

Arguments

-region_name <region_name>

Specifies the region name. The name must be unique. Do not use reserved names such as Bank 0 and Bank <N> for region names. If the region cannot be created, the name is empty. A default name is generated, if a name is not specified in this argument.

-type <inclusive | exclusive | empty>

Specifies the region type. The following table shows the acceptable values for this argument.

Table 5-4. -type Values

Region Type	Description
Empty	Empty regions cannot contain macros
Exclusive	Only contains macros assigned to the region
Inclusive	Can contain macros both assigned and unassigned to the region

-x1 -y1 -x2 -y2

Specifies the series of coordinate pairs that constitute the region. These rectangles may or may not overlap. They are given as x1, y1, x2, y2 (where x1, y1 is the lower left and x2 y2 is the upper right corner in row/column coordinates). You must specify at least one set of coordinates.

-route <value>

Specifies whether to direct the routing of all nets internal to a region to be constrained within that region. A net is internal to a region, if its source and destination pins are assigned to the region.

The following table lists the acceptable values for this argument.

Table 5-5. -route Values

Constrain Routing Value	Description
True	Constrain the routing of nets within the region as well as the placement.
False	Do not constrain the routing of nets within the region. Only constrain the placement. This is the default value.

Note: Local clocks and global clocks are excluded from the `-route` option. Also, interface nets are excluded from the `-route` option because they cross region boundaries.

An empty routing region is an empty placement region. If `-route` is true, then no routing is allowed inside the empty region. However, local and global clocks can cross empty regions.

An exclusive routing region is an exclusive placement region (rectilinear area with assigned macros) along with the following additional constraints:

- For all nets that are internal to the region (the source and all destinations belong to the region), routing must be inside the region (that is, such nets cannot be assigned any routing resource which is outside the region or crosses the region boundaries).
- Nets without pins inside the region cannot be assigned any routing resource, which is inside the region or crosses any region boundaries.

An inclusive routing region is an inclusive placement region (rectilinear area with assigned macros) along with the following additional constraints:

- For all nets that are internal to the region (the source and all destinations belong to the region), routing must be inside the region (that is, such nets cannot be assigned any routing resource which is outside the region or crosses the region boundaries).
- Nets that are not internal to the region can be assigned routing resources within the region.

Description

Unlocked macros are unassigned in the empty or exclusive regions. You cannot create empty regions in areas that contain locked macros.

Use inclusive or exclusive region constraints, if you intend to assign a logic to a region. An inclusive region constraint with no macros assigned to it has no effect. An exclusive region constraint with no macros assigned to it is equivalent to an empty region.

Note: If macros assigned to a region exceed the area's capacity, the region's Properties Window displays the overbooked resources (over 100 percent resource usage) in red.

Examples

The following example defines an empty rectangular region called UserRegion1 with lower-left coordinates (100, 46) and upper-right coordinates (102, 50).

```
define_region -region_name UserRegion1 -type empty -x1 100 -y1 46 -x2 102 -y2 50
```

The following example defines an inclusive rectilinear region with the name UserRegion2. This region contains two rectangular areas, one with lower-left coordinates (12, 39) and upper-right coordinates (23, 41) and another rectangle with lower-left coordinates (12, 33) and upper-right coordinates (23, 35).

```
define_region -region_name UserRegion2 -type exclusive -x1 12 -y1 39 -x2 23 -y2 41 -x1 12 -y1
33\ -x2 23 -y2 35
```

The following examples define three regions with three different colors:

```
define_region -region_name UserRegion0 -color 128 -x1 50 -y1 19 -x2 60 -y2 25 define_region
-region_name UserRegion1 -color 16711935 -x1 11 -y1 2 -x2 55 -y2 29 define_region
-region_name UserRegion2 -color 8388736 -x1 61 -y1 6 -x2 69 -y2 19
```

For more information about regions, see assign_region.

5.4. move_region (Ask a Question)

This PDC command moves the named region to the specified coordinates.

```
move_region -region_name <region_name> -x1 <integer> -y1 <integer> -x2 <integer> -y2 <integer>
```

Arguments

-region_name	Specifies the name of the region to move. This name must be unique.
-x1 -y1 -x2 -y2	Specifies the series of coordinate pairs representing the location in which to move the named region. These rectangles can overlap. They are given as x1, y1, x2, and y2, where x1 and y1 represent the lower-left corner of the rectangle and x2 and y2 represent the upper-right corner. You must specify at least one set of coordinates.

Example

This example moves the region named UserRegion1 to a new region with lower-left coordinates (0, 40) and upper-right coordinates (3, 42):

```
move_region -region_name UserRegion1 -x1 0 -y1 40 -x2 3 -y2 42
```

For more information about regions, see define_region.

6. Post Layout Edit PDC Commands (Ask a Question)

Post Layout Edit PDC Commands are used when the design is in post-layout state.

6.1. edit_io (Ask a Question)

Use this PDC command to make the changes related to GPIO and HSIO type I/Os in the `edit_post_layout_design` tool.

Tip: This command is also supported by RTG4 family of devices.

```
edit_io -port_name <port_name>\
        [-OUT_LOAD <value>]\
        [-RES_PULL <value>]\
        [-LOCK_DOWN <value>]\
        [-CLAMP_DIODE <value>]\
        [-SCHMITT_TRIGGER <value>]\
        [-SLEW <value>]\
        [-VCIM_RANGE <value>]\
        [-ODT <value>]\
        [-ODT_VALUE <value>]\
        [-OUT_DRIVE <value>]\
        [-IMPEDANCE <value>]\
        [-SOURCE_TERM <value>]\
        [-IN_DELAY <value>]\
        [-OUT_DELAY <value>]
```

For PolarFire Transceiver type I/Os, `edit_io` supports the following attributes.

Important: RX_CALIBRATION is determined by the data rate and cannot be changed by the user using the PDC command.

```
edit_io  -port_name <port_name>\
         [-TX_EMPHASIS_AMPLITUDE <value>]\
         [-TX_IMPEDANCE <value>]\
         [-TX_TRANSMIT_COMMON_MODE_ADJUSTMENT <value>]\
         [-RX_INSERTION_LOSS <value>]\
         [-RX_CALIBRATION <value>]\
         [-RX_CTLE <value>]\
         [-RX_CDR_GAIN <value>]\
         [-RX_TERMINATION <value>]\
         [-RX_PN_BOARD_CONNECTION]\
         [-RX_LOSS_OF_SIGNAL_DETECTOR_LOW <value>]\
         [-RX_LOSS_OF_SIGNAL_DETECTOR_HIGH <value>]
         [-RX_DFE_COEFFICIENT_H1 <value>]\
         [-RX_DFE_COEFFICIENT_H2 <value>]\
         [-RX_DFE_COEFFICIENT_H3 <value>]\
         [-RX_DFE_COEFFICIENT_H4 <value>]\
         [-RX_DFE_COEFFICIENT_H5 <value>]\
         [-RX_POLARITY <value>]
```

Arguments

For arguments related to GPIO and HSIO type I/Os, see set_io.

Important: The arguments **-pin_name**, **-fixed**, and **-io_std** are not supported by **edit_io** PDC command.

The following are the arguments for PolarFire Transceiver type I/Os.

-TX_EMPHASIS_AMPLITUDE <value>

Adjusts the transmit emphasis and DC amplitude settings of the transmitter output drivers. The default value is 400mV_with_-1.0dB.

Direction: Output

Table 6-1. TX Emphasis Amplitude Values

Name	Values
TX_EMPHASIS_AMPLITUDE	100mV_with_0dB
	200mV_with_0dB
	200mV_with_-1.0dB
	200mV_with_-2.5dB
	200mV_with_-3.5dB
	200mV_with_-4.4dB
	200mV_with_-6.0dB
	300mV_with_0dB
	400mV_with_0dB
	400mV_with_-1.0dB
	400mV_with_-2.5dB
	400mV_with_-3.5dB
	400mV_with_-4.4dB
	400mV_with_-6.0dB
	500mV_with_0dB
	600mV_with_-3.5dB
	600mV_with_-6.0dB
	800mV_with_0dB
	800mV_with_-1.0dB
	800mV_with_-2.5dB
	800mV_with_-3.5dB
	800mV_with_-4.4dB
	800mV_with_-6.0dB
	1000mV_with_0dB
	1000mV_with_-1.0dB
	1000mV_with_-2.5dB
	1000mV_with_-3.5dB
	1000mV_with_-4.4dB
	1000mV_with_-6.0dB

-TX_IMPEDANCE <value>

Adds calibrated internal impedance onto the differential outputs. The default value is 100.

Direction: Output

Table 6-2. TX_IMPEDANCE Values

Name	Values
TX_IMPEDANCE	150
	100
	85
	180

-TX_TRANSMIT_COMMON_MODE_ADJUSTMENT <value>

Transmit Common-mode level is used as a percentage of full Common-mode level or VDDA. It is only adjusted when DC coupled. For AC coupled systems, the level must remain as default. The default value is 50.

Direction: Output

Table 6-3. TX_TRANSMIT_COMMON_MODE_ ADJUSTMENT Values

Name	Values
TX_TRANSMIT_COMMON_MODE_ ADJUSTMENT	50
	60
	70
	80

-RX_INSERTION_LOSS <value>

Sets the predefined settings used to statically adjust the receiver CDR and DFE. The default value is 6.5 dB.

Direction: Input

Table 6-4. RX_INSERTION_LOSS Values

Name	Values
RX_INSERTION_LOSS	6.5 dB
	17.0 dB
	25.0 dB

-RX_CALIBRATION <value>

For more information about Receiver Calibration, see PolarFire Family Transceiver User Guide.

Direction: Input

Table 6-5. RX_CALIBRATION Values

Name	Values
RX_CALIBRATION	None_CDR
	On Demand
	On Demand and First Lock
	None_DFE

-RX_CTLE

Sets the receiver equalization settings used to reduce the low-frequency component of the signal while boosting the high frequency component. The default value is set based on data-rate and Rx insertion loss model.

Direction: Input

For Rx CTLE Settings table, see AC483: PolarFire FPGA Transceiver Signal Integrity Application Note.

-RX_CDR_GAIN <value>

CDR Gain denotes the effect of Gain on Jitter. Low CDR gain denotes low CDR lock time and better jitter tolerance whereas High CDR gain denotes faster CDR lock time and high jitter.

Direction: Input

Table 6-6. RX_CDR_GAIN Values

Name	Values
RX_CDR_GAIN	Low
	High

-RX_TERMINATION <value>

Sets a calibrated input termination for available differential impedances within the Rx buffer. The default value is 100.

Direction: Input

Table 6-7. RX_TERMINATION Values

Name	Values
RX_TERMINATION	150
	100
	85

-RX_PN_BOARD_CONNECTION <value>

Sets the coupling type for PCB. The default value is AC_COUPLED_WITH_EXT _CAP.

Direction: Input

Table 6-8. RX_PN_BOARD_CONNECTION Values

Name	Values
RX_PN_BOARD_CONNECTION	AC_COUPLED_WITH_EXT_CAP
	DC_COUPLED

-RX_LOSS_OF_SIGNAL_DETECTOR_LOW <value>

Sets the lower set point for a Loss Of Signal (LOS) detector to ensure that a good signal is applied to the receiver. The default value is OFF.

Direction: Input

Table 6-9. RX_LOSS_OF_SIGNAL_DETECTOR_LOW Values

Name	Values
RX_LOSS_OF_SIGNAL_DETECTOR_LOW	OFF
	PCIE
	SATA
	BMR
	1
	2
	3
	4
	5
	6
	7

-RX_LOSS_OF_SIGNAL_DETECTOR_HIGH <value>

Sets the higher set point for an LOS detector to ensure that a good signal is applied to the receiver. The default value is OFF.

Direction: Input

Table 6-10. RX_LOSS_OF_SIGNAL_DETECTOR_HIGH Values

Name	Values
RX_LOSS_OF_SIGNAL_DETECTOR_HIGH	OFF
	PCIE
	SATA
	BMR
	1
	2
	3
	4
	5
	6
	7

-RX_DFE_COEFFICIENT_H1 <value> Sets the DFE coefficients for a design set in static mode. These attributes are optional and take integer values between 0 and 15. The corresponding register fields are 5 bits wide in all cases with the MSB bit reserved for sign bit.

The same values are applicable for `-RX_DFE_COEFFICIENT_H2`, `-RX_DFE_COEFFICIENT_H3`, `-RX_DFE_COEFFICIENT_H4` and `-RX_DFE_COEFFICIENT_H5`.

-RX_POLARITY <value> Use this attribute to swap the P and N receiver pins, which provide flexible PCB routing by interchanging the devices physical pin to the logical signal. The default value is Normal.

Direction: Input

Table 6-11. RX_POLARITY Values

Name	Value
RX_POLARITY	Normal
	Inverted

Example

```
edit_io -port_name A -RES_PULL Down
-CLAMP_DIODE LVCMOS15 \
-OUT_DRIVE 12
```

6.2. edit_instance_delay (Ask a Question)

This command is used to modify the instance delay properties. This command can be used for floor plan PDC constraints and `edit_post_layout_design` tool.

```
edit_instance_delay \
        -inst_name <inst_name>\
        -properties  {<property_name>:<property_value>[ <property_name>:<property_value>]*}
```

Arguments

-inst_name <inst_name> Specifies the instance name of the I/O macro.

-properties {<property_name>:<property_value> The following are the supported user properties and values.

[<property_name>:<property_value>]*}}

Table 6-12. Supported User Properties and Values

Macro Name	Property	Values
LANECTRL	RX_DQS_DELAY	0–255
	TX_DQS_DELAY	0–255
ICB_CLKDIVDELAY	DELAY	(0–127, 128, 130, 132, ..., 254)
PLL_DELAY	DELAY	0–127

7. A – Packages/Memory Types (Ask a Question)

This appendix provides device, package, slot, and memory type information.

Device	Package	Slot	Memory Type
MPF050	FCVG484	NORTH_NE	DDR2
MPF050	FCVG484	NORTH_NE	DDR3
MPF050	FCVG484	NORTH_NE	DDR4
MPF050	FCVG484	NORTH_NE	QDR II+ x18
MPF050	FCVG484	NORTH_NE	QDR II+ x8
MPF050	FCVG484	NORTH_NE	QDR II+ x9
MPF050	FCVG484	NORTH_NW	DDR2
MPF050	FCVG484	NORTH_NW	DDR3
MPF050	FCVG484	NORTH_NW	DDR4
MPF050	FCVG484	NORTH_NW	QDR II+ x18
MPF050	FCVG484	NORTH_NW	QDR II+ x8
MPF050	FCVG484	NORTH_NW	QDR II+ x9
MPF050	FCVG484	SOUTH_SW	DDR2
MPF050	FCVG484	SOUTH_SW	DDR3
MPF050	FCVG484	SOUTH_SW	QDR II+ x8
MPF050	FCVG484	SOUTH_SW	QDR II+ x9
MPF050	FCSG325	NORTH_NE	DDR2
MPF050	FCSG325	NORTH_NE	DDR3
MPF050	FCSG325	NORTH_NE	DDR4
MPF050	FCSG325	NORTH_NE	QDR II+ x18
MPF050	FCSG325	NORTH_NE	QDR II+ x8
MPF050	FCSG325	NORTH_NE	QDR II+ x9
MPF050	FCSG325	NORTH_NW	DDR2
MPF050	FCSG325	NORTH_NW	DDR3
MPF050	FCSG325	NORTH_NW	DDR4
MPF050	FCSG325	NORTH_NW	QDR II+ x18
MPF050	FCSG325	NORTH_NW	QDR II+ x8
MPF050	FCSG325	NORTH_NW	QDR II+ x9
MPF050	FCSG325	SOUTH_SW	DDR2
MPF050	FCSG325	SOUTH_SW	DDR3
MPF100	FCG484	NORTH_NE	DDR3
MPF100	FCG484	NORTH_NE	DDR4
MPF100	FCG484	NORTH_NE	QDR II+ x18
MPF100	FCG484	NORTH_NE	QDR II+ x8
MPF100	FCG484	NORTH_NE	QDR II+ x9
MPF100	FCG484	NORTH_NW	DDR3
MPF100	FCG484	NORTH_NW	DDR4
MPF100	FCG484	NORTH_NW	QDR II+ x18
MPF100	FCG484	NORTH_NW	QDR II+ x8
MPF100	FCG484	NORTH_NW	QDR II+ x9
MPF100	FCG484	SOUTH_SW	DDR3
MPF100	FCG484	SOUTH_SW	QDR II+ x18

A – Packages/Memory Types (continued)

Device	Package	Slot	Memory Type
MPF100	FCG484	SOUTH_SW	QDR II+ x8
MPF100	FCG484	SOUTH_SW	QDR II+ x9
MPF100	FCSG325	NORTH_NE	DDR3
MPF100	FCSG325	NORTH_NE	DDR4
MPF100	FCSG325	NORTH_NE	QDR II+ x18
MPF100	FCSG325	NORTH_NE	QDR II+ x8
MPF100	FCSG325	NORTH_NE	QDR II+ x9
MPF100	FCSG325	NORTH_NW	DDR3
MPF100	FCSG325	NORTH_NW	DDR4
MPF100	FCSG325	NORTH_NW	QDR II+ x18
MPF100	FCSG325	NORTH_NW	QDR II+ x8
MPF100	FCSG325	NORTH_NW	QDR II+ x9
MPF100	FCSG325	SOUTH_SW	DDR3
MPF100	FCSG536	NORTH_NE	DDR3
MPF100	FCSG536	NORTH_NE	DDR4
MPF100	FCSG536	NORTH_NE	QDR II+ x18
MPF100	FCSG536	NORTH_NE	QDR II+ x8
MPF100	FCSG536	NORTH_NE	QDR II+ x9
MPF100	FCSG536	NORTH_NW	DDR3
MPF100	FCSG536	NORTH_NW	DDR4
MPF100	FCSG536	NORTH_NW	QDR II+ x18
MPF100	FCSG536	NORTH_NW	QDR II+ x8
MPF100	FCSG536	NORTH_NW	QDR II+ x9
MPF100	FCSG536	SOUTH_SW	DDR3
MPF100	FCSG536	SOUTH_SW	QDR II+ x18
MPF100	FCSG536	SOUTH_SW	QDR II+ x8
MPF100	FCSG536	SOUTH_SW	QDR II+ x9
MPF100	FCSG536	WEST_NW	DDR3
MPF100	FCSG536	WEST_NW	QDR II+ x8
MPF100	FCSG536	WEST_NW	QDR II+ x9
MPF100	FCVG484	NORTH_NE	DDR3
MPF100	FCVG484	NORTH_NE	DDR4
MPF100	FCVG484	NORTH_NE	QDR II+ x18
MPF100	FCVG484	NORTH_NE	QDR II+ x8
MPF100	FCVG484	NORTH_NE	QDR II+ x9
MPF100	FCVG484	NORTH_NW	DDR3
MPF100	FCVG484	NORTH_NW	DDR4
MPF100	FCVG484	NORTH_NW	QDR II+ x18
MPF100	FCVG484	NORTH_NW	QDR II+ x8
MPF100	FCVG484	NORTH_NW	QDR II+ x9
MPF100	FCVG484	SOUTH_SW	DDR3
MPF100	FCVG484	SOUTH_SW	QDR II+ x18
MPF100	FCVG484	SOUTH_SW	QDR II+ x8
MPF100	FCVG484	SOUTH_SW	QDR II+ x9
MPF100	FCVG484	WEST_NW	DDR3

A – Packages/Memory Types (continued)			
Device	**Package**	**Slot**	**Memory Type**
MPF100	FCVG484	WEST_NW	QDR II+ x8
MPF100	FCVG484	WEST_NW	QDR II+ x9
MPF200	FCG484	NORTH_NE	DDR3
MPF200	FCG484	NORTH_NE	DDR4
MPF200	FCG484	NORTH_NE	QDR II+ x18
MPF200	FCG484	NORTH_NE	QDR II+ x8
MPF200	FCG484	NORTH_NE	QDR II+ x9
MPF200	FCG484	NORTH_NW	DDR3
MPF200	FCG484	NORTH_NW	DDR4
MPF200	FCG484	NORTH_NW	QDR II+ x18
MPF200	FCG484	NORTH_NW	QDR II+ x8
MPF200	FCG484	NORTH_NW	QDR II+ x9
MPF200	FCG484	SOUTH_SW	DDR3
MPF200	FCG484	SOUTH_SW	QDR II+ x18
MPF200	FCG484	SOUTH_SW	QDR II+ x8
MPF200	FCG484	SOUTH_SW	QDR II+ x9
MPF200	FCG484	WEST_SW	DDR3
MPF200	FCG484	WEST_SW	QDR II+ x8
MPF200	FCG484	WEST_SW	QDR II+ x9
MPF200	FCG784	NORTH_NE	DDR3
MPF200	FCG784	NORTH_NE	DDR4
MPF200	FCG784	NORTH_NE	QDR II+ x18
MPF200	FCG784	NORTH_NE	QDR II+ x36
MPF200	FCG784	NORTH_NE	QDR II+ x8
MPF200	FCG784	NORTH_NE	QDR II+ x9
MPF200	FCG784	NORTH_NW	DDR3
MPF200	FCG784	NORTH_NW	DDR4
MPF200	FCG784	NORTH_NW	QDR II+ x18
MPF200	FCG784	NORTH_NW	QDR II+ x36
MPF200	FCG784	NORTH_NW	QDR II+ x8
MPF200	FCG784	NORTH_NW	QDR II+ x9
MPF200	FCG784	SOUTH_SW	DDR3
MPF200	FCG784	SOUTH_SW	QDR II+ x18
MPF200	FCG784	SOUTH_SW	QDR II+ x8
MPF200	FCG784	SOUTH_SW	QDR II+ x9
MPF200	FCG784	WEST_NW	DDR3
MPF200	FCG784	WEST_NW	QDR II+ x18
MPF200	FCG784	WEST_NW	QDR II+ x36
MPF200	FCG784	WEST_NW	QDR II+ x8
MPF200	FCG784	WEST_NW	QDR II+ x9
MPF200	FCG784	WEST_SW	DDR3
MPF200	FCG784	WEST_SW	QDR II+ x18
MPF200	FCG784	WEST_SW	QDR II+ x8
MPF200	FCG784	WEST_SW	QDR II+ x9
MPF200	FCSG325	NORTH_NE	DDR3

A – Packages/Memory Types (continued)

Device	Package	Slot	Memory Type
MPF200	FCSG325	NORTH_NE	DDR4
MPF200	FCSG325	NORTH_NE	QDR II+ x18
MPF200	FCSG325	NORTH_NE	QDR II+ x8
MPF200	FCSG325	NORTH_NE	QDR II+ x9
MPF200	FCSG325	NORTH_NW	DDR3
MPF200	FCSG325	NORTH_NW	DDR4
MPF200	FCSG325	NORTH_NW	QDR II+ x18
MPF200	FCSG325	NORTH_NW	QDR II+ x8
MPF200	FCSG325	NORTH_NW	QDR II+ x9
MPF200	FCSG325	SOUTH_SW	DDR3
MPF200	FCSG536	NORTH_NE	DDR3
MPF200	FCSG536	NORTH_NE	DDR4
MPF200	FCSG536	NORTH_NE	QDR II+ x18
MPF200	FCSG536	NORTH_NE	QDR II+ x8
MPF200	FCSG536	NORTH_NE	QDR II+ x9
MPF200	FCSG536	NORTH_NW	DDR3
MPF200	FCSG536	NORTH_NW	DDR4
MPF200	FCSG536	NORTH_NW	QDR II+ x18
MPF200	FCSG536	NORTH_NW	QDR II+ x8
MPF200	FCSG536	NORTH_NW	QDR II+ x9
MPF200	FCSG536	SOUTH_SW	DDR3
MPF200	FCSG536	SOUTH_SW	QDR II+ x18
MPF200	FCSG536	SOUTH_SW	QDR II+ x8
MPF200	FCSG536	SOUTH_SW	QDR II+ x9
MPF200	FCSG536	WEST_NW	DDR3
MPF200	FCSG536	WEST_NW	QDR II+ x18
MPF200	FCSG536	WEST_NW	QDR II+ x8
MPF200	FCSG536	WEST_NW	QDR II+ x9
MPF200	FCSG536	WEST_SW	DDR3
MPF200	FCSG536	WEST_SW	QDR II+ x8
MPF200	FCSG536	WEST_SW	QDR II+ x9
MPF200	FCVG484	NORTH_NE	DDR3
MPF200	FCVG484	NORTH_NE	DDR4
MPF200	FCVG484	NORTH_NE	QDR II+ x18
MPF200	FCVG484	NORTH_NE	QDR II+ x8
MPF200	FCVG484	NORTH_NE	QDR II+ x9
MPF200	FCVG484	NORTH_NW	DDR3
MPF200	FCVG484	NORTH_NW	DDR4
MPF200	FCVG484	NORTH_NW	QDR II+ x18
MPF200	FCVG484	NORTH_NW	QDR II+ x8
MPF200	FCVG484	NORTH_NW	QDR II+ x9
MPF200	FCVG484	SOUTH_SW	DDR3
MPF200	FCVG484	SOUTH_SW	QDR II+ x18
MPF200	FCVG484	SOUTH_SW	QDR II+ x8
MPF200	FCVG484	SOUTH_SW	QDR II+ x9

A – Packages/Memory Types (continued)

Device	Package	Slot	Memory Type
MPF200	FCVG484	WEST_NW	DDR3
MPF200	FCVG484	WEST_NW	QDR II+ x8
MPF200	FCVG484	WEST_NW	QDR II+ x9
MPF200	FCVG484	WEST_SW	DDR3
MPF200	FCVG484	WEST_SW	QDR II+ x8
MPF200	FCVG484	WEST_SW	QDR II+ x9
MPF300	FCG1152	NORTH_NE	DDR3
MPF300	FCG1152	NORTH_NE	DDR4
MPF300	FCG1152	NORTH_NE	QDR II+ x18
MPF300	FCG1152	NORTH_NE	QDR II+ x36
MPF300	FCG1152	NORTH_NE	QDR II+ x8
MPF300	FCG1152	NORTH_NE	QDR II+ x9
MPF300	FCG1152	NORTH_NW	DDR3
MPF300	FCG1152	NORTH_NW	DDR4
MPF300	FCG1152	NORTH_NW	QDR II+ x18
MPF300	FCG1152	NORTH_NW	QDR II+ x36
MPF300	FCG1152	NORTH_NW	QDR II+ x8
MPF300	FCG1152	NORTH_NW	QDR II+ x9
MPF300	FCG1152	SOUTH_SE	DDR3
MPF300	FCG1152	SOUTH_SE	DDR4
MPF300	FCG1152	SOUTH_SE	QDR II+ x8
MPF300	FCG1152	SOUTH_SE	QDR II+ x9
MPF300	FCG1152	SOUTH_SW	DDR3
MPF300	FCG1152	SOUTH_SW	QDR II+ x18
MPF300	FCG1152	SOUTH_SW	QDR II+ x8
MPF300	FCG1152	SOUTH_SW	QDR II+ x9
MPF300	FCG1152	WEST_NW	DDR3
MPF300	FCG1152	WEST_NW	QDR II+ x18
MPF300	FCG1152	WEST_NW	QDR II+ x36
MPF300	FCG1152	WEST_NW	QDR II+ x8
MPF300	FCG1152	WEST_NW	QDR II+ x9
MPF300	FCG1152	WEST_SW	DDR3
MPF300	FCG1152	WEST_SW	QDR II+ x18
MPF300	FCG1152	WEST_SW	QDR II+ x36
MPF300	FCG1152	WEST_SW	QDR II+ x8
MPF300	FCG1152	WEST_SW	QDR II+ x9
MPF300	FCG484	NORTH_NE	DDR3
MPF300	FCG484	NORTH_NE	DDR4
MPF300	FCG484	NORTH_NE	QDR II+ x18
MPF300	FCG484	NORTH_NE	QDR II+ x8
MPF300	FCG484	NORTH_NE	QDR II+ x9
MPF300	FCG484	NORTH_NW	DDR3
MPF300	FCG484	NORTH_NW	DDR4
MPF300	FCG484	NORTH_NW	QDR II+ x18
MPF300	FCG484	NORTH_NW	QDR II+ x8

A – Packages/Memory Types (continued)

Device	Package	Slot	Memory Type
MPF300	FCG484	NORTH_NW	QDR II+ x9
MPF300	FCG484	SOUTH_SW	DDR3
MPF300	FCG484	SOUTH_SW	QDR II+ x18
MPF300	FCG484	SOUTH_SW	QDR II+ x8
MPF300	FCG484	SOUTH_SW	QDR II+ x9
MPF300	FCG484	WEST_SW	DDR3
MPF300	FCG484	WEST_SW	QDR II+ x8
MPF300	FCG484	WEST_SW	QDR II+ x9
MPF300	FCG784	NORTH_NE	DDR3
MPF300	FCG784	NORTH_NE	DDR4
MPF300	FCG784	NORTH_NE	QDR II+ x18
MPF300	FCG784	NORTH_NE	QDR II+ x36
MPF300	FCG784	NORTH_NE	QDR II+ x8
MPF300	FCG784	NORTH_NE	QDR II+ x9
MPF300	FCG784	NORTH_NW	DDR3
MPF300	FCG784	NORTH_NW	DDR4
MPF300	FCG784	NORTH_NW	QDR II+ x18
MPF300	FCG784	NORTH_NW	QDR II+ x36
MPF300	FCG784	NORTH_NW	QDR II+ x8
MPF300	FCG784	NORTH_NW	QDR II+ x9
MPF300	FCG784	SOUTH_SW	DDR3
MPF300	FCG784	SOUTH_SW	QDR II+ x18
MPF300	FCG784	SOUTH_SW	QDR II+ x8
MPF300	FCG784	SOUTH_SW	QDR II+ x9
MPF300	FCG784	WEST_NW	DDR3
MPF300	FCG784	WEST_NW	QDR II+ x18
MPF300	FCG784	WEST_NW	QDR II+ x36
MPF300	FCG784	WEST_NW	QDR II+ x8
MPF300	FCG784	WEST_NW	QDR II+ x9
MPF300	FCG784	WEST_SW	DDR3
MPF300	FCG784	WEST_SW	QDR II+ x18
MPF300	FCG784	WEST_SW	QDR II+ x36
MPF300	FCG784	WEST_SW	QDR II+ x8
MPF300	FCG784	WEST_SW	QDR II+ x9
MPF300	FCSG536	NORTH_NE	DDR3
MPF300	FCSG536	NORTH_NE	DDR4
MPF300	FCSG536	NORTH_NE	QDR II+ x18
MPF300	FCSG536	NORTH_NE	QDR II+ x8
MPF300	FCSG536	NORTH_NE	QDR II+ x9
MPF300	FCSG536	NORTH_NW	DDR3
MPF300	FCSG536	NORTH_NW	DDR4
MPF300	FCSG536	NORTH_NW	QDR II+ x18
MPF300	FCSG536	NORTH_NW	QDR II+ x8
MPF300	FCSG536	NORTH_NW	QDR II+ x9
MPF300	FCSG536	SOUTH_SW	DDR3

A – Packages/Memory Types (continued)			
Device	**Package**	**Slot**	**Memory Type**
MPF300	FCSG536	SOUTH_SW	QDR II+ x18
MPF300	FCSG536	SOUTH_SW	QDR II+ x8
MPF300	FCSG536	SOUTH_SW	QDR II+ x9
MPF300	FCSG536	WEST_NW	DDR3
MPF300	FCSG536	WEST_NW	QDR II+ x18
MPF300	FCSG536	WEST_NW	QDR II+ x8
MPF300	FCSG536	WEST_NW	QDR II+ x9
MPF300	FCSG536	WEST_SW	DDR3
MPF300	FCSG536	WEST_SW	QDR II+ x8
MPF300	FCSG536	WEST_SW	QDR II+ x9
MPF300	FCVG484	NORTH_NE	DDR3
MPF300	FCVG484	NORTH_NE	DDR4
MPF300	FCVG484	NORTH_NE	QDR II+ x18
MPF300	FCVG484	NORTH_NE	QDR II+ x8
MPF300	FCVG484	NORTH_NE	QDR II+ x9
MPF300	FCVG484	NORTH_NW	DDR3
MPF300	FCVG484	NORTH_NW	DDR4
MPF300	FCVG484	NORTH_NW	QDR II+ x18
MPF300	FCVG484	NORTH_NW	QDR II+ x8
MPF300	FCVG484	NORTH_NW	QDR II+ x9
MPF300	FCVG484	SOUTH_SW	DDR3
MPF300	FCVG484	SOUTH_SW	QDR II+ x18
MPF300	FCVG484	SOUTH_SW	QDR II+ x8
MPF300	FCVG484	SOUTH_SW	QDR II+ x9
MPF300	FCVG484	WEST_NW	DDR3
MPF300	FCVG484	WEST_NW	QDR II+ x8
MPF300	FCVG484	WEST_NW	QDR II+ x9
MPF300	FCVG484	WEST_SW	DDR3
MPF300	FCVG484	WEST_SW	QDR II+ x8
MPF300	FCVG484	WEST_SW	QDR II+ x9
MPF500	FCG1152	NORTH_NE	DDR3
MPF500	FCG1152	NORTH_NE	DDR4
MPF500	FCG1152	NORTH_NE	QDR II+ x18
MPF500	FCG1152	NORTH_NE	QDR II+ x36
MPF500	FCG1152	NORTH_NE	QDR II+ x8
MPF500	FCG1152	NORTH_NE	QDR II+ x9
MPF500	FCG1152	NORTH_NW	DDR3
MPF500	FCG1152	NORTH_NW	DDR4
MPF500	FCG1152	NORTH_NW	QDR II+ x18
MPF500	FCG1152	NORTH_NW	QDR II+ x36
MPF500	FCG1152	NORTH_NW	QDR II+ x8
MPF500	FCG1152	NORTH_NW	QDR II+ x9
MPF500	FCG1152	SOUTH_SE	DDR3
MPF500	FCG1152	SOUTH_SE	DDR4
MPF500	FCG1152	SOUTH_SE	QDR II+ x18

A – Packages/Memory Types (continued)			
Device	**Package**	**Slot**	**Memory Type**
MPF500	FCG1152	SOUTH_SE	QDR II+ x8
MPF500	FCG1152	SOUTH_SE	QDR II+ x9
MPF500	FCG1152	SOUTH_SW	DDR3
MPF500	FCG1152	SOUTH_SW	QDR II+ x18
MPF500	FCG1152	SOUTH_SW	QDR II+ x8
MPF500	FCG1152	SOUTH_SW	QDR II+ x9
MPF500	FCG1152	WEST_NW	DDR3
MPF500	FCG1152	WEST_NW	QDR II+ x18
MPF500	FCG1152	WEST_NW	QDR II+ x36
MPF500	FCG1152	WEST_NW	QDR II+ x8
MPF500	FCG1152	WEST_NW	QDR II+ x9
MPF500	FCG1152	WEST_SW	DDR3
MPF500	FCG1152	WEST_SW	QDR II+ x18
MPF500	FCG1152	WEST_SW	QDR II+ x36
MPF500	FCG1152	WEST_SW	QDR II+ x8
MPF500	FCG1152	WEST_SW	QDR II+ x9
MPF500	FCG784	NORTH_NE	DDR3
MPF500	FCG784	NORTH_NE	DDR4
MPF500	FCG784	NORTH_NE	QDR II+ x18
MPF500	FCG784	NORTH_NE	QDR II+ x36
MPF500	FCG784	NORTH_NE	QDR II+ x8
MPF500	FCG784	NORTH_NE	QDR II+ x9
MPF500	FCG784	NORTH_NW	DDR3
MPF500	FCG784	NORTH_NW	DDR4
MPF500	FCG784	NORTH_NW	QDR II+ x18
MPF500	FCG784	NORTH_NW	QDR II+ x36
MPF500	FCG784	NORTH_NW	QDR II+ x8
MPF500	FCG784	NORTH_NW	QDR II+ x9
MPF500	FCG784	SOUTH_SW	DDR3
MPF500	FCG784	SOUTH_SW	QDR II+ x18
MPF500	FCG784	SOUTH_SW	QDR II+ x8
MPF500	FCG784	SOUTH_SW	QDR II+ x9
MPF500	FCG784	WEST_NW	DDR3
MPF500	FCG784	WEST_NW	QDR II+ x18
MPF500	FCG784	WEST_NW	QDR II+ x36
MPF500	FCG784	WEST_NW	QDR II+ x8
MPF500	FCG784	WEST_NW	QDR II+ x9
MPF500	FCG784	WEST_SW	DDR3
MPF500	FCG784	WEST_SW	QDR II+ x18
MPF500	FCG784	WEST_SW	QDR II+ x36
MPF500	FCG784	WEST_SW	QDR II+ x8
MPF500	FCG784	WEST_SW	QDR II+ x9

8. Revision History (Ask a Question)

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
Q	12/2025	The following changes are made in this revision. • Updated the -IN_DELAY and -OUT_DELAY arguments of the set_io command.
P	05/2025	The following changes are made in this revision. • Updated the set_preserve command. • Updated the edit_io command.
N	11/2024	The following changes are made in this revision. • Updated section set_ioff.
M	09/2024	The following changes are made in this revision. • Updated section set_io.
L	08/2024	The following changes are made in this revision. • Updated section set_ioff.
K	02/2024	This document is released with Libero SoC Design Suite v2024.1 without changes from v2023.2.
J	08/2023	This document is released with Libero SoC Design Suite v2023.2 without changes from v2023.1.
H	04/2023	The following changes are made in this revision. • Updated section Netlist Attributes NDC Commands. • Updated section edit_io. • Updated section edit_instance_delay.
G	12/2022	This document is released with Libero SoC Design Suite v2022.3 without changes from v2022.2.
F	08/2022	The following changes are made in this revision. • Updated the set_iobank command.
E	04/2022	This document is released with Libero SoC Design Suite v2022.1 without changes from v2021.3.
D	12/2021	Editorial updates only. No technical content updates.
C	08/2021	• set_io: Updated `res_pull` and `clamp_diode` arguments and added a new GPIO LVDS18G. • Post Layout Edit PDC Commands: Added new PDC Commands `edit_io` and `edit_instance_delay` • Packages/Memory Types: Added a new device MPF050.
B	04/2021	Editorial updates only. No technical content updates.
A	11/2020	Document converted to Microchip template. Initial Revision.

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

Trademarks

The “Microchip” name and logo, the “M” logo, and other names, logos, and brands are registered and unregistered trademarks of Microchip Technology Incorporated or its affiliates and/or subsidiaries in the United States and/or other countries (“Microchip Trademarks”). Information regarding Microchip Trademarks can be found at https://www.microchip.com/en-us/about/legal-information/microchip-trademarks.

ISBN: 979-8-3371-2424-7

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.

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 products are 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.