UG0715
PDC Commands User Guide
PolarFire FPGA
Libero SoC v12.0

Microsemi
a Microchip company

Microsemi Headquarters
One Enterprise, Aliso Viejo,
CA 92656 USA
Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996
Email: sales.support@microsemi.com
www.microsemi.com

About Microsemi

Microsemi, a wholly owned subsidiary of Microchip Technology Inc. (Nasdaq: MCHP), offers a comprehensive portfolio of semiconductor and system solutions for aerospace & defense, communications, data center and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; enterprise storage and communication solutions, security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Learn more at www.microsemi.com.

50200715-7.0/12.18

Table of Contents

Introduction

In the FPGA design world, constraint files are as important as design source files. Physical design constraints (PDC) are used to constrain the I/Os 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 I/O attributes changes and the Chip Planner is the interactive tool for making floorplanning changes. When changes are made in the I/O Attribute Editor or the Chip Planner, the PDC file(s) are updated to reflect the changes. These PDC commands can be used as part of a script file to constrain the Place

and Route step of your design.

Supported Families

This User Guide covers the PDC commands applicable to PolarFire devices.

PDC Syntax Conventions

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

Syntax Notation	Description
`command -argument`	Commands and arguments appear in Courier New typeface.
`variable`	Variables appear in blue, italic Courier New typeface. You must substitute an appropriate value for the variable.
[`-argument` *`value`*] [*`variable`*]+	Optional arguments begin and end with a square bracket with one exception: if the square bracket is followed by a plus sign (+), users must specify at least one argument. The plus sign (+) indicates that items within the square brackets can be repeated. Do not enter the plus sign character.

Note: PDC commands are case sensitive. However, their arguments are not.

Examples

Syntax for the `assign_local_clock` command followed by a sample command:

```
assign_local_clock -type value -net netname [LocalClock_region ]+
assign_local_clock -type hclk -net  reset_n tile1a tile2a
```

Syntax for the `set_io` command followed by a sample command:

```
set_io portname [-iostd value][-register value][-out_drive value][-slew value][-res_pull
value][-out_load value][-pinname value][-fixed value][-in_delay value]
          set_io ADDOUT2 \
          -iostd PCI \
          -register yes \
          -out_drive 16 \ -slew high \
          -out_load 10 \
          -pinname T21 \
          -fixed yes
```

Wildcard Characters

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

Wildcard	What It Does
\	Interprets the next character literally
?	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 wildcards in a PDC command.

Special Characters ([ ], { }, and \)

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 (\). If you do not, you will get an error message.

For example:

```
set_iobank {mem_data_in[57]} -fixed no 7 2
or set_iobank mem_data_in\[57\] -fixed no 7 2
```

Entering Arguments on Separate Lines

To enter an argument on a separate line, you 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 \
-register Yes \
-out_drive 16 \
-slew High \
-out_load 10 \
-pinname T21 \
-fixed yes
```

PDC Naming Conventions

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

Rules for Displaying Original Names

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 slash (\) that is not a hierarchy separator. For example, the instance named A/\B is displayed as A\/\\B.

Which Name Do I Use in PDC Commands?

Using PDC Commands

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 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:

Type of name and its location	Name in the imported netlist	Name to use in PDC file
Port name in 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.

Case Sensitivity When Importing PDC Files

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

File Type	Case Sensitivity
Verilog	Names in the netlist are case sensitive.
EDIF (SmartFusion2, IGLOO2, and RTG4)	Names in the netlist are always case sensitive because we use the Rename clause, which is 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" will fail if spelled as "a10" in the `set_location` command:

`set_location A10` (This command will succeed.)

`set_location a10` (This command will fail.)

1 – I/O PDC commands

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

For detailed information about I/Os and I/O standards, refer to UG0686: PolarFire FPGA User I/O User Guide.

set_iobank

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 and you do not need to set any pin on these banks.

There are two types of I/O banks: General-Purpose IO (GPIO) and High-Speed IO (HSIO).

Each bank type supports a different set of I/O standards as listed in the Table below.

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, LVDS25, LVDS33, RSDS25, RSDS33, MINILVDS25, MINILVDS33, SUBLVDS25, SUBLVDS33, PPDS25, PPDS33, SLVSE15, MLVDSE25, BUSLVDSE25, LCMDS33, LCMDS25, SHIELD33, SHIELD25, SHIELD18, SHIELD15, SHIELD12

```
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 <bank_name>

Specifies the name of the bank. I/O banks are numbered 0 through N (bank0, bank1,...bankN). The number of I/O banks varies with the device. Refer to the datasheet for your device to determine how many banks it has.

-vcci <vcci_voltage>

Sets the input/output supply voltage. You can enter one of the following values:

Vcci Voltage	Compatible Standards
3.3 V	LVTTL, LVCMOS33, PCI, LVDS33, LVPECL33, LVPECLE33, SLVS33, HCSL33, RSDS33, MINILVDS33, SUBLVDS33
2.5 V	LVCMOS25, SSTL25I,SSTL25II, PPDS25, SLVS25, HCSL25, MLVDSE25, MINILVDS25, RSDS25, SUBLVDS25, LVDS25, MLVDSE25, BUSLVDSE25

1.8 V	LVCMOS18, SSTL18I, SSTL18II, HSUL18I, HSUL18II, SLVS18, HCSL18, LVDS18, RSDS18, MINILVDS18, SUBLVDS18, PPDS18
1.5 V	LVCMOS15, SSTL15I, SSTL15II, HSTL15I, HSTL15II, SLVSE15
1.35 V	HSTL135I, HSTL135II, SSTL135I, SSTL135II
1.2 V	LVCMOS12, HSUL12I, HSTL12I, POD12I, MIPI12

-vref <vref_voltage>

Sets the input reference voltage. You can enter one of the following values:

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

-fixed <value>

Specifies if the I/O technologies (vcci and vccr voltage) assigned to the bank are locked. You can enter one of the following values:

Value	Description
true	The technologies are locked.
false	The technologies are not locked.

-update_iostd <value>

Specifies if the I/O technologies (vcci and vccr voltage) assigned to the bank are locked. You can enter one of the following values:

Value	Description
true	If there are I/O's placed on the bank, we keep the placement and change the host to one which is compatible with this bank setting. Check the I/O Attributes to see the one used by the tool.
false	If there are I/O's placed and locked on the bank, the command will fail. If they are placed I/Os they will be unplaced.

-auto_calib <value>

Specifies whether the I/O bank is auto-calibrated at power up. Values are true/false, and the default value is true.

Note: 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 can be 1–50, and 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 3.3 V to the input/output supply voltage (vcci) for I/O bank 0.

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

reserve

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"
```

set_io

PDC command; You can use the set_io command to assign an I/O technology, place, or lock the I/O at a given pin location. There are two I/O types available for PolarFire: GPIO and HSIO. Each I/O type supports different I/O 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, 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.

```
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>] \
[-VICM_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 portname 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 will invalidate its location and automatically unassign the I/O.

The following table shows a list of supported 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. The table below lists the different I/O standards and the type of I/O they support.

IO_STD Value	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
BUSLVDS	NO	YES
BUSLVDSE25	NO	YES
MLVDS	NO	YES
MIPI25	NO	YES
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 if the location of this port is fixed (i.e., locked). Locked ports are not moved during layout. The default value is true. You can enter one of the following values:

Value	Description
true	The location of this port is locked.
false	The location of this port is unlocked.

Examples

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

I/O Directions Not Supported

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

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, LVDS25, LVDS33, RSDS25, RSDS33, MINILVDS25, MINILVDS33, SUBLVDS25, SUBLVDS33, PPDS25, PPDS33, LCMDS25, LCMDS33, LCMDS18
Inout	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

-OUT_LOAD <value>

Sets the output load (in pF) of output signals.

The default is 5.

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 shows the acceptable values for the -RES_PULL attribute for the input buffer:

I/O Standard	Value	Description
LVCMOS25, LVCMOS33, LVTTL, PCI, LVCMOS18, LVCMOS15, LVCMOS12	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

For I/O standards listed in the table above, the default is Up.

For all other I/O standards, the value is None.

The default is None.

The following table shows the acceptable values for the -RES_PULL attribute for the output buffer:

I/O Standard	Value	Description
LVCMOS25, LVCMOS33, LVTTL, PCI, LVCMOS18, LVCMOS15, LVCMOS12	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

For all I/O standards, the default value for output buffer is None.

Direction: Inout

-LOCK_DOWN <value>

Security feature that locks down the I/Os if tampering is detected.

Values are ON, 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 over-voltage conditions.

Values are OFF, ON.

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

I/O Standard	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	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

-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, ON. The default is OFF.

I/O Standard	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, ON. The default is OFF.

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

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

Direction: Output

-VICM_RANGE <value>

Sets the VCM input range.

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

I/O Standard	Values
GPIO	
HSTL15I, HSTL15II, HSUL18I, HSUL18II, SSTL15I, SSTL15II, SSTL18I, SSTL18II, SSTL25I, SSTL25II	MID
HCSL33, HCSL25, 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, 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.	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 *Note:* While assigning VCM input range for true differential I/Os in the same bank, a mix of MID, 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.	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.

Values are OFF, ON.

The following table lists acceptable values.

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

Direction: Input

-ODT_VALUE

Sets the ODT value (in Ohms) for On Die Termination.

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

I/O Standard	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.
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. Also, each I/O standard has a different range of legal output drive strength values. The values you can choose from depend on which I/O standard you have specified for this command. The table below lists acceptable values.

I/O Standard	Values
GPIO	
LVCMOS12	2, 4, 6, 8. The default is 8.
LVCMOS15	2, 4, 6, 8, 10. The default is 8.
LVCMOS18	2, 4, 6, 8, 10, 12. The default is 8.
LVCMOS25	2, 4, 6, 8, 12, 16. The default is 8.
LVCMOS33, LVTTL	2, 4, 8, 12, 16, 20. The default is 8.
LVDS25, LVDS33, MINILVDS25, MINILVDS33, LCMDS33, LCMDS25	3, 3.5, 4, 6. The default is 6.
PPDS25, PPDS33, RSDS25, RSDS33	1.5, 2, 3. The default is 3.
SUBLVDS25, SUBLVDS33	1, 1.5, 2. The default is 2.

BUSLVDSE25, MLVDSE25, LVPECLE33	16
MIPIE25, SLVSE15	8
PCI	20
HSIO	
LVCMOS12, LVCMOS15	2, 4, 6, 8, 10. The default is 8.
LVCMOS18	2, 4, 6, 8, 10, 12. The default is 8
SLVSE15	8

Direction: Output

-IMPEDANCE

Sets the Impedance value (in Ohms).

Values vary depending on the I/O standard. LVCMOS25he table below lists acceptable values.

Direction: Output

-SOURCE_TERM

Near End termination for a differential output I/O. The table below lists acceptable values.

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

Direction: Output

-IN_DELAY

Sets the 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

Note: This attribute will not appear in the I/O attributes and cannot be used in the PDC for some I/Os with dynamic delays, such as DDR I/Os.

-OUT_DELAY

Sets the Output Delay.

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

Direction: Output

Note: This attribute will not appear in the I/O attributes and cannot be used in the PDC for some I/Os with dynamic delays, such as DDR I/Os.

set_location

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>
```

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 shows the acceptable values for this argument.

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 needs to 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. Check the following table for the correct location to place the DDR3 memory. The numbers in the table refer to the maximum DDR3 width.

	Location (Edge_Anchor} Edge={NORTH/SOUTH/WEST}, Anchor={NE/NW/SE/SW}					
Die/Package	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

See "Placement Rules for PLLs and DLLs " for more information.

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

See "Placement Rules for PLLs and DLLs " for more placment information.

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

See "Placement Rules for Transceivers " for more placment information.

Placement Rules for PLLs and DLLs

This section outlines the placement rules for PLL and DLL instances. You must place PLL and DLL instances using the set_location command.

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

PRPF_006: PLL/DLL <inst name> must be placed before running Place & Route.

All PLL and DLL instances must be placed before running Place and Route.

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, NW) 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 sum total of PLL/DLL reference clocks and fabric clocks that the fabric drives must be six or less.

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:

PRPF_001: Port <port name> for Interface <inst name> must be placed before running Place & 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)> be 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 will automatically be placed.

IOD instances with TRAINING/OVERLAY should not be placed by users. 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:

PRPF_007: TxPLL <inst name> must be placed before running Place & 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 & Route.*

The transceiver interface has dedicated ports. These must be placed using the set_io command.

Note: For information about rules for transceivers, refer to UG0677: User Guide PolarFire FPGA Transceiver.

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.

Note: For information about rules for transceivers, refer to UG0677: User Guide PolarFire FPGA Transceiver.

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.

Note: For information about rules, refer to UG0677: User Guide PolarFire FPGA Transceiver.

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.

Note: For information about rules for transceivers, refer to UG0677: User Guide PolarFire FPGA Transceiver.

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.

Note: For information about rules for transceivers, refer to UG0677: User Guide PolarFire FPGA Transceiver.

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.

Note: For information about rules for transceivers, refer to UG0677: User Guide PolarFire FPGA Transceiver.

2 – Netlist Attributes PDC Commands

Netlist Attributes PDC Commands are used to set netlist-specific constraints. 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.

set_preserve

This command sets a preserve property on instances before compile, so compile will preserve these instances and not combine them.

```
set_preserve -inst_name <instance_name>
```

Arguments

-inst_name

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

Exceptions

You must put this command in a PDC constraint file and associate it with Place and Route.

Example

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

3 – Floorplanning PDC Commands

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

assign_region

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. You can use the following wild card characters in macro names:

Wild Card	What it does
\	Interprets the next character as a non-special character
?	Matches any single character
*	Matches any string

Note:

- 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/*
```

See Also

""

assign_net_macros

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 are assigning 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. You can use the following wild card characters in net names:

Wild Card	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. You can enter one of the following values:

Value	Descriptions
true	Include the driver in the list of macros assigned to the region (default).
false	Do not assign the driver to the region.

Note:

- 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.

Examples

```
assign_net_macros -region_name UserRegion1 -net_name Y -include_driver false
```

define_region

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> [-color <integer>] [-route
<true|false>]
```

Note: The -color and -route parameters are optional.

Arguments

-region_name <region_name>

Specifies the region name. The name must be unique. Do not use reserved names such as "bank0" 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:

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.

-color <value>

Specifies the color of the region. The following table shows the recommended values for this argument:

Color	Decimal Value
	16776960
	65280
	16711680
	16760960
	255
	16711935
	65535
	33023
	8421631
	9568200
	8323199
	12632256

-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. You can enter one of the following 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 clocks and globals 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 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 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 not internal to the region can be assigned routing resources within the region.

Description

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

Use inclusive or exclusive region constraints if you intend to assign 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 utilization) in red.

Examples

The following example defines an empty rectangular region called UserRegion1 with lower-left co-ordinates (100,46) and upper-right co-ordinates (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 co-ordinates (12,39) and upper-right co-ordinates (23,41) and another rectangle with lower-left co-ordinates (12,33) and upper-right co-ordinates (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
```

See Also

"assign_region"

move_region

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

```
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, y2, where x1, y1 represents the lower-left corner of the rectangle and x2 y2 represents 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 co-ordinates (0,40) and upper-right co-ordinates (3,42):

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

See Also

"define_region"

A – Packages/Memory Types

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

DEVICE	PACKAGE	SLOT	MEMORY TYPE
PA5M100	FCG484	NORTH_NE	DDR3
PA5M100	FCG484	NORTH_NE	DDR4
PA5M100	FCG484	NORTH_NE	QDR II+ x18
PA5M100	FCG484	NORTH_NE	QDR II+ x8
PA5M100	FCG484	NORTH_NE	QDR II+ x9
PA5M100	FCG484	NORTH_NW	DDR3
PA5M100	FCG484	NORTH_NW	DDR4
PA5M100	FCG484	NORTH_NW	QDR II+ x18
PA5M100	FCG484	NORTH_NW	QDR II+ x8
PA5M100	FCG484	NORTH_NW	QDR II+ x9
PA5M100	FCG484	SOUTH_SW	DDR3
PA5M100	FCG484	SOUTH_SW	QDR II+ x18
PA5M100	FCG484	SOUTH_SW	QDR II+ x8
PA5M100	FCG484	SOUTH_SW	QDR II+ x9
PA5M100	FCSG325	NORTH_NE	DDR3
PA5M100	FCSG325	NORTH_NE	DDR4
PA5M100	FCSG325	NORTH_NE	QDR II+ x18
PA5M100	FCSG325	NORTH_NE	QDR II+ x8
PA5M100	FCSG325	NORTH_NE	QDR II+ x9
PA5M100	FCSG325	NORTH_NW	DDR3
PA5M100	FCSG325	NORTH_NW	DDR4
PA5M100	FCSG325	NORTH_NW	QDR II+ x18
PA5M100	FCSG325	NORTH_NW	QDR II+ x8
PA5M100	FCSG325	NORTH_NW	QDR II+ x9
PA5M100	FCSG325	SOUTH_SW	DDR3
PA5M100	FCSG536	NORTH_NE	DDR3
PA5M100	FCSG536	NORTH_NE	DDR4
PA5M100	FCSG536	NORTH_NE	QDR II+ x18
PA5M100	FCSG536	NORTH_NE	QDR II+ x8
PA5M100	FCSG536	NORTH_NE	QDR II+ x9
PA5M100	FCSG536	NORTH_NW	DDR3
PA5M100	FCSG536	NORTH_NW	DDR4
PA5M100	FCSG536	NORTH_NW	QDR II+ x18
PA5M100	FCSG536	NORTH_NW	QDR II+ x8
PA5M100	FCSG536	NORTH_NW	QDR II+ x9
PA5M100	FCSG536	SOUTH_SW	DDR3
PA5M100	FCSG536	SOUTH_SW	QDR II+ x18
PA5M100	FCSG536	SOUTH_SW	QDR II+ x8

PA5M100	FCSG536	SOUTH_SW	QDR II+ x9
PA5M100	FCSG536	WEST_NW	DDR3
PA5M100	FCSG536	WEST_NW	QDR II+ x8
PA5M100	FCSG536	WEST_NW	QDR II+ x9
PA5M100	FCVG484	NORTH_NE	DDR3
PA5M100	FCVG484	NORTH_NE	DDR4
PA5M100	FCVG484	NORTH_NE	QDR II+ x18
PA5M100	FCVG484	NORTH_NE	QDR II+ x8
PA5M100	FCVG484	NORTH_NE	QDR II+ x9
PA5M100	FCVG484	NORTH_NW	DDR3
PA5M100	FCVG484	NORTH_NW	DDR4
PA5M100	FCVG484	NORTH_NW	QDR II+ x18
PA5M100	FCVG484	NORTH_NW	QDR II+ x8
PA5M100	FCVG484	NORTH_NW	QDR II+ x9
PA5M100	FCVG484	SOUTH_SW	DDR3
PA5M100	FCVG484	SOUTH_SW	QDR II+ x18
PA5M100	FCVG484	SOUTH_SW	QDR II+ x8
PA5M100	FCVG484	SOUTH_SW	QDR II+ x9
PA5M100	FCVG484	WEST_NW	DDR3
PA5M100	FCVG484	WEST_NW	QDR II+ x8
PA5M100	FCVG484	WEST_NW	QDR II+ x9
PA5M100	FULLPKG	NORTH_NE	DDR3
PA5M100	FULLPKG	NORTH_NE	DDR4
PA5M100	FULLPKG	NORTH_NE	QDR II+ x18
PA5M100	FULLPKG	NORTH_NE	QDR II+ x8
PA5M100	FULLPKG	NORTH_NE	QDR II+ x9
PA5M100	FULLPKG	NORTH_NW	DDR3
PA5M100	FULLPKG	NORTH_NW	DDR4
PA5M100	FULLPKG	NORTH_NW	QDR II+ x18
PA5M100	FULLPKG	NORTH_NW	QDR II+ x8
PA5M100	FULLPKG	NORTH_NW	QDR II+ x9
PA5M100	FULLPKG	SOUTH_SW	DDR3
PA5M100	FULLPKG	SOUTH_SW	QDR II+ x18
PA5M100	FULLPKG	SOUTH_SW	QDR II+ x8
PA5M100	FULLPKG	SOUTH_SW	QDR II+ x9
PA5M100	FULLPKG	WEST_NW	DDR3
PA5M100	FULLPKG	WEST_NW	QDR II+ x8
PA5M100	FULLPKG	WEST_NW	QDR II+ x9
PA5M200	FCG484	NORTH_NE	DDR3
PA5M200	FCG484	NORTH_NE	DDR4
PA5M200	FCG484	NORTH_NE	QDR II+ x18
PA5M200	FCG484	NORTH_NE	QDR II+ x8
PA5M200	FCG484	NORTH_NE	QDR II+ x9
PA5M200	FCG484	NORTH_NW	DDR3
PA5M200	FCG484	NORTH_NW	DDR4

PA5M200	FCG484	NORTH_NW	QDR II+ x18
PA5M200	FCG484	NORTH_NW	QDR II+ x8
PA5M200	FCG484	NORTH_NW	QDR II+ x9
PA5M200	FCG484	SOUTH_SW	DDR3
PA5M200	FCG484	SOUTH_SW	QDR II+ x18
PA5M200	FCG484	SOUTH_SW	QDR II+ x8
PA5M200	FCG484	SOUTH_SW	QDR II+ x9
PA5M200	FCG484	WEST_SW	DDR3
PA5M200	FCG484	WEST_SW	QDR II+ x8
PA5M200	FCG484	WEST_SW	QDR II+ x9
PA5M200	FCG784	NORTH_NE	DDR3
PA5M200	FCG784	NORTH_NE	DDR4
PA5M200	FCG784	NORTH_NE	QDR II+ x18
PA5M200	FCG784	NORTH_NE	QDR II+ x36
PA5M200	FCG784	NORTH_NE	QDR II+ x8
PA5M200	FCG784	NORTH_NE	QDR II+ x9
PA5M200	FCG784	NORTH_NW	DDR3
PA5M200	FCG784	NORTH_NW	DDR4
PA5M200	FCG784	NORTH_NW	QDR II+ x18
PA5M200	FCG784	NORTH_NW	QDR II+ x36
PA5M200	FCG784	NORTH_NW	QDR II+ x8
PA5M200	FCG784	NORTH_NW	QDR II+ x9
PA5M200	FCG784	SOUTH_SW	DDR3
PA5M200	FCG784	SOUTH_SW	QDR II+ x18
PA5M200	FCG784	SOUTH_SW	QDR II+ x8
PA5M200	FCG784	SOUTH_SW	QDR II+ x9
PA5M200	FCG784	WEST_NW	DDR3
PA5M200	FCG784	WEST_NW	QDR II+ x18
PA5M200	FCG784	WEST_NW	QDR II+ x36
PA5M200	FCG784	WEST_NW	QDR II+ x8
PA5M200	FCG784	WEST_NW	QDR II+ x9
PA5M200	FCG784	WEST_SW	DDR3
PA5M200	FCG784	WEST_SW	QDR II+ x18
PA5M200	FCG784	WEST_SW	QDR II+ x8
PA5M200	FCG784	WEST_SW	QDR II+ x9
PA5M200	FCSG325	NORTH_NE	DDR3
PA5M200	FCSG325	NORTH_NE	DDR4
PA5M200	FCSG325	NORTH_NE	QDR II+ x18
PA5M200	FCSG325	NORTH_NE	QDR II+ x8
PA5M200	FCSG325	NORTH_NE	QDR II+ x9
PA5M200	FCSG325	NORTH_NW	DDR3
PA5M200	FCSG325	NORTH_NW	DDR4
PA5M200	FCSG325	NORTH_NW	QDR II+ x18
PA5M200	FCSG325	NORTH_NW	QDR II+ x8
PA5M200	FCSG325	NORTH_NW	QDR II+ x9

PA5M200	FCSG325	SOUTH_SW	DDR3
PA5M200	FCSG536	NORTH_NE	DDR3
PA5M200	FCSG536	NORTH_NE	DDR4
PA5M200	FCSG536	NORTH_NE	QDR II+ x18
PA5M200	FCSG536	NORTH_NE	QDR II+ x8
PA5M200	FCSG536	NORTH_NE	QDR II+ x9
PA5M200	FCSG536	NORTH_NW	DDR3
PA5M200	FCSG536	NORTH_NW	DDR4
PA5M200	FCSG536	NORTH_NW	QDR II+ x18
PA5M200	FCSG536	NORTH_NW	QDR II+ x8
PA5M200	FCSG536	NORTH_NW	QDR II+ x9
PA5M200	FCSG536	SOUTH_SW	DDR3
PA5M200	FCSG536	SOUTH_SW	QDR II+ x18
PA5M200	FCSG536	SOUTH_SW	QDR II+ x8
PA5M200	FCSG536	SOUTH_SW	QDR II+ x9
PA5M200	FCSG536	WEST_NW	DDR3
PA5M200	FCSG536	WEST_NW	QDR II+ x18
PA5M200	FCSG536	WEST_NW	QDR II+ x8
PA5M200	FCSG536	WEST_NW	QDR II+ x9
PA5M200	FCSG536	WEST_SW	DDR3
PA5M200	FCSG536	WEST_SW	QDR II+ x8
PA5M200	FCSG536	WEST_SW	QDR II+ x9
PA5M200	FCVG484	NORTH_NE	DDR3
PA5M200	FCVG484	NORTH_NE	DDR4
PA5M200	FCVG484	NORTH_NE	QDR II+ x18
PA5M200	FCVG484	NORTH_NE	QDR II+ x8
PA5M200	FCVG484	NORTH_NE	QDR II+ x9
PA5M200	FCVG484	NORTH_NW	DDR3
PA5M200	FCVG484	NORTH_NW	DDR4
PA5M200	FCVG484	NORTH_NW	QDR II+ x18
PA5M200	FCVG484	NORTH_NW	QDR II+ x8
PA5M200	FCVG484	NORTH_NW	QDR II+ x9
PA5M200	FCVG484	SOUTH_SW	DDR3
PA5M200	FCVG484	SOUTH_SW	QDR II+ x18
PA5M200	FCVG484	SOUTH_SW	QDR II+ x8
PA5M200	FCVG484	SOUTH_SW	QDR II+ x9
PA5M200	FCVG484	WEST_NW	DDR3
PA5M200	FCVG484	WEST_NW	QDR II+ x8
PA5M200	FCVG484	WEST_NW	QDR II+ x9
PA5M200	FCVG484	WEST_SW	DDR3
PA5M200	FCVG484	WEST_SW	QDR II+ x8
PA5M200	FCVG484	WEST_SW	QDR II+ x9
PA5M200	FULLPKG	NORTH_NE	DDR3
PA5M200	FULLPKG	NORTH_NE	DDR4
PA5M200	FULLPKG	NORTH_NE	QDR II+ x18

PA5M200	FULLPKG	NORTH_NE	QDR II+ x36
PA5M200	FULLPKG	NORTH_NE	QDR II+ x8
PA5M200	FULLPKG	NORTH_NE	QDR II+ x9
PA5M200	FULLPKG	NORTH_NW	DDR3
PA5M200	FULLPKG	NORTH_NW	DDR4
PA5M200	FULLPKG	NORTH_NW	QDR II+ x18
PA5M200	FULLPKG	NORTH_NW	QDR II+ x36
PA5M200	FULLPKG	NORTH_NW	QDR II+ x8
PA5M200	FULLPKG	NORTH_NW	QDR II+ x9
PA5M200	FULLPKG	SOUTH_SW	DDR3
PA5M200	FULLPKG	SOUTH_SW	QDR II+ x18
PA5M200	FULLPKG	SOUTH_SW	QDR II+ x8
PA5M200	FULLPKG	SOUTH_SW	QDR II+ x9
PA5M200	FULLPKG	WEST_NW	DDR3
PA5M200	FULLPKG	WEST_NW	QDR II+ x18
PA5M200	FULLPKG	WEST_NW	QDR II+ x36
PA5M200	FULLPKG	WEST_NW	QDR II+ x8
PA5M200	FULLPKG	WEST_NW	QDR II+ x9
PA5M200	FULLPKG	WEST_SW	DDR3
PA5M200	FULLPKG	WEST_SW	QDR II+ x18
PA5M200	FULLPKG	WEST_SW	QDR II+ x8
PA5M200	FULLPKG	WEST_SW	QDR II+ x9
PA5M300	FCG1152	NORTH_NE	DDR3
PA5M300	FCG1152	NORTH_NE	DDR4
PA5M300	FCG1152	NORTH_NE	QDR II+ x18
PA5M300	FCG1152	NORTH_NE	QDR II+ x36
PA5M300	FCG1152	NORTH_NE	QDR II+ x8
PA5M300	FCG1152	NORTH_NE	QDR II+ x9
PA5M300	FCG1152	NORTH_NW	DDR3
PA5M300	FCG1152	NORTH_NW	DDR4
PA5M300	FCG1152	NORTH_NW	QDR II+ x18
PA5M300	FCG1152	NORTH_NW	QDR II+ x36
PA5M300	FCG1152	NORTH_NW	QDR II+ x8
PA5M300	FCG1152	NORTH_NW	QDR II+ x9
PA5M300	FCG1152	SOUTH_SE	DDR3
PA5M300	FCG1152	SOUTH_SE	DDR4
PA5M300	FCG1152	SOUTH_SE	QDR II+ x8
PA5M300	FCG1152	SOUTH_SE	QDR II+ x9
PA5M300	FCG1152	SOUTH_SW	DDR3
PA5M300	FCG1152	SOUTH_SW	QDR II+ x18
PA5M300	FCG1152	SOUTH_SW	QDR II+ x8
PA5M300	FCG1152	SOUTH_SW	QDR II+ x9
PA5M300	FCG1152	WEST_NW	DDR3
PA5M300	FCG1152	WEST_NW	QDR II+ x18
PA5M300	FCG1152	WEST_NW	QDR II+ x36

PA5M300	FCG1152	WEST_NW	QDR II+ x8
PA5M300	FCG1152	WEST_NW	QDR II+ x9
PA5M300	FCG1152	WEST_SW	DDR3
PA5M300	FCG1152	WEST_SW	QDR II+ x18
PA5M300	FCG1152	WEST_SW	QDR II+ x36
PA5M300	FCG1152	WEST_SW	QDR II+ x8
PA5M300	FCG1152	WEST_SW	QDR II+ x9
PA5M300	FCG484	NORTH_NE	DDR3
PA5M300	FCG484	NORTH_NE	DDR4
PA5M300	FCG484	NORTH_NE	QDR II+ x18
PA5M300	FCG484	NORTH_NE	QDR II+ x8
PA5M300	FCG484	NORTH_NE	QDR II+ x9
PA5M300	FCG484	NORTH_NW	DDR3
PA5M300	FCG484	NORTH_NW	DDR4
PA5M300	FCG484	NORTH_NW	QDR II+ x18
PA5M300	FCG484	NORTH_NW	QDR II+ x8
PA5M300	FCG484	NORTH_NW	QDR II+ x9
PA5M300	FCG484	SOUTH_SW	DDR3
PA5M300	FCG484	SOUTH_SW	QDR II+ x18
PA5M300	FCG484	SOUTH_SW	QDR II+ x8
PA5M300	FCG484	SOUTH_SW	QDR II+ x9
PA5M300	FCG484	WEST_SW	DDR3
PA5M300	FCG484	WEST_SW	QDR II+ x8
PA5M300	FCG484	WEST_SW	QDR II+ x9
PA5M300	FCG784	NORTH_NE	DDR3
PA5M300	FCG784	NORTH_NE	DDR4
PA5M300	FCG784	NORTH_NE	QDR II+ x18
PA5M300	FCG784	NORTH_NE	QDR II+ x36
PA5M300	FCG784	NORTH_NE	QDR II+ x8
PA5M300	FCG784	NORTH_NE	QDR II+ x9
PA5M300	FCG784	NORTH_NW	DDR3
PA5M300	FCG784	NORTH_NW	DDR4
PA5M300	FCG784	NORTH_NW	QDR II+ x18
PA5M300	FCG784	NORTH_NW	QDR II+ x36
PA5M300	FCG784	NORTH_NW	QDR II+ x8
PA5M300	FCG784	NORTH_NW	QDR II+ x9
PA5M300	FCG784	SOUTH_SW	DDR3
PA5M300	FCG784	SOUTH_SW	QDR II+ x18
PA5M300	FCG784	SOUTH_SW	QDR II+ x8
PA5M300	FCG784	SOUTH_SW	QDR II+ x9
PA5M300	FCG784	WEST_NW	DDR3
PA5M300	FCG784	WEST_NW	QDR II+ x18
PA5M300	FCG784	WEST_NW	QDR II+ x36
PA5M300	FCG784	WEST_NW	QDR II+ x8
PA5M300	FCG784	WEST_NW	QDR II+ x9

PA5M300	FCG784	WEST_SW	DDR3
PA5M300	FCG784	WEST_SW	QDR II+ x18
PA5M300	FCG784	WEST_SW	QDR II+ x36
PA5M300	FCG784	WEST_SW	QDR II+ x8
PA5M300	FCG784	WEST_SW	QDR II+ x9
PA5M300	FCSG536	NORTH_NE	DDR3
PA5M300	FCSG536	NORTH_NE	DDR4
PA5M300	FCSG536	NORTH_NE	QDR II+ x18
PA5M300	FCSG536	NORTH_NE	QDR II+ x8
PA5M300	FCSG536	NORTH_NE	QDR II+ x9
PA5M300	FCSG536	NORTH_NW	DDR3
PA5M300	FCSG536	NORTH_NW	DDR4
PA5M300	FCSG536	NORTH_NW	QDR II+ x18
PA5M300	FCSG536	NORTH_NW	QDR II+ x8
PA5M300	FCSG536	NORTH_NW	QDR II+ x9
PA5M300	FCSG536	SOUTH_SW	DDR3
PA5M300	FCSG536	SOUTH_SW	QDR II+ x18
PA5M300	FCSG536	SOUTH_SW	QDR II+ x8
PA5M300	FCSG536	SOUTH_SW	QDR II+ x9
PA5M300	FCSG536	WEST_NW	DDR3
PA5M300	FCSG536	WEST_NW	QDR II+ x18
PA5M300	FCSG536	WEST_NW	QDR II+ x8
PA5M300	FCSG536	WEST_NW	QDR II+ x9
PA5M300	FCSG536	WEST_SW	DDR3
PA5M300	FCSG536	WEST_SW	QDR II+ x8
PA5M300	FCSG536	WEST_SW	QDR II+ x9
PA5M300	FCVG484	NORTH_NE	DDR3
PA5M300	FCVG484	NORTH_NE	DDR4
PA5M300	FCVG484	NORTH_NE	QDR II+ x18
PA5M300	FCVG484	NORTH_NE	QDR II+ x8
PA5M300	FCVG484	NORTH_NE	QDR II+ x9
PA5M300	FCVG484	NORTH_NW	DDR3
PA5M300	FCVG484	NORTH_NW	DDR4
PA5M300	FCVG484	NORTH_NW	QDR II+ x18
PA5M300	FCVG484	NORTH_NW	QDR II+ x8
PA5M300	FCVG484	NORTH_NW	QDR II+ x9
PA5M300	FCVG484	SOUTH_SW	DDR3
PA5M300	FCVG484	SOUTH_SW	QDR II+ x18
PA5M300	FCVG484	SOUTH_SW	QDR II+ x8
PA5M300	FCVG484	SOUTH_SW	QDR II+ x9
PA5M300	FCVG484	WEST_NW	DDR3
PA5M300	FCVG484	WEST_NW	QDR II+ x8
PA5M300	FCVG484	WEST_NW	QDR II+ x9
PA5M300	FCVG484	WEST_SW	DDR3
PA5M300	FCVG484	WEST_SW	QDR II+ x8

PA5M300	FCVG484	WEST_SW	QDR II+ x9
PA5M300	FULLPKG	NORTH_NE	DDR3
PA5M300	FULLPKG	NORTH_NE	DDR4
PA5M300	FULLPKG	NORTH_NE	QDR II+ x18
PA5M300	FULLPKG	NORTH_NE	QDR II+ x36
PA5M300	FULLPKG	NORTH_NE	QDR II+ x8
PA5M300	FULLPKG	NORTH_NE	QDR II+ x9
PA5M300	FULLPKG	NORTH_NW	DDR3
PA5M300	FULLPKG	NORTH_NW	DDR4
PA5M300	FULLPKG	NORTH_NW	QDR II+ x18
PA5M300	FULLPKG	NORTH_NW	QDR II+ x36
PA5M300	FULLPKG	NORTH_NW	QDR II+ x8
PA5M300	FULLPKG	NORTH_NW	QDR II+ x9
PA5M300	FULLPKG	SOUTH_SE	DDR3
PA5M300	FULLPKG	SOUTH_SE	DDR4
PA5M300	FULLPKG	SOUTH_SE	QDR II+ x8
PA5M300	FULLPKG	SOUTH_SE	QDR II+ x9
PA5M300	FULLPKG	SOUTH_SW	DDR3
PA5M300	FULLPKG	SOUTH_SW	QDR II+ x18
PA5M300	FULLPKG	SOUTH_SW	QDR II+ x8
PA5M300	FULLPKG	SOUTH_SW	QDR II+ x9
PA5M300	FULLPKG	WEST_NW	DDR3
PA5M300	FULLPKG	WEST_NW	QDR II+ x18
PA5M300	FULLPKG	WEST_NW	QDR II+ x36
PA5M300	FULLPKG	WEST_NW	QDR II+ x8
PA5M300	FULLPKG	WEST_NW	QDR II+ x9
PA5M300	FULLPKG	WEST_SW	DDR3
PA5M300	FULLPKG	WEST_SW	QDR II+ x18
PA5M300	FULLPKG	WEST_SW	QDR II+ x36
PA5M300	FULLPKG	WEST_SW	QDR II+ x8
PA5M300	FULLPKG	WEST_SW	QDR II+ x9
PA5M500	FCG1152	NORTH_NE	DDR3
PA5M500	FCG1152	NORTH_NE	DDR4
PA5M500	FCG1152	NORTH_NE	QDR II+ x18
PA5M500	FCG1152	NORTH_NE	QDR II+ x36
PA5M500	FCG1152	NORTH_NE	QDR II+ x8
PA5M500	FCG1152	NORTH_NE	QDR II+ x9
PA5M500	FCG1152	NORTH_NW	DDR3
PA5M500	FCG1152	NORTH_NW	DDR4
PA5M500	FCG1152	NORTH_NW	QDR II+ x18
PA5M500	FCG1152	NORTH_NW	QDR II+ x36
PA5M500	FCG1152	NORTH_NW	QDR II+ x8
PA5M500	FCG1152	NORTH_NW	QDR II+ x9
PA5M500	FCG1152	SOUTH_SE	DDR3
PA5M500	FCG1152	SOUTH_SE	DDR4

PA5M500	FCG1152	SOUTH_SE	QDR II+ x18
PA5M500	FCG1152	SOUTH_SE	QDR II+ x8
PA5M500	FCG1152	SOUTH_SE	QDR II+ x9
PA5M500	FCG1152	SOUTH_SW	DDR3
PA5M500	FCG1152	SOUTH_SW	QDR II+ x18
PA5M500	FCG1152	SOUTH_SW	QDR II+ x8
PA5M500	FCG1152	SOUTH_SW	QDR II+ x9
PA5M500	FCG1152	WEST_NW	DDR3
PA5M500	FCG1152	WEST_NW	QDR II+ x18
PA5M500	FCG1152	WEST_NW	QDR II+ x36
PA5M500	FCG1152	WEST_NW	QDR II+ x8
PA5M500	FCG1152	WEST_NW	QDR II+ x9
PA5M500	FCG1152	WEST_SW	DDR3
PA5M500	FCG1152	WEST_SW	QDR II+ x18
PA5M500	FCG1152	WEST_SW	QDR II+ x36
PA5M500	FCG1152	WEST_SW	QDR II+ x8
PA5M500	FCG1152	WEST_SW	QDR II+ x9
PA5M500	FCG784	NORTH_NE	DDR3
PA5M500	FCG784	NORTH_NE	DDR4
PA5M500	FCG784	NORTH_NE	QDR II+ x18
PA5M500	FCG784	NORTH_NE	QDR II+ x36
PA5M500	FCG784	NORTH_NE	QDR II+ x8
PA5M500	FCG784	NORTH_NE	QDR II+ x9
PA5M500	FCG784	NORTH_NW	DDR3
PA5M500	FCG784	NORTH_NW	DDR4
PA5M500	FCG784	NORTH_NW	QDR II+ x18
PA5M500	FCG784	NORTH_NW	QDR II+ x36
PA5M500	FCG784	NORTH_NW	QDR II+ x8
PA5M500	FCG784	NORTH_NW	QDR II+ x9
PA5M500	FCG784	SOUTH_SW	DDR3
PA5M500	FCG784	SOUTH_SW	QDR II+ x18
PA5M500	FCG784	SOUTH_SW	QDR II+ x8
PA5M500	FCG784	SOUTH_SW	QDR II+ x9
PA5M500	FCG784	WEST_NW	DDR3
PA5M500	FCG784	WEST_NW	QDR II+ x18
PA5M500	FCG784	WEST_NW	QDR II+ x36
PA5M500	FCG784	WEST_NW	QDR II+ x8
PA5M500	FCG784	WEST_NW	QDR II+ x9
PA5M500	FCG784	WEST_SW	DDR3
PA5M500	FCG784	WEST_SW	QDR II+ x18
PA5M500	FCG784	WEST_SW	QDR II+ x36
PA5M500	FCG784	WEST_SW	QDR II+ x8
PA5M500	FCG784	WEST_SW	QDR II+ x9
PA5M500	FULLPKG	NORTH_NE	DDR3
PA5M500	FULLPKG	NORTH_NE	DDR4

PA5M500	FULLPKG	NORTH_NE	QDR II+ x18
PA5M500	FULLPKG	NORTH_NE	QDR II+ x36
PA5M500	FULLPKG	NORTH_NE	QDR II+ x8
PA5M500	FULLPKG	NORTH_NE	QDR II+ x9
PA5M500	FULLPKG	NORTH_NW	DDR3
PA5M500	FULLPKG	NORTH_NW	DDR4
PA5M500	FULLPKG	NORTH_NW	QDR II+ x18
PA5M500	FULLPKG	NORTH_NW	QDR II+ x36
PA5M500	FULLPKG	NORTH_NW	QDR II+ x8
PA5M500	FULLPKG	NORTH_NW	QDR II+ x9
PA5M500	FULLPKG	SOUTH_SE	DDR3
PA5M500	FULLPKG	SOUTH_SE	DDR4
PA5M500	FULLPKG	SOUTH_SE	QDR II+ x18
PA5M500	FULLPKG	SOUTH_SE	QDR II+ x8
PA5M500	FULLPKG	SOUTH_SE	QDR II+ x9
PA5M500	FULLPKG	SOUTH_SW	DDR3
PA5M500	FULLPKG	SOUTH_SW	QDR II+ x18
PA5M500	FULLPKG	SOUTH_SW	QDR II+ x8
PA5M500	FULLPKG	SOUTH_SW	QDR II+ x9
PA5M500	FULLPKG	WEST_NW	DDR3
PA5M500	FULLPKG	WEST_NW	QDR II+ x18
PA5M500	FULLPKG	WEST_NW	QDR II+ x36
PA5M500	FULLPKG	WEST_NW	QDR II+ x8
PA5M500	FULLPKG	WEST_NW	QDR II+ x9
PA5M500	FULLPKG	WEST_NW	RLDRAM II
PA5M500	FULLPKG	WEST_SW	DDR3
PA5M500	FULLPKG	WEST_SW	QDR II+ x18
PA5M500	FULLPKG	WEST_SW	QDR II+ x36
PA5M500	FULLPKG	WEST_SW	QDR II+ x8
PA5M500	FULLPKG	WEST_SW	QDR II+ x9