Appendix A Physical Constraints Syntax Definition

A.1 I/O Constraints

The IO constraints can constrain port and Buffer to the specified IOB location.

Syntax

"IO_LOC" """obj_name""" obj_location ["exclusive"] ";"

Constraints Elements

obj_name

Obj_name can be the name of port and Buffer.

obj_location

Obj_location is the IOB location, such as "A11", "B12", etc. If multiple locations are specified, they need to be separated by commas, such as "A11, B2".

exclusive

Exclusive is optional, which indicates that the obj_location in the constraints can only place the primitives specified by obj_name after locations constraints.

Note!

If obj_name is the escaped name format (begin with backslash and end with space), the obj_name must be quoted on both sides.

Examples

Example 1

IO_LOC "io_1" A1;

// io_1 should be located to the pin A1.

Example 2

IO_LOC "io_1" A1, B14, A15;

// io_1 should be located to the pin A1, pin B14, or pin A15, one of the three locations will be taken for the placement.

Example 3

IO_LOC "io_2" A1 exclusive;

// io_2 should be located to the pin A1, and pin A1 can only be used by io_2.

Example 4

IO_LOC "io_2" A1, B14, A15 exclusive;

// io_2 should be located to pin A1, B14, or A15, and all these locations can only be used by io_2.

A.2 PORT Constraints

Port attributes constraints can set attribute values for ports, such as IO_TYPE, PULL_MODE and DRIVE, etc. You can see datasheets for the details.

Syntax

IO_PORT “port_name” attribute = attribute_value;

Multiple attributes can be set in a constraint statement. Each attribute can be separated by spaces.

Constraints Elements

It needs to constrain the port name, attribute and attribute value.

Examples

Example 1

IO_PORT "port_1" IO_TYPE = LVTTL33;

// Set the IO_TYPE as "LVTTL33".

Example 2

IO_PORT "port_2" IO_TYPE = LVTTL33 SLEW_RATE = FAST PULL_MODE =KEEPER;

// Set the IO_TYPE as the "LVTTL33', SLEW_RATE value is "FAST", PULL_MODE value is "KEEPER".

Example 3

IO_PORT "port_3" IO_TYPE = LVDS25;

// Buffer at port_3 is IBUF, and convert the IBUF to TLVDS_IBUF via the constraints.

A.3 Primitive Constraints

Primitive Constraints are used to place the instances to the specified GRIDs. LUT/BSRAM/SSRAM/DSP/PLL/DQS instances can be constrained using Primitive Constraints.

Syntax

"INS_LOC" """ obj_name""" obj_location ["exclusive"];"

Constraints Elements

obj_name

The instance name

obj_location

obj_location includes:

- A signal location is specified to LUT, such as, RxCy[0-3][A-B];
- A range of the locations are specified to the multiple rows or columns:
  - Include multiple CLS or LUT: "RxCy", "RxCy[0-3]"
  - Specify multiple rows: "R[x:y]Cm", "R[x:y]Cm[0-3]", "R[x:y]Cm[0-3][A-B]"
  - Specify multiple columns: "RxC[m:n]", "RxC[m:n][0-3]", "RxC[m:n][0-3][A-B]"
  - Specify multiple rows and columns: "R[x:y]C[m:n]", "R[x:y]C[m:n][0-3]", "R[x:y]C[m:n][0-3][A-B]"

Note!

Multiple ins_locations can be included in a constraint statement, and they are separated by ",".

PLL Constraints Location

For the "PLL_L" or "PLL_R" of the PLL constraints locations, if more than one PLL are placed on the left side, it can be set to "PLL_L[0]", "PLL_L[1]" ...; If more than one PLL are placed on the right side, it can be set to "PLL_R[0]", "PLL_R[1]" ...

BSRAM Constraints Location

The BSRAM constraints location is "BSRAM_R10[0]" (the first BSRAM at row 10), "BSRAM_R10[1]"….

DSP Constraints Location

The DSP constraints location is "DSP_R19[0]" (the first DSP Block at row 19), "DSP_R19[1]"… If it specifies a macro, it can be marked as: DSP_R19[0][A] or DSP_R19[0][B].

exclusive

Exclusive is optional, which indicates that the obj_location in the constraints can only place the instance specified by obj_name after locations constraints.

Examples

Example 1

INS_LOC "lut_1" R2C3, R5C10[0][A];

// lut_1 is constrained at the R2C3 and the first CLS of the R5C10 in the first LUT.

Example 2

INS_LOC "ins_2 " R5C6[2] exclusive;

// ins_2 is constrained at the third CLS of the R5C6, and only the instance can be placed at this location.

Example 3

INS_LOC "ins_3" R[2:6]C1;

// ins_1 is constrained between the row 2 and the row 6 and in the column 1.

Example 4

INS_LOC "ins_4" R[1:4]C[2:6] exclusive;

// ins_3 is constrained between row 2 and row 5, between column 3

and column 7. This location can only be occupied by this instance.

Example 5

INS_LOC "ins_5" R[1:4]C[2:6][1];

// ins_4 is constrained between row 2 and row 5, and in the second CLS of a GRID between column 3 and column 7.

Example 6

INS_LOC "reg_name" B14;

// It is constrained to the IOB B14 by REGISTER/IOLOGIC INS_LOC constraint.

Example 7

INS_LOC "pll_name" PLL_L;

// It is constrained to the PLL left by INS_LOC constraint.

Example 8

INS_LOC "bsram_name" BSRAM_R10[2];

// It is constrained to the third BSRAM in row 10 by INS_LOC constraint.

Example 9

INS_LOC "dsp_name" DSP_R19[2];

// It is constrained to the third DSP in row 19 by INS_LOC constraint.

A LUT1/LUT2/LUT3/LUT4 can be placed in LUT4. A LUT5 needs to occupy two LUT4s (one CLS). A LUT6 needs to occupy four LUT4s (two CLSs). A LUT7 needs to occupy four CLSs (one GRID). A LUT8 needs to occupy eight CLSs (two GRIDs). Therefore, for different instance constraints, the minimum unit of the constraint location is also different. For BSRAM/SSRAM/DSP (one DSP includes two MACROs and one MACRO includes two UNITs), the example is as follows.

Example 10

LUT4 Constraints

INS_LOC "lut4_name" R5C15[1][A];

// lut4_name is constrained to the second LUT in the first CLS of the R5C15.

Example 11

CLS Constraints

INS_LOC "lut5_name" R5C15[3];

// lut5_name is constrained to the fourth CLS of the R5C15.

Example 12

CLS Constraints

INS_LOC "lut6_name" R5C15[0];

// lut6_name is constrained to the first CLS of the R5C15 (the CLS[0] and CLS[1] will be occupied).

Example 13

GRID Constraints

INS_LOC "lut7_name" R5C15;

// lut7_name is constrained to the R5C15, and the LUT7 will occupy one GRID.

Example 14

GRID Constraints

INS_LOC "lut8_name" R5C15；

// lut8_name is constrained to the R5C15; lut8_name will occupy R5C15 and the R5C16.

Example 15

DSP MACRO Constraints

INS_LOC "mult_name" DSP_R19[1][A];

// mult_name is constrained to the first macro of the second DSP in row 19.

A4. Group Constraints

The group constraints include Primitive Group Constraints and Relative Group Constraints.

A.4.1 Primitive Group Constraints

Primitive Group Constraint is used to define a group constraint. A group is a collection of various instance objects. The instances such as LUT, DFF, etc., or Buffer, IOLOGIC, etc. can be added to a group using the Primitive Group constraints. And the location constraints of all objects in the group can be achieved by constraining the location of the group.

Syntax

Definition of GROUP:

GROUP group_name = { “obj_names ” } [exclusive];

Add the instance to the group:

GROUP group_name += { “obj_names ” } [exclusive];

The location of the group is constrained:

GRP_LOC group_name group_location[exclusive];

Note!

If group_name is the escaped name format (begin with backslash and end with space), the quotes at two sides of group_name are necessary.

Constraints Elements

group_name

Define a name as the name of the group.

obj_name

Obj_name is used to add the specified instance to the group.

group_location

Specify the constraints location of the group, and the group_location can at the IOB and the GRID.

exclusive

The "exclusive" is optional, which is at the end of the group definition or the location constraints;

An object can be included in multiple groups, but the object can only be included in the group that the "exclusive" is added;

The "exclusive" indicates that the constraints location can only be occupied by the objects in the group.

Examples

Example 1

GROUP group_1 = { "ins_1" "ins_2" "ins_3" "ins_4" };

// Create a group named group_1 and add the ins_1, ins_2, ins_3, ins_4 to the group.

Example 2

GROUP group_2 = { "ins_5" "ins_6" "ins_7" } exclusive;

// Create a group named group_2 and the ins_5, ins_6, ins_7 only can be added to this group.

Example 3

GROUP group_1 += { "io_1" "io_2"};

// Add io_1, io_2 to group_1.

Example 4

GRP_LOC group_1 R3C4, A14, B4;

// The objects in group_1 can be placed at R3C4, A14, B4.

Example 5

GRP_LOC group_2 R[1:3]C[1:4] exclusive;

// The Instance in group_2 can be placed in the range of R[1:3]C[1:4], and the instances in group_2 can only be placed in the range.

A.4.2 Relative Group Constraints

The instance relative location constraints can be realized using the Relative Group Constraints.

Syntax

Define Relative Group Constraints:

REL_GROUP group_name = { “obj_names ” };

Add the instance to the defined group:

REL_GROUP group_name += { “obj_names ” };

The instance relative location is constrained in the group:

INS_RLOC “obj_name” relative_location;

Constraints Elements

obj_name

The name of the constraint object.

relative_location

The description of the relative locations in row and column.

Examples

Example 1

```
REL_GROUP grp_1 = { "ins_1" "ins_2" "ins_3" "ins_4" };
INS_RLOC "ins_1" R0C0;
INS_RLOC "ins_2" R2C3;
INS_RLOC "ins_3" R3C5;
```

// Define a group constraint named grp_1 and add the ins_1, ins_2, ins_3, ins_4 to grp_1. The ins_1 is the relative location origin R0C0, the ins_2 is constrained to the R2C3 relative to the ins_1, and the ins_3 is constrained to the R3C5 relative to ins_1.

A.5 Resource Reservation Constraints

The specified location or range can be reserved using the Resource Reservation constraints.

Syntax

"LOC_RESERVE" location [res_obj] ";"

Examples

Example 1

```
LOC_RESERVE R2C3[0][A] -LUT;
LOC_RESERVE R2C3[0][A] -REG;
```

Example 2

```
LOC_RESERVE IOR3, IOR6, R2C3, R3C4;
```

Example 3

LOC_RESERVE R[2:5]C[3:6], R3C[8:9];

// The locations constraints in the above examples will be reserved during placement.

A.6 Vref Constraints

The chip supports the external reference voltage, which is valid for the BANK. The Vref Constraints can be used to constrain the name and location of the input pin of the external reference voltage.

Note!

- The input pin location where the external reference voltage can be set must have IOLOGIC resource.
- Vref Constraints and PORT constraints are valid when used together. When a single-ended input or inout port with IO Type SSTL/HSTL, the Vref attribute can be set to the created Vref Constraints, indicating that the reference voltage of this port uses the external reference voltage input from the Vref Constraints location.

Syntax

"USE_VREF_DRIVER" vref_name [location]";"

Constraints Elements

vref_name

Customized VREF pin name

location

Any IO location with IOLOGIC in the device can be used as a location for the VREF pin constraints.

Examples

Example 1

USE_VREF_DRIVER vref_pin;

IO_PORT “port_1” IO_TYPE = SSTL25_I VREF=vref_pin;

IO_PORT “port_2” IO_TYPE = SSTL25_I VREF=vref_pin;

// Define a VREF pin named "vref_pin" and set the port_1 and port_2 to vref_pin.

Example 2

USE_VREF_DRIVER vref_pin C7;

IO_PORT “port_1” IO_TYPE = SSTL25_I VREF=vref_pin;

IO_PORT “port_2” IO_TYPE = SSTL25_I VREF=vref_pin;

// Define a VREF pin named "vref_pin" and constrain it to C7. Set the VREF of port_1 and port_1 as vref_pin, and port_1 and port_1 will be placed in the bank where C7 locates.

A.7 Quadrant Constraints

The Quadrant Constraints is used to constrain objects such as DCS/DQCE to a specified quadrant.

Syntax

INS_LOC “obj_name” quadrant;

Constraints Elements

obj_name

The name of the constraint object.

quadrant

LittleBee® family: GW1N-9, GW1NR-9, GW1N-9C, GW1NR-9C can constrain 4 quadrants of "TOPLEFT", "TOPRIGHT", "BOTTOMLEFT", "BOTTOMRIGHT"; other devices can only constrain 2 quadrants of "LEFT", "RIGHT".

Arora family can constrain 4 quadrants of "TOPLEFT", "TOPRIGHT", "BOTTOMLEFT", "BOTTOMRIGHT".

Examples

Example 1

INS_LOC "dcs_name" LEFT;

// Constrain the DCS dcs_name to the LEFT quadrant.

A.8 Clock Assignment Constraints

The clock assignment constrains a specific net to the global clock wire or non-wire clock in the design. There are eight BUFGs and eight BUFSs in each quadrant of the chip resources. It can constrain the global clock wire for specific fanout (CLK/CE/SR/LOGIC) of the net.

- BUFG[0-7] represents the eight BUFGs.
- BUFS represents BUFS.
- LOCAL_CLOCK means this net is not to route the clock wire.

The CLK signal is the signal connected to the clock pin. The CE signal is the signal connected to the clock enable pin. The SR signal is the signal connected to the SET/RESET/CLEAR/PRESET pins, and the LOGIC is the signal connected to logic input pins.

Syntax

CLOCK_LOC “net_name” global_clocks = fanout ;

Constraints Elements

net_name

The net name

global_clocks

BUFG[0-7] represents the eight BUFGs.

BUFS represents BUFS.

LOCAL_CLOCK means this net is not to route clock wire.

fanout

CLK: fanout is the net of the clock pin.

CE: fanout is the net of the clock enable.

SR: fanout is the net of SET/RESET (synchronous reset signal), CLEAR/PRESET (asynchronous reset signal).

LOGIC: fanout is the net other than the fanout above.

The sign "|" can be used to separate multiple specified fanout.

Note!

If LOCAL_CLOCK is selected for LOCAL_CLOCK, fanout is not available.

Examples

Example 1

CLOCK_LOC "net" BUFG[0] = CLK;

// Constrain the net fanout as the clock pin net routing to the first BUFG.

Example 2

CLOCK_LOC "net" BUFG = CLK|CE;

NET_LOC "net" BUFG = CLK|CE;

// Constrain the net fanout as the clock pin/clock enable net routing to BUFG.

Example 3

CLOCK_LOC "net" BUFS = CE;

NET_LOC "net" BUFS = CE;

// Constrain the net fanout as the clock enable net routing to BUFS.

Example 4

CLOCK_LOC "net" LOCAL_CLOCK;

// Constrain the net not to route the clock line.

A.9 Hclk Constraints

The CLKDIV/DLLDLY can be constrained to the relevant locations via the CLKDIV/DLLDLY constraints. The constraints locations of CLKDIV/DLLDLY are different from the ones of other general instances. The "TOPSIDE", "BOTTOMSIDE", "LEFTSIDE", and "RIGHTSIDE" indicate the four sides of the constraints location.

Syntax

INS_LOC “obj_name” location;

Constraints Elements

obj_name

The instance name of the CLKDIV/DLLDLY is the obj_name.

location

"TOPSIDE[0-1]"

"BOTTOMSIDE[0-1]"

"LEFTSIDE[0-1]"

"RIGHTSIDE[0-1]"

Example

INS_LOC “clkdiv_name” TOPSIDE[0];

// Place the clkdiv_name to TOPSIDE[0].


GOWIN
PROGRAMMING FOR THE FUTURE