

How to Modify MOSFET PCells for Expert's Device Link

Expert 4.6.0.R has a device link feature which links a couple of PCells such as MOSFETs that share the source or drain region. This is useful for CMOS logic design, and simplifies editing of PCell placement.

Some of modifications are required on PCells for the device link:

- 1) Add some PCell parameters for device link
- 2) Write a call-back function to control PCell parameters on the device link
- 3) Add some attributes (linkdir, linkcallback) to pin objects in the PCell

The following parameters need to be added on MOSFET PCells for the device link.

- LeftLink / RightLink
- LeftCont / RightCont

First, a call-back function should be prepared to control PCell parameters for device link. The script needs to be written in Lisa or JavaScript. It sets a PCell parameter "LeftLink" TRUE when the left-side pin of the target MOSFET is shared with another one. For the opposite side, it sets "RightLink" to TRUE.

When the source or drain region is shared, duplicated contacts should be deleted. On one of the target PCells, another parameter "LeftCont" or "RightCont" should be set to FALSE.

These PCell parameters for the device link are controlled by a call-back function for the device link like in the following Lisa script:

```
define procedure "mos_linker" /replace
  parameter device1Params
  parameter device2Params
  parameter device1Pin
  parameter device2Pin
  parameter externalConn
do begin
  if (device1Pin EQL "S") then begin
    device1Params["RightCont"] = true;
    device1Params["RightLink"] = true;
  end;
  if (device1Pin EQL "D") then begin
    device1Params["LeftCont"] = true;
    device1Params["LeftLink"] = true;
  end;
  if (device2Pin EQL "S") then begin
    device2Params["RightCont"] = false;
    device2Params["RightLink"] = true;
  end;
  if (device2Pin EQL "D") then begin
    device2Params["LeftCont"] = false;
    device2Params["LeftLink"] = true;
  end;
end;
```

According the above Lisa script, the PCells are supposed to behave as shown in Figure 1.

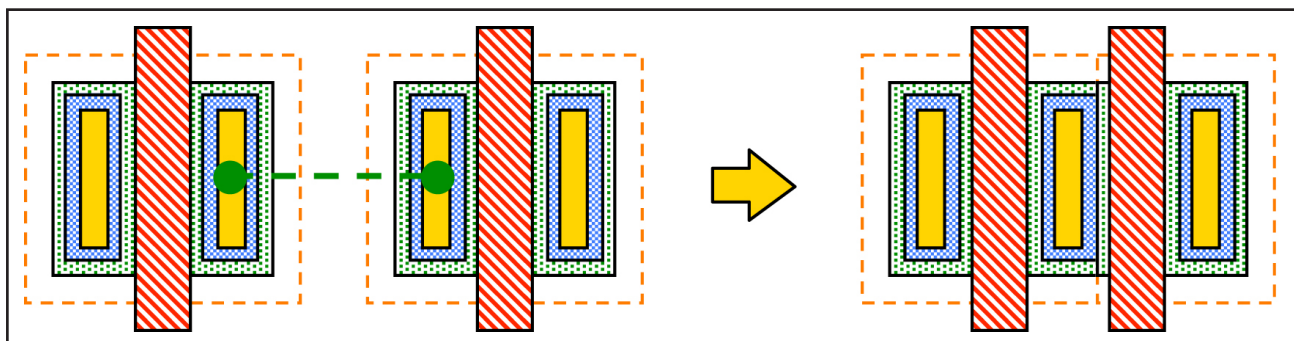


Figure 1. Linking MOSFET devices with a shared contact.

In this case, the original PCell has the following Lisa code:

```

DEFINE PCELL "PMOS" /REPLACE
  PARAMETER W /TYPE = (Double) /DEFAULT
= (6)
  PARAMETER L /TYPE = (Double) /DEFAULT
= (4)
  BODY BEGIN

    box (-L/2) (-W/2 - 4) (L) (W+8) /
layer="POLY" /net="g" /pin;
    box (-L/2-1) (-W/2+1) (-4) (W-2) /
layer="METAL1" /net="d" /pin;
    box (-L/2-2) (-W/2+2) (-2) (W-4) /
layer="CONTACT";
    box ( L/2+1) (-W/2+1) ( 4) (W-2) /
layer="METAL1" /net="s" /pin;
    box ( L/2+2) (-W/2+2) ( 2) (W-4) /
layer="CONTACT";
    box (-L/2-6) (-W/2) (L+12) (W) /
layer="ACTIVE";
    box (-L/2-8) (-W/2-2) (L+16) (W+4) /
layer="PIMP";

  END;

```

For the device link in Figure 1, some modifications should be added in the above Lisa script.

First, some PCell parameters should be added as follows:

```

DEFINE PCELL "PMOS" /REPLACE

  PARAMETER W /TYPE = (Double) /DEFAULT = (6)
  PARAMETER L /TYPE = (Double) /DEFAULT = (4)
  PARAMETER LeftLink /TYPE = (Boolean) /
  DEFAULT = (FALSE)
  PARAMETER RightLink /TYPE = (Boolean) /
  DEFAULT = (FALSE)
  PARAMETER LeftCont /TYPE = (Boolean) /
  DEFAULT = (TRUE)
  PARAMETER RightCont /TYPE = (Boolean) /
  DEFAULT = (TRUE)
  BODY BEGIN

```

Next, a few attributes need to be added on pin objects to enable the device link on this PCell.

```

box (-L/2) (-W/2 + 4) (L) (W+8) /
layer="POLY" /net="g" /pin;

leftpin =(box (-L/2-1) (-W/2+1) (-4) (W-2) /
layer="METAL1" /net="d" /pin /create /get);
if (leftpin NEQ NIL) then begin
  leftpin.linkdir = DLD_LEFT;
  leftpin.linkcallback = "mos_linker";
end;
box (-L/2-2) (-W/2+2) (-2) (W-4) /
layer="CONTACT";

rightpin = (box ( L/2+1) (-W/2+1) ( 4) (W-
2) /layer="METAL1" /net="s" /pin /create
/get);
if (rightpin NEQ NIL) then begin
  rightpin.linkdir = DLD_RIGHT;
  rightpin.linkcallback = "mos_linker";
end;
box ( L/2+2) (-W/2+2) ( 2) (W-4) /
layer="CONTACT";

```

The direction for linking with another PCell should be specified in an attribute <LayoutObject>. linkdir. The name of call-back function written as above should be set in another attribute "<LayoutObject>. Linkcallback. With those attributes, the call-back function is called when the device link is executed.

Finally, some modifications should be added to modify the shape of the PCell according to the added parameters for the device link.

The shape of the linked side where the contacts are deleted should be deleted as shown in Figure 2.

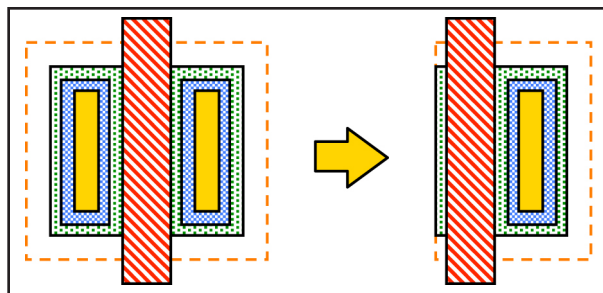


Figure 2. Modified MOSFET p-cell to be linked without a shared contact.

In this case, the PCell parameter should be set as follows:

```

LeftLink = true / LeftCont = false
RightLink = false / RightCont = true

```

According to the above PCell parameters, the contact objects and metal objects around the contacts should be deleted, just like the following:

```

if (LeftCont) then begin
  leftpin =(box (-L/2-1) (-W/2+1) (-4)
    (W-2) /layer="METAL1" /net="d" /pin /
    create /get);
  if (leftpin NEQ NIL) then begin
    leftpin.linkdir = DLD_LEFT;
    leftpin.linkcallback = "mos_linker";
  end;
  box (-L/2-2) (-W/2+2) (-2) (W-4) /
    layer="CONTACT";
end;

if (RightCont) then begin
  rightpin = (box ( L/2+1) (-W/2+1) ( 4)
    (W-2) /layer="METAL1" /net="s" /pin /
    create /get);
  if (rightpin NEQ NIL) then begin
    rightpin.linkdir = DLD_RIGHT;
    rightpin.linkcallback = "mos_linker";
  end;
  box ( L/2+2) (-W/2+2) ( 2) (W-4) /
    layer="CONTACT";
end;

```

Also, the corresponding source, or drain region, should shrink.

This requires the following modification in the PCell code.

```

active_dx_left = 0;
select_dx_left = 0;
active_dx_right = 0;
select_dx_right = 0;

if (not LeftCont) then begin
  active_dx_left = 5;
  select_dx_left = 7;
  active_dx_right = 5;
  select_dx_right = 7;
end;

if (not RightCont) then begin
  active_dx_right = active_dx_right +
5;
  select_dx_right = select_dx_right + 7;
end;
box (-L/2-6 + active_dx_left) (-W/2)
(L+12
- active_dx_right) (W) /layer="ACTIVE";
box (-L/2-8 + select_dx_left) (-W/2-2)
(L+16
- select_dx_right) (W+4) /layer="PIMP";

```

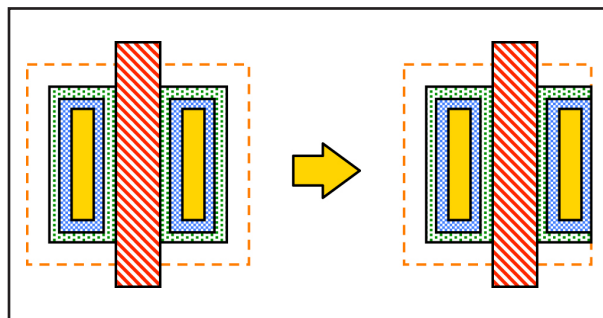


Figure 3. Modified MOSFET p-cell to be linked with the shared contact.

Next, the shape of the linked side with contacts should be modified as shown in Figure 3.

In this case, the PCell parameter should be set as follows:

```

LeftLink = false / LeftCont = true

RightLink = true / RightCont = true

```

The linked PCells connect with each other on the cell bounding box. Therefore, the size and shape of the linked PCells should be modified properly so that they touch.

In this example, the shapes in ACTIVE and PIMP layers should be modified as follows:

```

if (not LeftCont) then begin
  active_dx_left = 5;
  select_dx_left = 7;
  active_dx_right = 5;
  select_dx_right = 7;
end;

if (not RightCont) then begin
  active_dx_right = active_dx_right + 5;
  select_dx_right = select_dx_right + 7;
end;

if (LeftCont and LeftLink) then begin
  active_dx_left = 1;
  select_dx_left = 3;
  active_dx_right = 1;
  select_dx_right = 3;
end;

if (RightCont and RightLink) then begin
  active_dx_right = active_dx_right + 1;
  select_dx_right = select_dx_right + 3;
end;

box (-L/2-6 + active_dx_left) (-W/2)
(L+12
- active_dx_right) (W) /layer="ACTIVE";
box (-L/2-8 + select_dx_left) (-W/2-2)
(L+16
- select_dx_right) (W+4) /layer="PIMP";

```

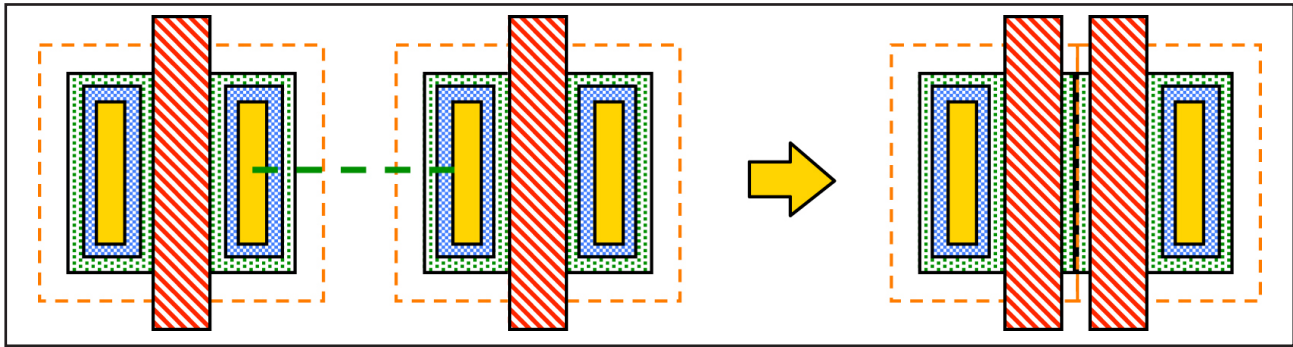


Figure 4. Linking MOSFET devices without a shared contact.

If the shared source / drain region on the pair of linked MOSFETs does not have any other connection to an external object, the contact objects in this region can be deleted for a smaller cell area (Figure 4).

In this case, the call-back function for the device link should be modified as follows:

```

define procedure "mos_linker" /replace
  parameter device1Params
  parameter device2Params
  parameter device1Pin
  parameter device2Pin
  parameter externalConn
  do begin
    if (device1Pin EQL "S") then begin
      device1Params["RightCont"] =
        externalConn;
      device1Params["RightLink"] = true;
    end;
    if (device1Pin EQL "D") then begin
      device1Params["LeftCont"] =
        externalConn;
      device1Params["LeftLink"] = true;
    end;
    if (device2Pin EQL "S") then begin
      device2Params["RightCont"] = false;
      device2Params["RightLink"] = true;
    end;
    if (device2Pin EQL "D") then begin
  
```

```

    device2Params["LeftCont"] = false;
    device2Params["LeftLink"] = true;
  end;
end;
end;

```

If the shared source / drain region does not have any external connection to another object, the parameter "externalConn" has the value FALSE in the call-back function. This allows removing shared contacts without any modification in the PCell code.

Wide space is required between MOSFET gates to keep the design rule, therefore some modification is required in the PCell code. (Figure 5)

For this modification, the following parameters should be added in the PCell code to control the width of the source / drain regions.

```

DEFINE PCELL "PMOS" /REPLACE
  PARAMETER W /TYPE = (Double) /DEFAULT = (6)
  PARAMETER L /TYPE = (Double) /DEFAULT = (4)
  PARAMETER LeftLink /TYPE = (Boolean) /
    DEFAULT = (FALSE)
  PARAMETER RightLink /TYPE = (Boolean) /
    DEFAULT = (FALSE)
  PARAMETER LeftCont /TYPE = (Boolean) /
    DEFAULT = (TRUE)
  PARAMETER RightCont /TYPE = (Boolean) /
    DEFAULT = (TRUE)

```

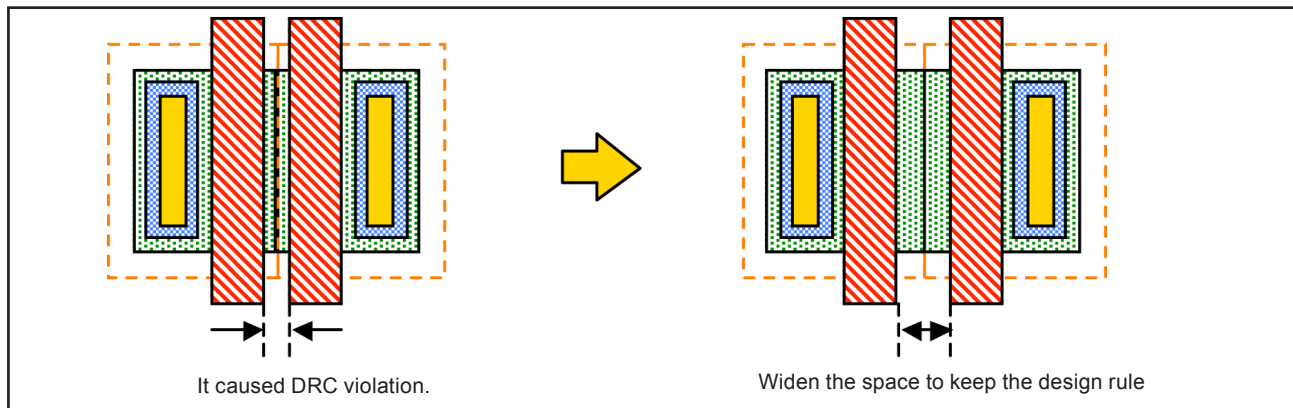


Figure 5. Linking MOSFET devices with enough spacing between each gate.

```

PARAMETER LeftStretch /TYPE = (Boolean) /
  DEFAULT = (FALSE)
PARAMETER RightStretch /TYPE = (Boolean) /
  DEFAULT = (FALSE)
BODY BEGIN

```

Next, the call-back function should be modified as follows:

```

define procedure "mos_linker" /replace
parameter device1Params
parameter device2Params
parameter device1Pin
parameter device2Pin
parameter externalConn
do begin
if (device1Pin EQL "S") then begin
device1Params["RightCont"] =
  externalConn;
device1Params["RightLink"] = true;
device1Params["RightStretch"] =
  not externalConn;
end;
if (device1Pin EQL "D") then begin
device1Params["LeftCont"] =
  externalConn;
device1Params["LeftLink"] = true;
device1Params["LeftStretch"] =
  not externalConn;
end;
if (device2Pin EQL "S") then begin
device2Params["RightCont"] = false;
device2Params["RightLink"] = true;
device2Params["RightStretch"] =
  not externalConn;
end;
if (device2Pin EQL "D") then begin
device2Params["LeftCont"] = false;
device2Params["LeftLink"] = true;
device2Params["LeftStretch"] =
  not externalConn;
end;
end;
end;

```

Finally, the following script should be added to change the shape of the source / drain regions.

```

if (not LeftCont and not LeftStretch)
  then begin
    active_dx_left = 5;
    select_dx_left = 7;
    active_dx_right = 5;
    select_dx_right = 7;
  end;
if (not LeftCont and LeftStretch)

```

```

  then begin
    active_dx_left = 4;
    select_dx_left = 6;
    active_dx_right = 4;
    select_dx_right = 6;
  end;

```

```

if (not RightCont and not RightStretch)
  then begin
    active_dx_right = active_dx_right + 5;
    select_dx_right = select_dx_right + 7;
  end;
if (not RightCont and RightStretch)
  then begin
    active_dx_right = active_dx_right + 4;
    select_dx_right = select_dx_right + 6;
  end;

```

To summarize, the modified script is as follows:

<Call-back function for device link>

```

define procedure "mos_linker" /replace
parameter device1Params
parameter device2Params
parameter device1Pin
parameter device2Pin
parameter externalConn
do begin
if (device1Pin EQL "S") then begin
device1Params["RightCont"] =
  externalConn;
device1Params["RightLink"] = true;
device1Params["RightStretch"] =
  not externalConn;
end;
if (device1Pin EQL "D") then begin
device1Params["LeftCont"] =
  externalConn;
device1Params["LeftLink"] = true;
device1Params["LeftStretch"] =
  not externalConn;
end;
if (device2Pin EQL "S") then begin
device2Params["RightCont"] = false;
device2Params["RightLink"] = true;
device2Params["RightStretch"] =
  not externalConn;
end;
if (device2Pin EQL "D") then begin
device2Params["LeftCont"] = false;
device2Params["LeftLink"] = true;
device2Params["LeftStretch"] =
  not externalConn;
end;
end;
end;

```

<Lisa script for PCell definition>

```

DEFINE PCELL "PMOS" /REPLACE
PARAMETER W /TYPE = (Double) /
  DEFAULT = (6)
PARAMETER L /TYPE = (Double) /
  DEFAULT = (4)
PARAMETER LeftLink /TYPE = (Boolean)/
  DEFAULT = (FALSE)
PARAMETER RightLink /TYPE = (Boolean) /
  DEFAULT = (FALSE)
PARAMETER LeftCont /TYPE = (Boolean) /
  DEFAULT = (TRUE)
PARAMETER RightCont /TYPE = (Boolean)/
  DEFAULT = (TRUE)
PARAMETER LeftStretch /TYPE = (Boolean)/
  DEFAULT = (FALSE)
PARAMETER RightStretch /TYPE = (Boolean) /
  DEFAULT = (FALSE)
BODY BEGIN

```

```

box (-L/2) (-W/2 - 4) (L) (W+8) /
  layer="POLY" /net="g" /pin;

```

```

if (LeftCont) then begin
  leftpin =(box (-L/2-1) (-W/2+1) (-4)
    (W-2) /layer="METAL1" /net="d" /pin /
    create /get);
  if (leftpin NEQ NIL) then begin
    leftpin.linkdir = DLD_LEFT;
    leftpin.linkcallback = "mos_linker";
  end;
  box (-L/2-2) (-W/2+2) (-2) (W-4) /
    layer="CONTACT";
end;

```

```

if (RightCont) then begin
  rightpin = (box ( L/2+1) (-W/2+1) ( 4)
    (W-2) /layer="METAL1" /net="s" /pin /
    create /get);
  if (rightpin NEQ NIL) then begin
    rightpin.linkdir = DLD_RIGHT;
    rightpin.linkcallback = "mos_linker";
  end;
  box ( L/2+2) (-W/2+2) ( 2) (W-4) /
    layer="CONTACT";
end;

```

```

active_dx_left = 0;
select_dx_left = 0;
active_dx_right = 0;
select_dx_right = 0;

```

```

if (not LeftCont and not LeftStretch)
  then begin
    active_dx_left = 5;
    select_dx_left = 7;
    active_dx_right = 5;
    select_dx_right = 7;
  end;
if (not LeftCont and LeftStretch)
  then begin

```

```

  active_dx_left = 4;
  select_dx_left = 6;
  active_dx_right = 4;
  select_dx_right = 6;
end;

if (not RightCont and not RightStretch)
  then begin
    active_dx_right = active_dx_right + 5;
    select_dx_right = select_dx_right + 7;
  end;
if (not RightCont and RightStretch)
  then begin
    active_dx_right = active_dx_right + 4;
    select_dx_right = select_dx_right + 6;
  end;

if (LeftCont and LeftLink) then begin
  active_dx_left = 1;
  select_dx_left = 3;
  active_dx_right = 1;
  select_dx_right = 3;
end;

if (RightCont and RightLink) then begin
  active_dx_right = active_dx_right + 1;
  select_dx_right = select_dx_right + 3;
end;

box (-L/2-6 + active_dx_left)
  (-W/2) (L+12 - active_dx_right) (W) /
  layer="ACTIVE";
box (-L/2-8 + select_dx_left) (-W/2-2)
  (L+16 - select_dx_right) (W+4) /
  layer="PIMP";

end;

```

The actual behavior of the PCells is shown in Figure 6.

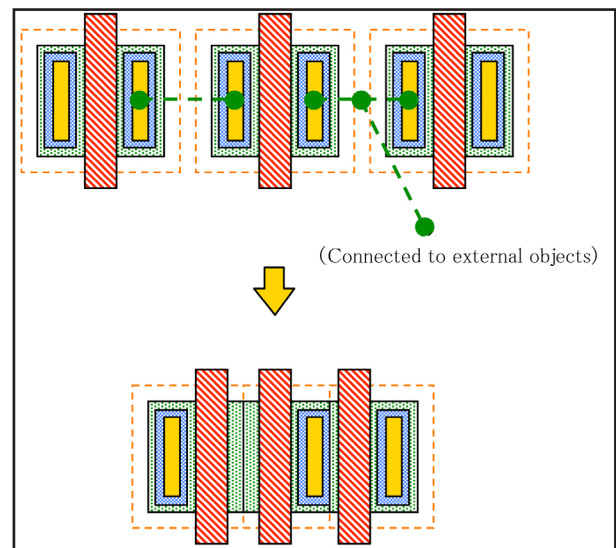


Figure 6. A combination of linked MOSFET devices in an actual design.