

# Expertのデバイス・リンク機能を使用するためのP-Cellの修正方法 ＜MOSFET編＞

Expert 4.6.0.Rには、デバイス・リンク機能があります。これは、拡散領域を共有するMOSFETなどの複数のP-Cellをリンクする機能で、CMOS論理設計に役立ちます。また、P-Cell配置の編集を簡略化できます。

一般的なP-Cellにデバイス・リンク機能を実装するためには、以下の作業が必要になります

- 1) P-Cellに対して、デバイス・リンク用のパラメータを追加
- 2) デバイス・リンクの際の属性をコントロールするためのコールバック関数を作成
- 3) P-Cell内のピン図形に対して、デバイス・リンク用の属性（linkdir, linkcallback）を追加

MOSFETの場合、以下のようなP-Cellパラメータを追加します。

- LeftLink / RightLink
- LeftCont / RightCont

リンク対象となるデバイスの左側の端子が共有される場合は、LeftLinkを有効にします。逆の場合、RightLinkを有効にします。

また、リンクされる場合には、重複するコンタクトを消す必要がありますので、どちらかのP-Cellインスタンスについて、LeftContまたはRightContを無効に設定します。

これらのパラメータは、次のLISAスクリプトのようなデバイス・リンク用コールバック関数により、制御されます。このコールバック関数は、LISAスクリプトまたはJavaScriptで作成できます。

```
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;
```

この設定により、P-Cellは図1のように動作するものとします。

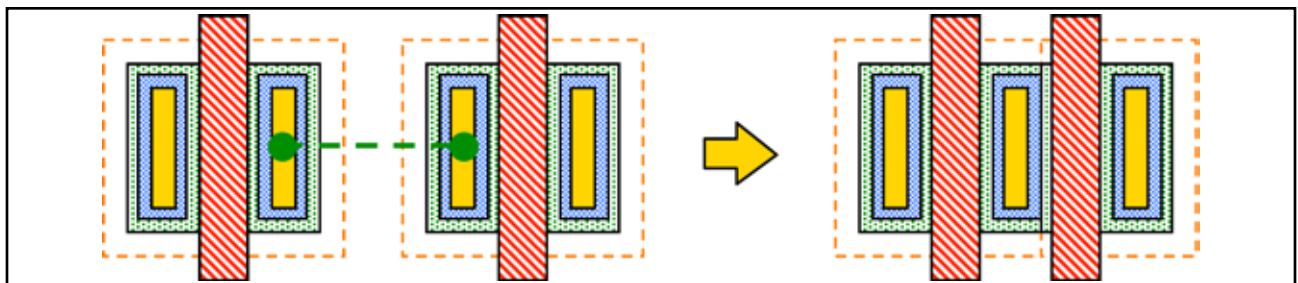


図1. MOSFETデバイスをリンクし、コンタクトを共有

元のP-Cellコードは以下のようになります。

```

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;

```

では、実際に、図1のデバイス・リンクに必要なP-Cellの設定を、順を追って説明していきます。

まず、デバイス・リンクに必要なデバイス・パラメータを、以下のように追加します。

```

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

```

次に、デバイス・リンクに必要な属性を、ピン図形に追加します。

```

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";

```

<LayoutObject>.linkdirには、他のデバイスとリンクする辺を指定します。また、<LayoutObject>. linkcallbackには、デバイス・リンクの際の属性の設定を規定するXスクリプトのユーザー定義関数を指定します。これにより、デバイス・リンクのコマンドが実行されると、この関数が呼び出されます。

最後に、設定されたパラメータに応じて、図形の形状を変更できるように、記述を追加します。

まず、コンタクトを消去する方の図形の形状は、図2のようになります。

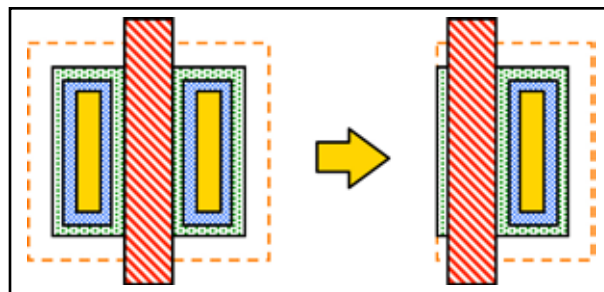


図2. コンタクトを削除する側の変更後のMOSFET P-Cell

この際のデバイス・パラメータは、以下のようになります。

```

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

```

設定されたデバイス・パラメータに従って、コンタクト、およびメタル図形を削除する必要があります。具体的なスクリプトの変更例は、以下のようになります。

```

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;

```

また、コンタクトが消えるのと同時に拡散層の領域も小さくする必要があります。

具体的には、以下のよう修正します。

```

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";

```

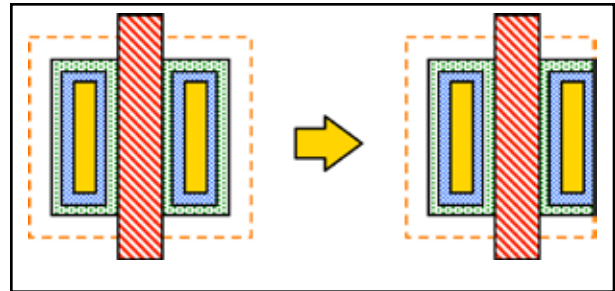


図3. コンタクトを残す側の変更後のMOSFET P-Cell

次に、コンタクトを残す方の図形の形状は、図3のようになります。

この際のデバイス・パラメータは、以下のようになります。

```

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

```

デバイス・リンク機能では、リンクされるデバイスは、お互いにそのセル・バウンディング・ボックスで接する形となります。そのため、デバイス・リンクの際には、適切な形で他の図形と接するよう、セルを構成する図形の寸法を調整する必要があります。

今回の例では、ACTIVEレイヤ、およびPIMPレイヤについて、記述を修正します。

具体的には、以下のようになります。

```

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;

```

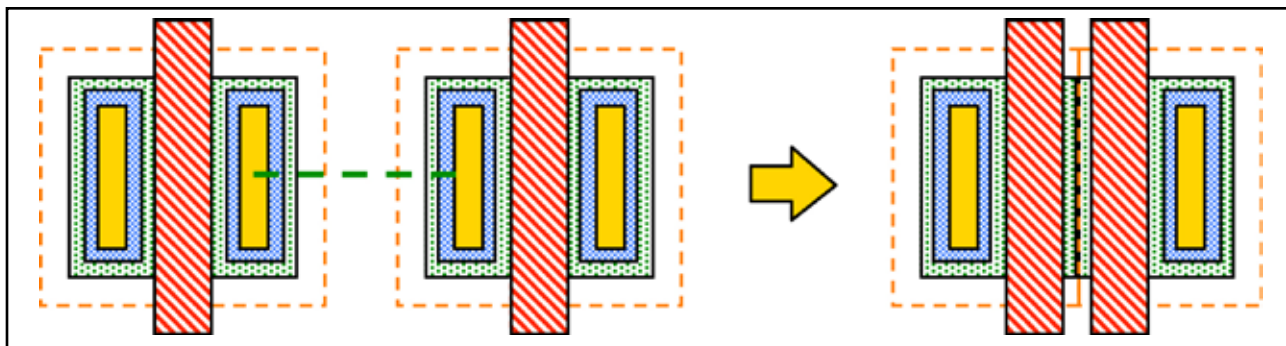


図4. MOSFETデバイスをリンクし、共有コンタクトを消去

```

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";

```

2つのMOSFETに共有される拡散層領域が、他の図形への接続を持たない場合、この領域のコンタクトを消去することができます(図4を参照)。

この場合、まず、デバイス・リンクの際に呼び出されるコールバック関数を、以下のように書き換えます。

```

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;

```

共有される拡散層領域から他の図形への接続がない場合、コールバック関数のパラメータexternalConnにはfalseが入ります。これにより、P-Cell側のコードの修正を必要とせず、共有コンタクトを消去することが可能です。

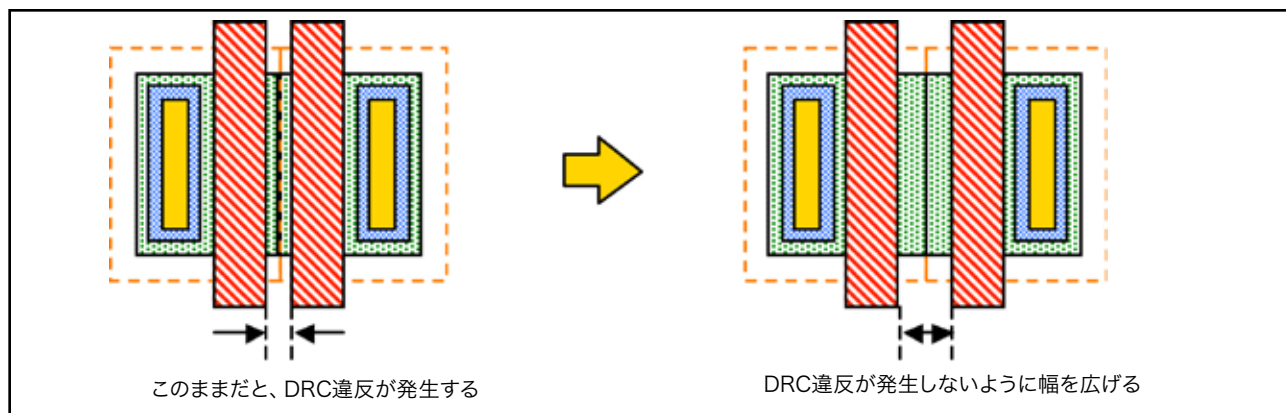


図5. MOSFETデバイスのリンク時に、ゲート間に十分な間隔を確保

ただし、拡散層の領域を任意の大きさに広げる必要がある場合は、コードの修正が必要です(図5を参照)。

このように修正するためには、まず、P-Cell側で、幅の調整に必要なパラメータを追加します。

```
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
```

次に、デバイス・リンク用コールバック関数を、以下のように修正します。

```
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;
```

最後に、拡散層の形状を変更するために、以下のような記述を追加します。

```
    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;
```

最終的に出来上がったスクリプトは以下ようになります。  
<デバイス・リンク用コールバック関数>

```
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;

```

#### <P-Cell定義スクリプト>

```

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;

```

実際の動作は、図6のようになります。

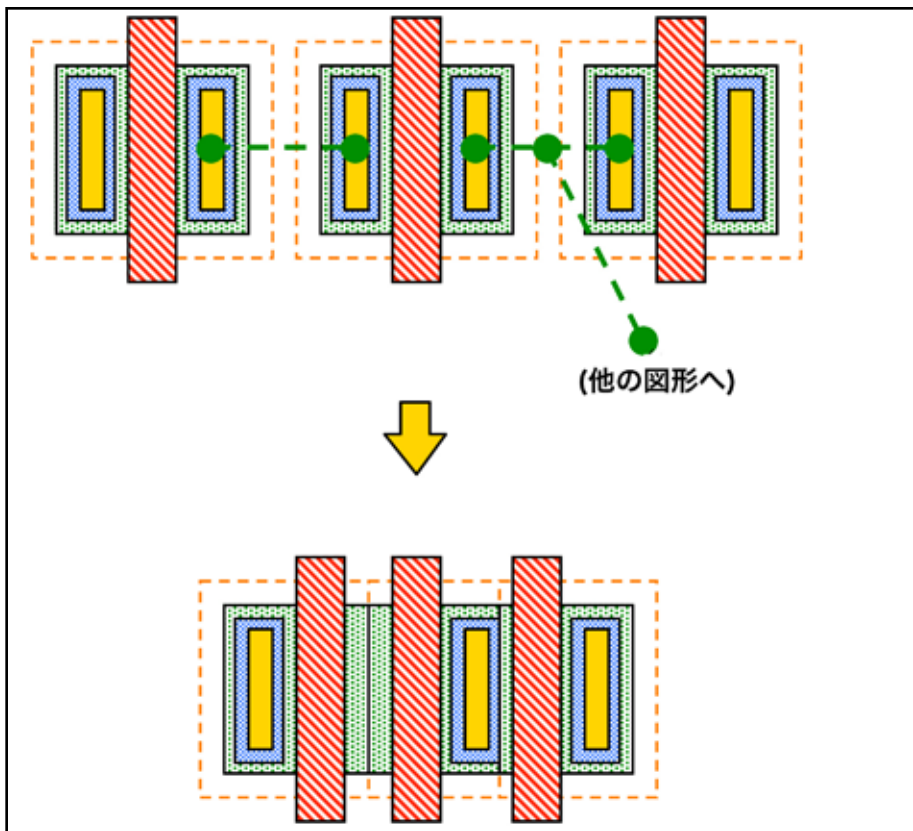


図6. 実際のデザインにおける、リンクされたMOSFETデバイスの組み合わせ