

# Expert의 소자 연결을 위한 MOSFET PCell 수정

Expert 4.6.0.R은 소스 또는 드레인 영역을 공유하는 MOSFET과 같이 PCell을 연결하는 소자 연결 기능이 있습니다. CMOS 논리 설계에 유용하며, 간단하게 PCell 배치를 편집할 수 있습니다.

소자 연결을 위한 PCell 수정 사항:

- 1) 소자 연결을 위한 PCell 파라미터 추가
- 2) 소자 연결에 관한 PCell 파라미터를 제어하기 위해 콜백 함수 작성
- 3) PCell의 핀 객체에 속성 (linkdir, linkcallback) 추가

소자 연결을 위해 MOSFET PCells에 다음 파라미터를 추가합니다.

- LeftLink / RightLink
- LeftCont / RightCont

우선, 소자 연결을 위해 PCell 파라미터를 제어하기 위한 콜백 함수가 필요합니다. Lisa 또는 자바스크립트로 스크립트를 작성합니다. 대상 MOSFET의 왼쪽 핀을 공유할 때, PCell 파라미터 "LeftLink"를 TRUE로 설정합니다. 반대의 경우, "RightLink"를 TRUE로 설정합니다.

소스 또는 드레인 영역을 공유할 경우, 중복된 컨택을 삭제합니다. 대상 PCell 중에서 다른 파라미터 "LeftCont" 또는 "RightCont"를 FALSE로 설정합니다.

소자 연결을 위한 PCell 파라미터는 아래의 Lisa 스크립트처럼 콜백 함수로 제어합니다:

```
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;
```

위의 Lisa 스크립트에 따라 PCell은 그림 1과 같이 동작합니다.

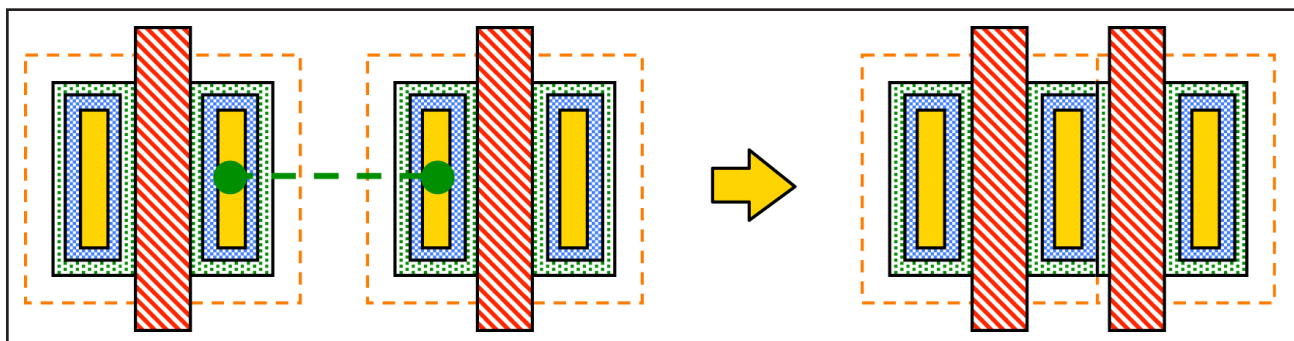


그림 1. MOSFET 소자를 공유 컨택으로 연결

이 경우, 원본 PCell의 Lisa 코드는 아래와 같습니다:

```

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;
  
```

그림 1의 소자 연결을 위해, 위의 Lisa 스크립트를 수정합니다.

먼저, 아래와 같이 PCell 파라미터를 추가합니다:

```

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
  
```

다음으로, 소자 연결을 위해 핀 객체에 속성을 추가합니다.

```

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";
  
```

다른 PCell에 대한 연결 방향은 <LayoutObject>.linkdir 속성에서 지정합니다. 위에서 작성한 콜백 함수의 이름은 <LayoutObject>.Linkcallback 속성에서 설정합니다. 이러한 속성을 이용하여 소자를 연결할 때 콜백 함수를 호출합니다.

마지막으로, 소자 연결을 위해 추가된 파라미터에 따라 PCell을 변형하기 위한 수정을 추가합니다.

컨택을 삭제한 연결 부분은 그림 2와 같이 삭제합니다.

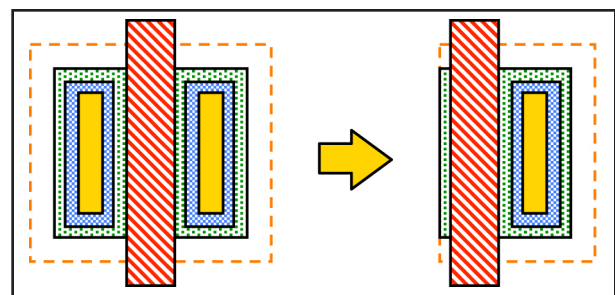


그림 2. 컨택을 공유하지 않고 연결한 MOSFET PCell

이 경우, PCell 파라미터는 다음과 같이 설정합니다:

LeftLink = true / LeftCont = false

RightLink = false / RightCont = true

위의 PCell 파라미터에 따라, 다음과 같이 컨택 주위의 컨택 객체 및 메탈 객체를 삭제합니다:

```
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;
```

또한, 해당 소스 또는 드레인 영역을 축소합니다.

이를 위해, PCell 코드를 다음과 같이 수정합니다.

```
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";
```

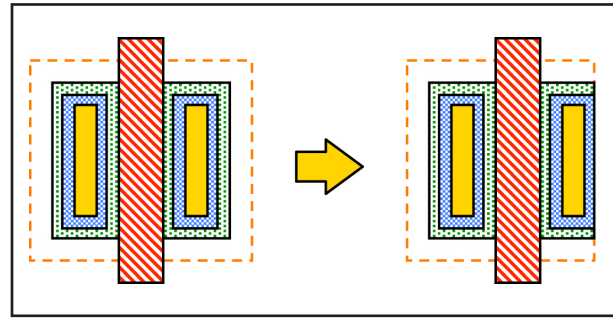


그림 3. 컨택을 공유하여 연결한 MOSFET p-cell

다음으로, 컨택으로 연결한 부분의 형태를 그림 3과 같이 수정합니다.

이 경우, PCell 파라미터는 다음과 같이 설정합니다.

LeftLink = false / LeftCont = true

RightLink = true / RightCont = true

연결된 PCell은 셀의 경계에서 서로 연결됩니다. 연결된 PCell이 접할 수 있도록, 크기와 모양을 적절하게 수정합니다.

예시의 ACTICE 및 PIMP 레이어의 모양을 다음과 같이 수정합니다.

```
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";
```

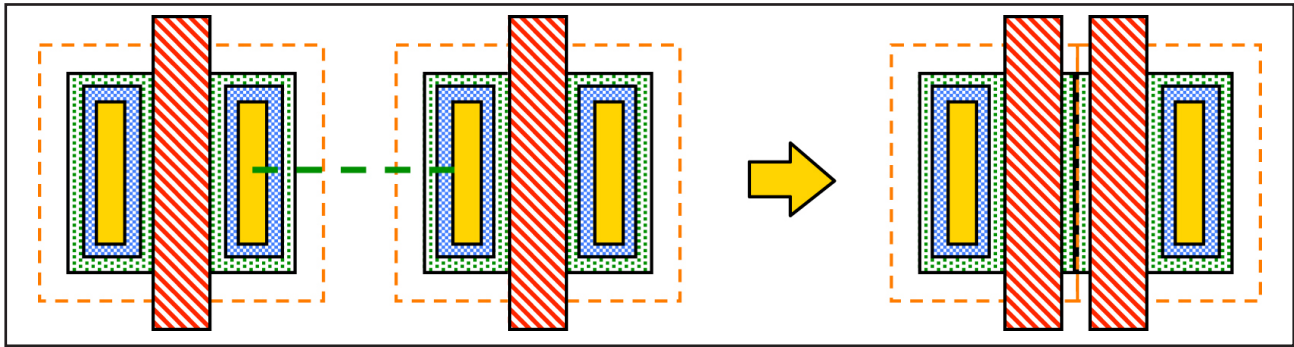


그림 4. 컨택을 공유하지 않고 연결한 MOSFET 소자

연결한 MOSFET 쌍의 공유 소스 / 드레인 영역이 외부 객체에 연결되지 않은 경우, 더 작은 셀 영역을 위해 이 영역의 컨택 객체를 삭제할 수 있습니다 (그림 4).

이 때, 소자 연결을 위한 콜백 함수를 다음과 같이 수정합니다:

```
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;
```

공유 소스 / 드레인 영역이 외부 객체에 연결되지 않은 경우, 콜백 함수의 "externalConn" 파라미터는 FALSE 입니다. 이를 통해, PCell 코드를 변경하지 않고 공유 컨택을 제거할 수 있습니다.

설계 규칙을 유지하려면 MOSFET 게이트 사이에 넓은 공간이 필요하며, PCell 코드를 일부 수정해야 합니다. (그림 5)

다음 파라미터를 PCell 코드에 추가하여, 소스 / 드레인 영역의 폭을 제어합니다.

```
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)
```

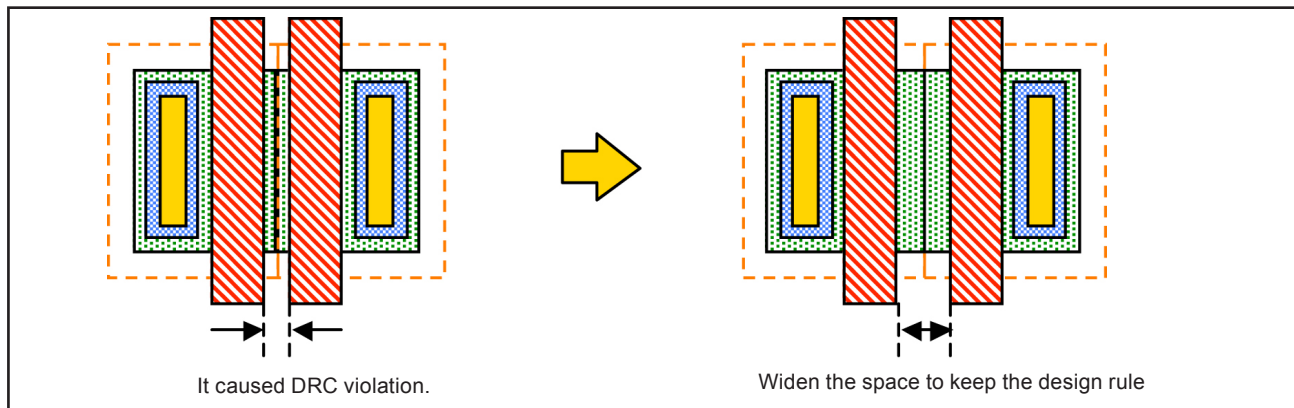


그림 5. 게이트 사이에 충분한 간격을 두고 MOSFET 소자를 연결

```

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

```

다음으로, 콜백 함수를 아래와 같이 수정합니다:

```

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;

```

마지막으로, 소스 / 드레인 영역의 모양을 변경하기 위해 다음 스크립트를 추가합니다.

```

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;

```

요약하면, 수정된 스크립트는 다음과 같습니다:

<소자 연결 콜백 함수>

```

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;

```

<PCell 정의 Lisa 스크립트>

```

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;
```

PCell의 실제 동작을 그림 6에서 보여줍니다.

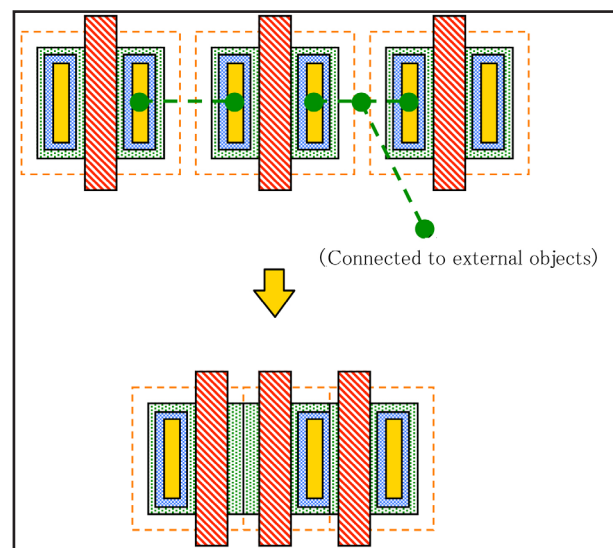


그림 6. 실제 설계에서 연결된 MOSFET 소자의 조합