Guardian LPEによるウェル近接効果およびSTIストレス効果パラメータ抽出

はじめに

90nmテクノロジ・プロセス以降、ウェル近接効果およびSTI (shallow trench isolation)ストレス効果がMOSデバイス特性の変化に大きな影響を与えるようになりました。ポスト・レイアウトの正確なSPICEシミュレーションは、これらの効果を考慮して実行する必要があります。BSIM4など新しいSPICEデバイス・モデルには、ウェル近接およびSTIストレス効果のシミュレーション用のパラメータがあります。

シルバコのGuardianLPEツールは、ポスト・レイアウトのSPICEシミュレーションに使用可能なウェル近接およびSTIストレス効果パラメータを計算するための特別な関数を提供します。この目的のためにLISAスクリプトでジェネリック・デバイス用に使用できる関数が2つあります。device_enclosure_vectorとdevice_enclosureです。

1つ目の関数は、2つ目の関数よりもシンプルですが、実行速度が速くなります。関数は、ユーザの特定のニーズに応じて選択することができます。

EnclosureVector

ウェル近接およびストレス効果パラメータは、enclosure vectorを使用して計算できます。Guardian LPEツールには、enclosure vectorの計算を可能にする一対のLISA関数があります。シンタックスは次のとおりです。

1. enclosure_vector=device_enclosure_vector(<layer_name>, max_distance)
2. device_set_enclosure_property(enclosure_vector)

1つ目の関数(device_enclosure_vector)は、デバイスのシード図形に対して指定された測定レイヤ<layer_name>のenclosure vectorを計算します。デバイス・シード図形は、次の条件を満たす必要があります。

・矩形である。

・測定レイヤのポリゴンと完全に重複している。

・シード図形の両側が測定レイヤの辺に外接している。

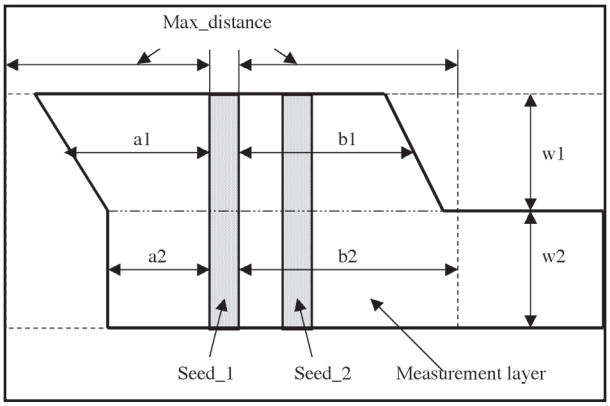

図1：デバイス図形Seed_1のEnclosure Vector。測定レイヤは2つの台形に分割されるので、enclosure vectorは2つのトリプレット (a1、b1、w1)と(a2、b2、w2)を持ちます。b2はmax_distanceで定義します。デバイス図形Seed_2は別のenclosure vectorを持ちます。

測定レイヤのポリゴンは、レイヤがデバイス・シード図形に外接した辺と平行な辺を持つ台形に分割されます。各台形はenclosure vector要素として3つの値 a、b、wを持ちます。max_distanceパラメータは、検索領域の大きさを定義し、aとbの最大値を定義します。

enclosure vectorの要素数は台形の数に対応し、enclosure vectorのSIZEフィールドにより取得できます。

前述のとおり、enclosure vectorはa、b、wの3つのフィールドを持ち、enclosure_vector[i].a、enclosure_vector[i].b、enclosure_vector[i].wによりこれらのフィールドにアクセスできます。ここで、iは要素インデックスです。

測定したレイヤが複数のデバイス・シード図形を含む場合、enclosure vectorはシードごとに計算されます。enclosure vectorの計算中、その他のすべてのデバイス・シード図形は無視されます。デバイス・シード図形が必要条件を満たさない場合、a、b、wの値は-1に設定されます。

2つ目の関数(device_set_enclosure_property)は、enclosure vectorをネットリストに出力するために使用します。この関数のフォーマットを次に示します。

sa1=... sb1=... sw1=... [sa2=... sb2=... sw2=... [...]]

例

次の例は、ユーザ定義のLISAでのenclosure vector関数の使用方法を示しています。

```
define procedure NGATE
do begin
  W = 0.0;
  L = 0.0;
  PD = 0.0;
  PS = 0.0;
  AS = 0.0;
  AD = 0.0;

  AREA = device_area("ngate", "");
  W1 = device_perimeter(REL_BUTTING,
"ngate", "S");
  W2 = device_perimeter(REL_BUTTING,
"ngate", "D");
  W = (W1 + W2) / 2;
  IF (W NEQ 0.0) THEN (L = AREA / W);

  AD = device_area("D", "");
  AS = device_area("S", "");
  PD = device_perimeter(REL_NONE, "D",
"");
  PS = device_perimeter(REL_NONE, "S",
"");

!!! device_enclosure_vector !!!
ENCL_VEC = device_enclosure_vector("act",
1000.0);
!!!

  device_set_property("L", L);
  device_set_property("W", W);
  device_set_property("PD", PD);
  device_set_property("PS", PS);
  device_set_property("AD", AD);
  device_set_property("AS", AS);

!!! device_set_enclosure_property !!!
device_set_enclosure_property(ENCL_VEC);
!!!

end;
```

ネットリストの出力例は次のとおりです。

```
MI2 d g 90 6 NMOS sb2=2.3e-006
+sa2=1.2e-005 sw2=2.1e-006 sb1=2.3e-006
+sa1=1.5e-005 sw1=6e-006 AS=72.09P
+AD=18.63P PS=34U PD=20.8U W=8.1U L=3.8U
MI1 90 5 s 6 NMOS  sb2=1.5e-005
+sa2=9e-007 sw2=2.1e-006 sb1=1.5e-005
+sa1=3.9e-006 sw1=6e-006 AS=25.29P
+AD=72.09P PS=24U PD=34U W=8.1U L=2.2U
```

ウェル近接 Enclosure 関数

この関数はウェル近接およびSTIの計算に使用できます。Guardian LPEに対応したLISA関数のシンタックスは、次のとおりです。

```
enclosure_vector = device_enclosure (direction,
              <base_layer_or_pin>,
              <meas_layer_or_pin>,
              <orient_layer_or_pin>,
              max_distance)
```

ここで、enclosure_vector はトリプレット(a、b、w)のベクタです。directionは、ベース・レイヤ<base_layer_or_pin>とオリエンテーション・レイヤ<orient_layer_or_pin>の一致した辺を基準にした測定の向きです。directionは、2つの値ORIEN_PERPENDICULARとORIEN_PARALLELを持ちます。<meas_layer_or_pin>は、ベース・レイヤを囲む測定レイヤの名前です。詳細については、図2と図3を参照してください。

次に関数入力引数の条件を説明します。

ベース・レイヤは、デバイス・レイヤ、任意のピン・レイヤ/補助レイヤ、あるいはピン名になります。ベース・レイヤの図形は矩形であり、完全に測定レイヤのポリゴンと重複する必要があります。

測定レイヤは、任意のピン・レイヤ/補助レイヤ、あるいはピン名になります。測定レイヤのポリゴンは、完全にベース・レイヤの矩形と重複する必要があります。

オリエンテーション・レイヤは、任意のピン・レイヤ/補助レイヤ、あるいはピン名になります。オリエンテーション・レイヤからのポリゴンはベース・レイヤの図形と一致した辺を少なくとも1つ持たなければなりません。一致した辺が複数ある場合、すべての辺が互いに平行である必要があります。

ベース・レイヤ、測定レイヤ、オリエンテーション・レイヤは、すべて異なるレイヤである必要があります。

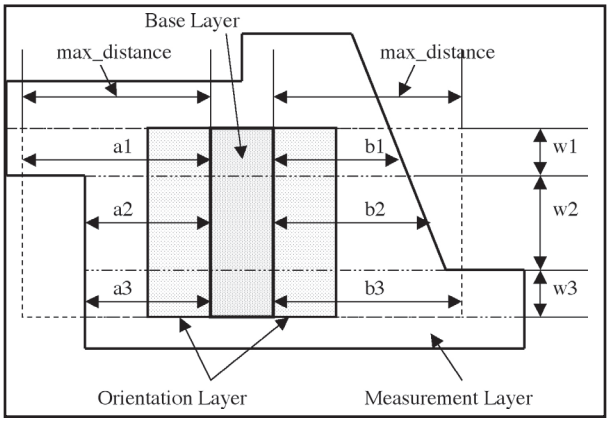

図2：ORIEN_PERPENDICULAR引数を持つenclosure関数。測定レイヤの重複部分は3つの台形に分割され、出力enclosure vectorは3つのトリプレット(a1、b1、w1)、(a2、b2、w2)、(a3、b3、w3)を持ちます。a1とb3はmax_distanceによって定義されます。

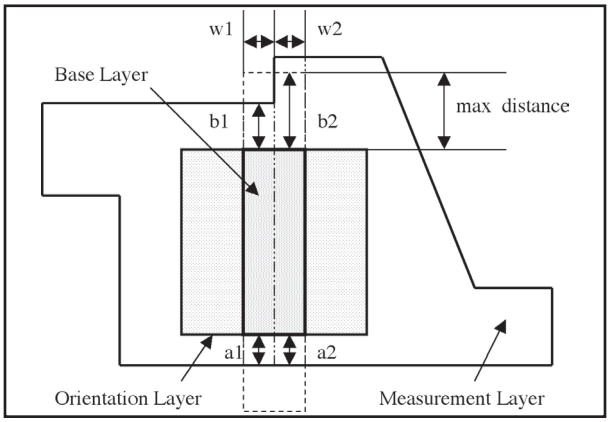

図3：ORIEN_PARALLEL引数を持つenclosure関数。測定レイヤのポリゴンの重複部分は2つの台形に分割され、出力enclosure vectorは2つのトリプレット(a1、b1、w1)と(a2、b2、w2)を持ちます。b2はmax_distanceによって定義されます。

測定レイヤのポリゴンは台形に分割されます。その台形の辺は、ORIEN_PARALLEL 方向引数が指定された場合、ベース・レイヤの矩形とオリエンテーション・レイヤのポリゴンが一致した辺と平行になります。また、ORIEN_PERPENDICULAR 方向引数が指定された場合、一致した辺に垂直になります。各台形には、enclosure vector要素としてa、b、wの3つの値があります（図2、図3参照）。max_distanceパラメータは、検索領域の大きさを定義し、つまり、a、bの最大値を定義します。

もう１つの関数(device_set_named_enclosuse_property)は、専用の名前を用いてenclosure vectorをネットリストに出力するために使用します。

```
device_set_named_enclosure_property(<name>,
enclosure_vector)
```

ここで、<name>は文字列変数です。

出力のフォーマットを次に示します。

```
<name>a1=...<name>b1=...<name>w1=...[<name>
a2=...<name>b2=...<name>w2=...[..]]
```

例

次の例は、ユーザ定義のLISAでのenclosure関数の使用方法を示しています。

```
define procedure PGATE
do begin
  W = 0.0;
  L = 0.0;
  PD = 0.0;
  PS = 0.0;
  AS = 0.0;
  AD = 0.0;
  AREA = device_area("pgate", "");
  W1 = device_perimeter(REL_BUTTING,
"pgate", "S");
  W2 = device_perimeter(REL_BUTTING,
"pgate", "D");
!  W = (W1 + W2) / 2;
  IF (W NEQ 0.0) THEN (L = AREA / W);

  AD = device_area("D", "");
  AS = device_area("S", "");
  PD = device_perimeter(REL_NONE, "D",
"");
  PS = device_perimeter(REL_NONE, "S",
"");

  c = device_count("D");
  net_name = device_pin_net("S");

  device_set_property("L", L);
  device_set_property("W", W);
  device_set_property("PD", PD);
  device_set_property("PS", PS);
  device_set_property("AD", AD);
  device_set_property("AS", AS);

! PERP and PARA are enclosure vectors
  PERP = device_enclosure(ORIEN_PERPEN-
DICULAR, "pgate", "nwell", "S", 1000.0);
  PARA = device_enclosure(ORIEN_PARALLEL,
"pgate", "nwell", "S", 1000.0);

  N_PERP = PROX_PERP.SIZE;
  E_PERP = 0.0;
  I = 0;
```

```
  LOOP
   BEGIN
     I = I + 1;
     E_PERP = E_PERP + (PROX_PERP[I].a +
PROX_PERP[I].b) * PROX_PERP[I].w;
     IF (I EQL N_PERP) THEN (LEAVE LOOP);
   END;
   E_PERP = E_PERP / L / 2;

  N_PARA = PROX_PARA.SIZE;
  E_PARA = 0.0;
  I = 0;
  LOOP
   BEGIN
     I = I + 1;
     E_PARA = E_PARA + (PROX_PARA[I].a +
PROX_PARA[I].b) * PROX_PARA[I].w;
     IF (I EQL N_PARA) THEN (LEAVE LOOP);
   END;
   E_PARA = E_PARA / W / 2;

  device_set_named_enclosure_
property("PERP", PERP);
  device_set_named_enclosure_
property("PARA", PARA);
  device_set_property("E_PERP", E_PERP);
  device_set_property("E_PARA", E_PARA);

end;
```

ネットリスト出力例は次のとおりです。

```
MI1 2 G 1 NW PMOS  E_PARA=9.61832e-007
+E_PERP=2.98389e-005 PARAb1=3e-006
+PARAa1=4e-006 PARAw1=3.6e 006
+PERPb1=8.4e-006 PERPa1=8e-006
PERPw1=1.31e-005
+sb1=5.5e-006 sa1=4.1e-006 sw1=1.31e-005
+AS=53.71P AD=72.05P PS=34.4U PD=37.2U
+W=13.1U L=3.6U
```

まとめ

本稿では、Guardian LPEでのウェル近接およびSTIストレス効果パラメータを計算する新しい関数を紹介しました。これらの機能により、Guardian LPEを最新（90nm以降）のCMOSテクノロジ・プロセスにおけるデバイス・レイアウト・パラメータ抽出に使用することができます。