New Enhanced Possibilities of Netlist Comparison in Guardian LVS

1. Introduction

The Guardian LVS (Layout versus Schematic) netlist comparison tool compares two SPICE netlists. One of the compared netlists usually corresponds to the schematic of a circuit, which represents the logic of the circuit. The other netlist represents the actual physical layout of the circuit. Often it is necessary to perform schematic versus schematic (SVS) comparison for two variants of the same circuit to check the differences in electrical connectivity. A netlist corresponding to a schematic describes the circuit in terms of the usual SPICE elements, such as transistors, diodes, resistors, capacitors, inductors, as well as specific SPICE elements, such as independent sources, voltage and current controlled sources, transmission lines, switches, coupled mutual inductors, multi-terminal networks, and subcircuit-device elements. Most known netlist comparators can process only the usual SPICE elements. Guardian LVS can handle both usual and specific SPICE elements. This application note describes the types of specific SPICE elements handled by Guardian LVS and some advantages of using the elements.

2. Overview of Specific SPICE Elements Supported by Guardian LVS

Guardian LVS can operate with specific SPICE elements, such as independent sources, voltage and current controlled sources, transmission lines, switches, coupled mutual inductors, multi-terminal networks, subcircuit-device elements. The “Models” panel of the “Project Settings” dialog box shows the name list and the options which can be set for these elements (Figure 1). Statement description of specific SPICE elements can be found in Guardian LVS Verification User’s Manual. A description of some elements is shown below:

- **Coupled Mutual Inductor**

  Syntax

  `Kxxx Laaa Lbbb <Lccc…> kvalue <mname>`

  `Kxxx`: Mutual inductor name. The name must begin with the letter `K`.

  `Laaa, Lbbb, Lccc, ...:` The names of two or more coupled inductors.

  `kvalue`: The mutual coupling coefficient, which is a unitless number. The permitted range of values for kvalue is between -1.0 and +1.0. All other values are also accepted by Guardian LVS, but a warning message is reported in the Message file. The default is 0.

  `mname`: Magnetic core model name. Indicates if the specified magnetic core model should be used instead of the simple coupled inductor model. The model statement syntax is as follows:

  `.MODEL mname CORE`

  Example

  In the following example, the mutual inductor `K1` defines the coupling between inductors `L1` and `L2`, with a coupling coefficient of 0.3.

```
L1 1 0 0.5m
L2 2 0 0.6m
K1 L1 L2 0.3
```

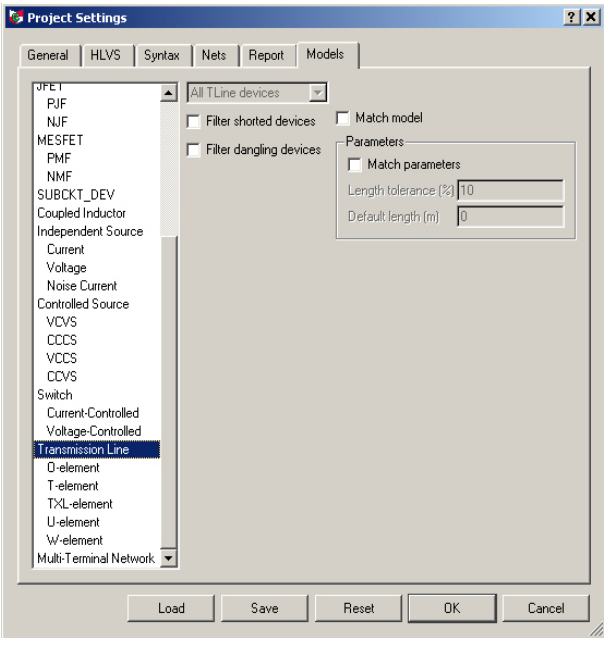

Figure 1. Specific SPICE elements.

- **Independent Current Source Element**

  Syntax

  ```
  Ixxx n+ n- <<DC> dcval> <M=val>
  ```

  Ixxx: Independent current source name. The name must begin with the letter I.

  n+, n-: Positive and negative terminal node names, respectively.

  DC: Keyword for the DC source value.

  dcval: value of the DC current source in amperes. Default is 0.

  M: Multiplier representing the number of parallel current sources. Default is 1.

  Example

  The example describes the current source between nodes 1 and 0 with a DC current value of 1mA.

  ```
  I1 1 0 DC 1mA
  ```

- **Independent Voltage Source Element**

  Syntax

  Vxxx n+ n- <<DC> dcval>

  Vxxx: Independent voltage source name. The name must begin with the letter V.

  n+, n-: Positive and negative terminal node names, respectively.

  DC: Keyword for the DC source value.

  dcval: value of the DC voltage source. Default is 0.

  Example

  The example describes a voltage source connected between nodes 1 and 0 with a DC value of 5V.

  V1 1 0 DC 5

Guardian LVS supports four types of voltage and current controlled elements: E, F, G, and H elements. These controlled elements are used to model MOS and bipolar transistors, tunnel diodes, analog functions such as operational amplifiers, comparators, summers, voltage controlled oscillators, modulators, and switched capacitor circuits. The controlled elements are either linear or nonlinear transfer functions of controlling node voltages or currents, depending on whether the polynomial or piecewise linear functions are used. Guardian LVS doesn't support the parameters of voltage and current controlled elements, but recognizes different types of transfer functions.

- **Voltage-Controlled Voltage Source**

  Syntax

  Linear

  Exxx n+ n- <VCVS> nc1+ nc1-

  Polynomial

  Exxx n+ n- <VCVS> POLY(ndim) nc1+ nc1- nc2+ nc2- ... ncdim+ ncdim-

Piecewise Linear

Exxx n+ n- <VCVS> PWL(1) | PPWL(1) | NPWL(1) nc1+ nc1-

Multi-Input Gates

Exxx n+ n- <VCVS> gatetype(k) nc1+ nc1- nc2+ nc2- ... nck+ nck-

Delay Element

Exxx n+ n- <VCVS> DELAY nc1+ nc1-

Behavior

Exxx n+ n- VOL | VALUE='expression'

Table

Exxx n+ n- TABLE

Ideal Op-Amp

Exxx n+ n- OPAMP nc1+ nc1-

Transformer

Exxx n+ n- TRANSFORMER nc1+ nc1-

Voltage Gate H(s) (LAPLACE)

Exxx n+ n- LAPLACE nc1+ nc1-

Voltage Gate H(s) (POLE)

Exxx n+ n- POLE nc1+ nc1-

Voltage Gate H(s) (FREQ)

Exxx n+ n- FREQ nc1+ nc1-

Voltage Gain H(z) (ZTRANS)

Exxx n+ n- ZTRANS nc1+ nc1-

Exxx: Voltage-controlled voltage source name. The name must begin with the letter E.

n+, n-: Positive and negative terminal node names of the controlled source.

VCVS: Keyword for voltage controlled voltage source. It should not be used as a node name.

nc1+, nc1-...: Positive and negative controlling node names, in pairs, one pair for each dimension in polynomial form or one pair for each gate in multi-input gate form.

POLY: Polynomial function parameter. It should not be used as a node name.

ndim: The number of dimensions of the polynomial.

PWL (or **NWL**): Piecewise linear function parameter. It should not be used as a node name.

gatetype(k): AND, NAND, OR and NOR parameters set the type of multi-input gate device. The value of k is the number of inputs. AND, NAND, OR and NOR should not be used as node names.

DELAY: Delay element parameter. It should not be used as a node name.

VOL | VALUE: Keyword used to indicate that the output voltage is defined by expression.

TABLE: Parameter used to indicate that the output voltage will be found by using a lookup table. It should not be used as a node name.

OPAMP: Ideal op-amp element parameter. It should not be used as a node name.

TRANSFORMER: Ideal transformer parameter. It should not be used as a node name.

LAPLACE: Keyword indicates that the transfer function is described by a Laplace transform function in the form of a rational function. It should not be used as a node name.

POLE: Keyword indicates that the transfer function is described by the location of the poles and zeroes. It should not be used as a node name.

FREQ: Keyword indicates that the transfer function is described by a frequency-response table. It should not be used as a node name.

ZTRANS: Keyword indicates that the transfer function is described by a Z-transform function in the form of a rational function. It should not be used as a node name.

Examples

1. The device EON describes a linear VCVS connected between nodes 3 and 9, with controlling nodes 8and 6:

```
EON 3 9 8 6
```

2. Linear VCVS with terminals n1 n2, with controlling nodes 1 and 2:

```
EOUT n1 n2 POLY(1) 1 2
```

3. Nonlinear VCVS whose voltage is a polynomial function of two voltages v(3,0) and v(4,1):

```
Ea va gnd POLY(2) 3 0 4 1
```

4. Nonlinear VCVS whose voltage is given by the expression that follows the parameter VOL:

```
Eb 13 0 VOL='10.*log(i(vin)/10m)'
```

5. VCVS with the transfer function described by a Laplace transform function:

```
E1 out grnd LAPLACE in grnd
```

Guardian LVS supports current-controlled and voltage-controlled switches. Parameter comparison is not supported for switch elements.

- **Current-Controlled Switch**

Syntax

Wxxx n+ n- vcontrolname mname

Wxxx: Current-controlled switch element name. The name must begin with the letter W.

n+, n-: Positive and negative terminal node names.

vcontrolname: Name of the voltage source of the controlling current flow.

mname: Model name. The model name must reference a current-controlled switch model. The model statement syntax is as follows:

.MODEL mname CSW

Example

```
.MODEL WCMOD CSW
W1 n1 n2 VIN WCMOD
```

The example specifies the switch W1 placed between nodes n1 and n2. The state of the switch is controlled by the voltage source VIN. The switch is defined by WCMOD model.

Guardian LVS supports five types of transmission lines: Lossy (O-element), Lossless (T-element), Lossy (TXL-element), Coupled Lossy (U-element), Coupled Lossy (W-element) transmission lines.

- **Lossy (O-element) Transmission Line**

Syntax

Oxxx n1 n2 n3 n4 mname <L | LEN | LENGTH=val> <M=val>

Oxxx: The lossy uniform transmission line element name. The name must begin with the letter O.

n1, n2: The nodes at port 1.

n3, n4: The nodes at port 2.

mname: Model reference name.

L (or LEN, LENGTH): Physical length of the transmission line (meters). Default is L=1m.

M: Multiplier used to represent parallel devices. Default is M=1.

Every model of the lossy transmission line must have the model statement. The model statement syntax is as follows:

.MODEL mname LTRA

Example

```
.MODEL tlline LTRA
.MODEL connect LTRA
O1 1 0 2 0 tlline l=.6095
ONET 1 2 3 2 connect length=.16
```

3. Subcircuit-device and Calculation of it Parameters Using LISA

Subcircuit-device is an analogue of subcircuit and can be used instead of any SPICE element. Subcircuit-device instances are represented in a netlist by subcircuit instances and can be interpreted as both simple devices and subcircuit instances. The *.SUBCKT_DEV command is used to specify the subcircuit-device type for a subcircuit name:

***.SUBCKT_DEV sname stype nPins [(Pin_i,..,Pin_j) …]**

- `sname` is a subcircuit name. The subcircuit with this name may or may not be defined in the netlist.
- `nPins` is pin number of the subcircuit-device.
- `(Pin_i,…,Pin_j)…` is an optional parameter which specifies the groups of pin indexes which are swappable. The `*.SUBCKT_DEV` statement can contain several groups of swappable pins.
- `stype` is subcircuit-device type, and can be one of the following:
  - **M:** MOS transistor
  - **Q:** Bipolar transistor
  - **J:** JFET transistor
  - **B or Z:** MESFET transistor
  - **Y:** User-defined element
  - **X:** Subcircuit
  - **R:** Resistor
  - **C:** Capacitor
  - **L:** Inductor
  - **D:** Diode
  - **NMOS:** N-channel MOSFET
  - **PMOS:** P-channel MOSFET
  - **NPN:** NPN BJT
  - **PNP**: PNP BJT
  - **NJF:** N-channel JFET
  - **PJF:** P-channel JFET
  - **NMF:** N-channel MESFET
  - **PMF:** P-channel MESFET
  - **CORE:** Coupled inductor
  - **I:** Independent current source
  - **V:** Independent voltage source
  - **N:** Independent noise current source
  - **VCVS**: Voltage-controlled voltage source
  - **CCCS:** Current-controlled current source
  - **VCCS**: Voltage-controlled current source
  - **CCVS:** Current-controlled voltage source
  - **CSW**: Current-controlled switch
  - **SW:** Voltage-controlled switch
  - **LTRA:** Lossy (O-element) transmission line
  - **TXL:** Lossy (TXL-element) transmission line
  - **T:** Lossless (T-element) transmission line
  - **U:** Coupled lossy (U-element) transmission line
  - **W:** Coupled lossy (W-element) transmission line
  - **SP:** Multi-terminal network
  - **VLG:** User-defined Verilog-A element

Depending on the subcircuit-device type you specify, the subcircuit-device instances will be interpreted either as subcircuit instances without looking inside the subcircuit definition, or simple devices of a specified type. If the subcircuit-device type is `X`, all instances of the subcircuit `sname` are interpreted as subcircuit instances. If the subcircuit-device type is `M, Q, J, B` or `Z`, the instances of the subcircuit sname can be processed either as subcircuit-device instances or simple devices. For all other types, the instances of the subcircuit sname are simple devices of corresponding type. To recognize the instances of subcircuits with subcircuit-device type `M, Q, J, B` or `Z` in `*.SUBCKT_DEV` statement as simple devices, the following requirements should be satisfied:

- A model name, coded as comment, should be specified in parameter section of subcircuit instance statement.
- A device type should be specified for the model name by a `.MODEL` statement.
- A `*.SUBCKT_DEV` statement should be defined for this subcircuit with one of the subcircuit-device types `M, Q, J, B` or `Z` which correspondents to the device type (for example, M correspondents to `PMOS` or `NMOS`).

Examples

1. The instances `X0` and `X1` of subcircuit `PCH` are recognized as `PMOS` transistors with model names `PCH1` and `PCH2`, accordingly.

```
.MODEL PCH1 PMOS
.MODEL PCH2 PMOS
*.SUBCKT_DEV PCH M 4 (1,3)
X0 1 2 3 4 PCH L=1u W=1u $[PCH1]
X1 3 2 6 4 PCH L=1u W=1u $[PCH2]
```

2. The instances X0 and X1 are interpreted as subcircuit-devices.

```
*.SUBCKT_DEV PCH1 M 4 (1,3)
*.SUBCKT_DEV PCH2 X 3
X0 1 2 3 4 PCH1 L=1u W=1u
X1 a b c PCH2 L=1u W=1u
```

3. The instance X0 is PMOS transistor with model name PCH, and X1 is resistor with model name RMP.

```
*.SUBCKT_DEV PCH PMOS 4 (1,3)
*.SUBCKT_DEV RMP R 2
X0 1 2 3 4 PCH L=1u W=1u
X1 a b RMP R=1000
```

Guardian LVS allows the user to compare the parameters of subcircuit-devices. By default, all parameters will be compared, if the user chooses the “Match parameters” option for subcircuit-devices in the “Models” panel of the “Project Settings” dialog box. Often the subcircuit-device elements contain a lot of parameters, but the user wants to compare some of them. In this case, script procedures can be used to determine which parameters of subcircuit-devices will be compared and to specify parameter computation algorithms for any complex elements obtained as result of parallel merging of simple subcircuit-devices. The procedures should be implemented using the LISA script language. The complete description of LISA may be found in the document “Language for Interfacing Silvaco Applications (LISA) User’s Manual”.

All script procedures are placed in one script file that should be specified in the “LISA script file” field in the “General” panel of the “Project Settings” dialog box. The user can determine a separate LISA procedure for any type of subcircuit-device (Figure 2). At the same time, a common LISA procedure can be used for the subcircuit-devices which are not specified in the popup combo box of the “Models” panel. If at least one LISA procedure is used, a LISA script file containing the procedure must be specified in the “General” settings panel. To handle the subcircuit-device parameters of a SPICE netlist, Guardian LVS Interface procedures are used at the time of implementation of LISA script procedures:

GetParameterValues

The procedure returns the sequence of parameter values of the subcircuit-devices merged in parallel.

Syntax

param_values = GetParameterValues(param_name);

Parameters

param_name (type STRING) – parameter name.

Return Value

param_values (type SEQUENCE) – sequence of parameter values.

SetParameterValue

The procedure sets the calculated value for specified parameter.

Syntax

SetParameterValue(param_name, param_value);

Parameters

param_name (type STRING) – parameter name.

param_value (type FLOAT) – value of calculated parameter.

The following is an example of using the LISA script procedures for subcircuit-device parameter comparison. Compared netlists contain the subcircuit-devices with the names PCH and NCH. The LISA script file includes two procedures: LW_CALC_PCH for subcircuit-devices with the name PCH, and LW_CALC_NCH for subcircuit-devices with the name NCH. Each netlist element has four parameters L, W, A and B. But only some of them are compared: L, W, A for PCH-elements, and L, W, B for NCH-elements. The parameter match report contains the results of calculations and comparison of subcircuit-device parameters.

Example

First netlist:

```
*.SUBCKT_DEV PCH M 4 (1,3)
*.SUBCKT_DEV NCH M 4 (1,3)

X0 n1 IN n2 VDD  PCH L=3u  W=2u A=3
B=20
X1 n1 IN n2 VDD  PCH L=3u  W=2u A=6
B=50
X2 n1 IN n2 VDD  PCH L=3u  W=2u A=3
B=80
X3 net1 IN2 net2 GND AAA
X4 net1 IN2 net2 GND AAA

.SUBCKT AAA d g s b
X0 d g s b  NCH  L=4u W=10.5u A=3 B=30
.ENDS
.END
```

Second netlist:

```
X0 n1 IN n2 VDD  PCH L=3u W=6u A=6 B=30
```

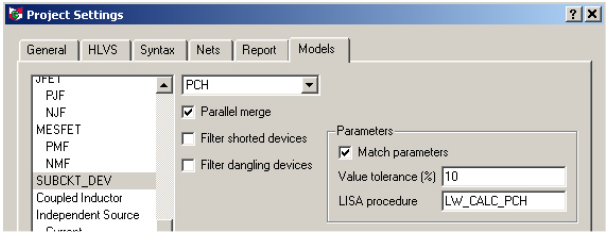

Figure 2. Specifying LISA procedure for subcircuit-device parameters.

```
X1 net1 IN2 net2 GND  NCH L=4u  W=6u
A=3 B=60
X2 net1 IN2 net2 GND  NCH  L=4u  W=7u
A=3 B=90
X3 net1 IN2 net2 GND  NCH  L=4u  W=8u
A=3 B=60
.END
```

Parameter match report:

```
         PARAMETER MATCHES
L1: X:(@)top:X3:X0 ......... = ($)  top:X3
          B=0.866025      B=0.866025
          L=4E-006        L=4E-006
          W=2.1E-005     W=2.1E-005

L0: X:($)top:X2 ............ = top:X0
          A=6             A=6
          L=3E-006         L=3E-006
          W=6E-006       W=6E-006
(@) - indicates device instances formed
by device reduction, see merge file
($) - indicates device instances formed
by device merging, see merge file
```

LISA script file:

```
DEFINE PROCEDURE /REPLACE LW_CALC_PCH
  DO
    BEGIN
      L = GetParameterValues(“L”);
      W = GetParameterValues(“W”);
      A = GetParameterValues(“A”);
      N = L.SIZE;
      I = 0;
      P = 0.0;
      Q = 0.0;
      MAXA = 0.0;
      LOOP
        BEGIN
          I = I + 1;
          P = P + L[I] * W[I];
          Q = Q + W[I] / L[I];
          IF (A[I] GTR MAXA) THEN
(MAXA = A[I]);
          IF (I EQL N) THEN (LEAVE
LOOP);
        END;

      LRES = SQRT(P / Q);
      WRES = SQRT(P * Q);
      SetParameterValue(“L”, LRES);
      SetParameterValue(“W”, WRES);
      SetParameterValue(“A”, MAXA);
    END;

DEFINE PROCEDURE /REPLACE LW_CALC_NCH
  DO
    BEGIN
      L = GetParameterValues(“L”);
      W = GetParameterValues(“W”);
      B = GetParameterValues(“B”);
      N = L.SIZE;
      I = 0;
      P = 0.0;
      Q = 0.0;
      BVAL = 0.0;
      LOOP
        BEGIN
          I = I + 1;
          P = P + L[I] * W[I];
          Q = Q + W[I] / L[I];
          BVAL = BVAL + SIN(B[I]) *
COS(B[I]);
          IF (I EQL N) THEN (LEAVE
LOOP);
        END;

      LRES = SQRT(P / Q);
      WRES = SQRT(P * Q);
      SetParameterValue(“B”, BVAL);
      SetParameterValue(“L”, LRES);
      SetParameterValue(“W”, WRES)
    END;
```

4. Conclusion

Usage of specific SPICE elements in SVS verification makes schematic design more efficient. Subcircuit-device elements are often useful in representing generic devices that can be compared to usual devices such as transistors, diodes, and resistors. LISA script procedures give flexibility in calculation and comparison of subcircuit-device parameters.