

Guardian LVS의 넷리스트 비교 기능 향상

1. 소개

Guardian LVS (레이아웃 대 스키매틱) 넷리스트 비교 툴은 두 개의 SPICE 넷리스트를 비교합니다. 대상 넷리스트 중 하나는 일반적으로 회로의 로직을 나타내는 회로도에 해당합니다. 다른 넷리스트는 회로의 실제의 물리적인 레이아웃을 나타냅니다. 전기적 연결의 차이점을 검사하기 위해, 동일 회로의 두 유형에 대해 스키매틱 대 스키매틱 (SVS) 비교를 하는 경우가 종종 있습니다. 스키매틱에 해당하는 넷리스트는 트랜지스터, 다이오드, 저항, 캐패시터, 인덕터와 같은 일반적인 SPICE 요소뿐만 아니라 독립원, 전압 및 전류 제어원, 전송선, 스위치, 상호 결합 인덕터, 다중 단자 네트워크, 부회로-소자 요소 등의 특정 SPICE 요소를 기준으로 회로를 서술합니다. 대부분의 알려진 넷리스트 비교 툴은 일반적인 SPICE 요소만 처리할 수 있습니다. Guardian LVS는 일반적인 SPICE 요소와 특정 SPICE 요소를 모두 처리할 수 있습니다. 본 애플리케이션 노트는 Guardian LVS에서 처리하는 특정 SPICE 요소의 유형과 해당 요소를 사용할 경우의 이점을 설명합니다.

2. Guardian LVS에서 지원하는 특정 SPICE 요소의 개요

Guardian LVS는 독립원, 전압 및 전류 제어원, 전송선, 스위치, 상호 결합 인덕터, 다중 단자 네트워크, 부회로-소자 요소와 같은 특정 SPICE 요소와 함께 동작할 수 있습니다. "Project Settings" 대화 상자의 "Models" 패널에서 이름의 목록과 설정할 수 있는 옵션을 나타냅니다 (그림 1). 특정 SPICE 요소에 대한 구문 설명은 "Guardian LVS Verification User's Manual" 을 참조하십시오. 일부 요소에 대해 아래에서 설명합니다:

- **Coupled Mutual Inductor**

구문

Kxxx Laaa Lbbb <Lccc...> kvalue <mname>

Kxxx: 상호 인덕터 이름으로서, 영문자 K로 시작합니다.

Laaa, Lbbb, Lccc, ...: 두 개 이상의 상호 인덕터 이름입니다.

kvalue: 상호 결합 계수로서 단위가 없습니다. kvalue에 허용되는 값은 -1.0~+1.0입니다. Guardian LVS에서 다른 값도 허용되지만, 메시지 파일에 경고 메시지가 표시됩니다. 기본값은 0입니다.

mname: 자기 코어 모델 이름입니다. 단순 결합 인덕터 모델 대신에 특정 자기 코어 모델을 사용하는지 여부를 나타냅니다. 구문은 다음과 같습니다.

.MODEL mname CORE

예제

다음 예제에서 상호 인덕터 K1은 결합 계수 0.3으로, 인덕터 L1과 L2 사이의 커플링을 정의합니다.

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

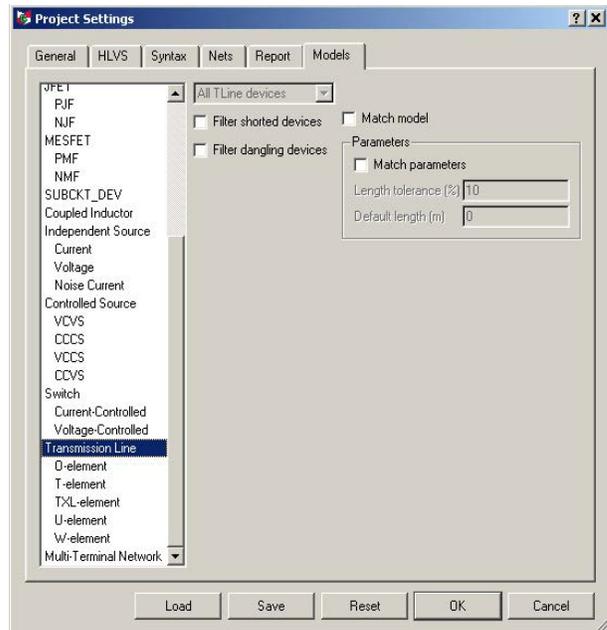


그림 1. 지정 SPICE 요소

• Independent Current Source Element

구문

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

Ixxx: 독립 전류원 이름으로서, 영문자 I로 시작합니다.

n+, **n-**: 양극 및 음극 단자 노드 이름입니다.

DC: DC 소스 값에 대한 키워드입니다.

dcval: DC 전류원의 값(암페어)으로서, 기본값은 0입니다.

M: 병렬 전류원의 갯수를 나타내는 승수로서, 기본값은 1입니다.

예제

DC 전류 1mA를 사용하여, 노드 1과 0 사이의 전류원을 서술합니다.

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

• Independent Voltage Source Element

구문

`Vxxx n+ n- <<DC> dcval>`

Vxxx: 독립 전압원 이름으로서, 영문자 V로 시작합니다.

n+, **n-**: 양극 및 음극 단자 노드 이름입니다.

DC: DC 소스 값에 대한 키워드입니다.

dcval: DC 전압원의 값으로서, 기본값은 0입니다.

예제

DC 5V 값을 사용하여, 노드 1과 0 사이에 연결된 전압원을 서술합니다.

```
V1 1 0 DC 5
```

Guardian LVS는 E, F, G, H 요소의 4가지 전압 및 전류 제어 요소를 지원합니다. 이러한 제어 요소는 MOS, 바이폴라 트랜지스터, 터널 다이오드 및 동작 증폭기, 비교기, 전압 제어 오실레이터, 변조기, 스위치 캐패시터 회로 등의 아날로그 함수를 모델링하는 데 사용됩니다. 제어 요소는 다항식 또는 부분적 선형 함수 사용 여부에 따라, 노드 전압 또는 전류를 제어하는 선형 또는 비선형 전달 함수입니다. Guardian LVS는 전압 및 전류 제어 요소의 파라미터를 지원하지 않지만, 다양한 유형의 전달 함수를 인식합니다.

• Voltage-Controlled Voltage Source

구문

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: 전압 제어 전압원 이름으로서, 영문자 E로 시작합니다.

n+, **n-**: 제어원의 양극 및 음극 단자 노드 이름입니다.

vcvs: 전압 제어 전압원에 대한 키워드로서, 노드 이름으로 사용할 수 없습니다.

nc1+, **nc1-...:** 양극 및 음극 제어 노드 쌍의 이름입니다. 한 쌍은 다항식 형태로 각각의 차원을, 다른 쌍은 다중 입력 게이트 형태로 각각의 게이트를 나타냅니다.

POLY: 다항식 함수 파라미터로서, 노드 이름으로 사용할 수 없습니다.

ndim: 다항식의 차원 수를 나타냅니다.

PWL (or NWL): 각각의 선형 함수 파라미터로서, 노드 이름으로 사용할 수 없습니다.

gatetype(k): AND, NAND, OR, NOR 파라미터로 다중 입력 게이트 소자의 유형을 설정합니다. K는 입력의 갯수를 나타내며, AND, NAND, OR, NOR는 노드 이름으로 사용할 수 없습니다.

DELAY: 지연 요소 파라미터로서, 노드 이름으로 사용할 수 없습니다.

VOL | VALUE: 출력 전압을 식으로 정의하였음을 나타내는 키워드입니다.

TABLE: 조회 테이블을 사용하여 출력 전압을 찾을 수 있음을 나타내는 파라미터입니다. 노드 이름으로 사용할 수 없습니다.

OPAMP: 이상적인 op-amp 요소 파라미터로서, 노드 이름으로 사용할 수 없습니다.

TRANSFORMER: 이상적인 변환 파라미터로서, 노드 이름으로 사용할 수 없습니다.

LAPLACE: 유리 함수 형태의 라플라스 변환 함수로 전달 함수를 서술함을 나타내는 키워드입니다. 노드 이름으로 사용할 수 없습니다.

POLE: 극과 0의 위치에 의해 전달 함수를 서술함을 나타내는 키워드입니다. 노드 이름으로 사용할 수 없습니다.

FREQ: 주파수-응답 테이블에 의해 전달 함수를 서술함을 나타내는 키워드입니다. 노드 이름으로 사용할 수 없습니다.

ZTRANS: Z-변환 함수에 의해 전달 함수를 서술함을 나타내는 키워드입니다. 노드 이름으로 사용할 수 없습니다.

예제

1. 소자 EON은 노드 8과 6을 제어하여, 노드 3과 9 사이에 연결된 선형 VCVS를 서술합니다:

```
EON 3 9 8 6
```

2. 노드 1과 2를 제어하여, 단자 n1, n2를 갖는 선형 VCVS:

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

3. 두 전압 v(3,0), v(4,1)의 다항식 함수로 전압을 나타내는 비선형 VCVS:

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

4. 파라미터 VOL 다음의 식으로 전압을 나타내는 비선형 VCVS:

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

5. 라플라스 변환 함수로 전달 함수를 나타내는 VCVS:

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

Guardian LVS는 전류 제어 및 전압 제어 스위치를 지원합니다. 스위치 요소에 대한 파라미터 비교는 지원하지 않습니다.

• Current-Controlled Switch

구문

```
Wxxxx n+ n- vcontrolname mname
```

Wxxxx: 전류 제어 스위치 요소 이름으로서, 영문자 W로 시작합니다.

n+, n-: 양극 및 음극 단자 노드 이름입니다.

vcontrolname: 제어 전류 플로우의 전압원 이름입니다.

mname: 모델 이름으로서, 전류 제어 스위치 모델을 참조합니다. 모델 구문은 다음과 같습니다:

```
.MODEL mname CSW
```

예제

```
.MODEL WCMOD CSW
```

```
W1 n1 n2 VIN WCMOD
```

예제는 노드 n1과 n2 사이의 스위치 W1을 지정합니다. 스위치의 상태는 전압원 VIN으로 제어합니다. 스위치는 WCMOD 모델로 정의합니다.

Guardian LVS는 5가지 유형의 전송선을 지원합니다: 손실 (O-element), 비손실 (T-element), 손실 (TXL-element), 결합 손실 (U-element), 결합 손실 (W-element) 전송선.

• Lossy (O-element) Transmission Line

구문

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

Oxxxx: 손실된 균일 전송선 요소 이름으로서, 영문자 O로 시작합니다.

n1, n2: 포트 1의 노드입니다.

n3, n4: 포트 2의 노드입니다.

mname: 모델 참조 이름입니다.

L (or LEN, LENGTH): 전송선의 물리적인 길이(미터)로서 기본값은 L=1m입니다.

M: 병렬 소자를 나타내는 승수로서 기본값은 M=1입니다.

손실 전송선의 모든 모델은 모델 구문을 포함합니다. 모델 구문은 다음과 같습니다:

```
.MODEL mname LTRA
```

예제

```
.MODEL tlline LTRA
```

```
.MODEL connect LTRA
```

```
O1 1 0 2 0 tlline l=.6095
```

```
ONET 1 2 3 2 connect length=.16
```

3. 부회로-소자 및 LISA를 이용한 파라미터 계산

부회로-소자는 부회로를 아날로그로 나타낸 것이며, SPICE 요소 대신 사용할 수 있습니다. 부회로-소자 인스턴스는 부회로 인스턴스에 의해 넷리스트에 표시하며, 단순 소자와 부회로 인스턴스로 해석할 수 있습니다. *.SUBCKT_DEV 명령어는 부회로 이름에 대해 부회로 소자의 유형을 지정합니다:

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

- sname은 부회로 이름입니다. 본 이름을 갖는 부회로는 넷리스트에 정의하거나 정의하지 않을 수 있습니다.
- nPins은 부회로-소자의 핀 갯수입니다.
- (Pin_i,...,Pin_j)는 대체 가능한 핀 인덱스 그룹을 지정하는 선택적 파라미터입니다. *.SUBCKT_DEV 구문에 대체 가능한 핀 그룹을 여럿 포함할 수 있습니다.
- stype는 부회로-소자의 유형으로서, 다음 중 하나입니다:
 - **M**: MOS 트랜지스터
 - **Q**: 바이폴라 트랜지스터
 - **J**: JFET 트랜지스터
 - **B or Z**: MESFET 트랜지스터
 - **Y**: 사용자 정의 요소
 - **X**: 부회로
 - **R**: 저항
 - **C**: 캐패시터
 - **L**: 인덕터
 - **D**: 다이오드
 - **NMOS**: N-채널 MOSFET
 - **PMOS**: P-채널 MOSFET
 - **NPN**: NPN BJT
 - **PNP**: PNP BJT
 - **NJF**: N-채널 JFET
 - **PJF**: P-채널 JFET
 - **NMF**: N-채널 MESFET
 - **PMF**: P-채널 MESFET
 - **CORE**: 결합 인덕터
 - **I**: 독립 전류원
 - **V**: 독립 전압원
 - **N**: 독립 노이즈 전류원
 - **VCVS**: 전압 제어 전압원
 - **CCCS**: 전류 제어 전류원
 - **VCCS**: 전압 제어 전류원
 - **CCVS**: 전류 제어 전압원

- **CSW**: 전류 제어 스위치
- **SW**: 전압 제어 스위치
- **LTRA**: 손실 (O-element) 전송선
- **TXL**: 손실 (TXL-element) 전송선
- **T**: 비손실 (T-element) 전송선
- **U**: 결합 손실 (U-element) 전송선
- **W**: 결합 손실 (W-element) 전송선
- **SP**: 다중 단자 네트워크
- **VLG**: 사용자 정의 Verilog-A 요소

지정한 부회로-소자 유형에 따라, 부회로-소자 인스턴스는 부회로 내부의 확인 없이 부회로 인스턴스 또는 지정한 유형의 단순한 소자로 해석합니다. 부회로-소자 유형이 X인 경우, 부회로 sname의 모든 인스턴스는 부회로 인스턴스로 해석합니다. 부회로-소자 유형이 M, Q, J, B 또는 Z인 경우, 부회로 sname의 인스턴스는 부회로-소자 인스턴스 또는 단순한 소자로 처리할 수 있습니다. 다른 모든 유형의 경우, 부회로 sname의 인스턴스는 해당 유형의 단순한 소자입니다. *.SUBCKT_DEV 구문에서 부회로 소자-유형 M, Q, J, B 또는 Z를 갖는 부회로 인스턴스를 단순한 소자로서 인식하려면, 다음 사항을 충족해야 합니다.

- 주석으로 코딩된 모델 이름을 부회로 인스턴스 구문의 파라미터 섹션에 지정합니다.
- 소자 유형은 .MODEL 구문으로 모델 이름에 대해 지정합니다.
- *.SUBCKT_DEV 구문은 소자 유형에 해당하는 부회로-소자 유형 M, Q, J, B 또는 Z 중에서 하나를 갖는 부회로를 정의합니다 (예: M이 PMOS 또는 NMOS에 대응).

예제

1. 부회로 PCH의 X0 및 X1 인스턴스는 PCH1 및 PCH2 모델 이름을 갖는 PMOS 트랜지스터로 인식됩니다.

```
.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. 인스턴스 X0 및 X1은 부회로-소자로 해석됩니다.

```
*.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. X0는 모델 이름 PCH인 PMOS 트랜지스터이며, X1은 모델 이름 RMP인 저항입니다.

```

*.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 LVS를 통해 부회로-소자의 파라미터를 비교할 수 있습니다. "Project Settings" 대화 상자의 "Models" 패널에서 부회로-소자에 대해 "Match parameters" 옵션을 선택하는 경우, 기본적으로 모든 파라미터를 비교합니다. 종종 부회로-소자 요소는 많은 파라미터를 포함하지만, 사용자는 그 중에서 일부를 비교하고자 합니다. 이 경우, 스크립트 프로시저를 사용하여 부회로-소자의 파라미터를 비교하고, 단순한 부회로-소자의 병렬 병합에서 얻은 복잡한 요소에 대해 파라미터 계산 알고리즘을 지정할 수 있습니다. 프로시저는 LISA 스크립트 언어를 사용하여 구현합니다. LISA에 대한 전체적인 설명은 "Language for Interfacing Silvaco Applications (LISA) User's Manual"을 참조하십시오.

모든 스크립트 프로시저는 "Project Settings" 대화 상자의 "General" 패널에 있는 "LISA script file" 항목에서 지정한 스크립트 파일에 있습니다. 사용자는 모든 유형의 부회로-소자에 대해 별도의 LISA 프로시저를 정할 수 있습니다 (그림 2). 동시에, "Models" 패널의 팝업 콤보 상자에 지정되지 않은 부회로-소자에 대해 공통적인 LISA 프로시저를 사용할 수 있습니다. LISA 프로시저를 사용하는 경우, 이를 포함하는 LISA 스크립트 파일을 "General" 설정 패널에서 지정합니다. SPICE 넷리스트의 부회로-소자 파라미터를 처리하기 위해, Guardian LVS 인터페이스 프로시저는 LISA 스크립트 프로시저를 구현할 때 사용됩니다.

GetParameterValues

프로시저는 병렬로 병합된 부회로-소자의 파라미터 값의 시퀀스를 반환합니다.

```
param_values = GetParameterValues(param_name);
```

Parameters

param_name (type STRING) – parameter name.

Return Value

param_values (type SEQUENCE) – sequence of parameter values.

SetParameterValue

프로시저는 지정 파라미터에 계산된 값을 설정합니다.

Syntax

```
SetParameterValue(param_name, param_value);
```

Parameters

param_name (type STRING) – parameter name.

param_value (type FLOAT) – value of calculated parameter.

다음은 부회로-소자 파라미터 비교를 위한 LISA 스크립트 프로시저의 예입니다. 비교한 넷리스트는 부회로-소자 PCH, NCH를 포함합니다. LISA 스크립트 파일은 두 가지 프로시저를 포함합니다: 부회로-소자 PCH에 대한 LW_CALC_PCH, 부회로-소자 NCH에 대한 LW_CALC_NCH. 각 넷리스트 요소는 L, W, A, B의 4가지 파라미터가 있습니다. 그러나 그들 중 일부만 비교합니다: PCH-요소의 경우 L, W, A, NCH-요소의 경우 L, W, B. 파라미터 일치 보고서는 부회로-소자 파라미터의 계산 및 비교 결과를 포함합니다.

예제

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
```

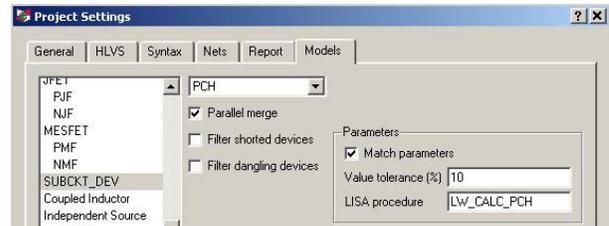


그림 2. 부회로-소자 파라미터에 대한 LISA 프로시저 지정

```
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. 결론

SVS 검증에서 특정 SPICE 요소를 사용하면, 스키매틱을 효율적으로 설계할 수 있습니다. 부회로-소자 요소는 종종 트랜지스터, 다이오드 및 저항과 같은 소자와 비교되는 일반적인 소자를 나타내는데 유용합니다. LISA 스크립트 프로시저는 부회로-소자 파라미터의 계산 및 비교를 유연하게 할 수 있습니다.