5 Timing Analysis and Signal Integrity in BPR

Introduction

You can use L-Edit to perform signal integrity and timing analysis of the IC design at any stage of the design process when using BPR. By providing input signal properties, driver/receiver models, and net characteristics, you can simulate signal performance on single or multiple nets.

Performing analyses interactively during placement and routing allows you to evaluate the impact of your chosen layout, driver and receiver parameters, and interconnect topology on signal integrity and timing constraints of a design. You can use net simulation following placement and routing for electrical and timing rules verification and interactive correction of their violations.

L-Edit/BPR provides two types of electrical analysis:

- Delay calculation (timing analysis)
- Time domain simulation (signal integrity analysis)

Timing Analysis

Delay calculation provides a fast estimation of signal timing in routed and unrouted designs. This analysis measures the amount of time the signal at the receiving pin takes to reach the voltage at a threshold you specify. The threshold is expressed as a percentage of the peak voltage.

The delay calculator models each net as a sequence of transmission lines with a characteristic resistance and capacitance, which you set in the **Timing Analysis** dialog (see Running Timing Analysis on page 2-243). It assumes the drivers and receivers are attached to the pins in a treelike structure also characterized by a single capacitance and resistance for all drivers and receivers.

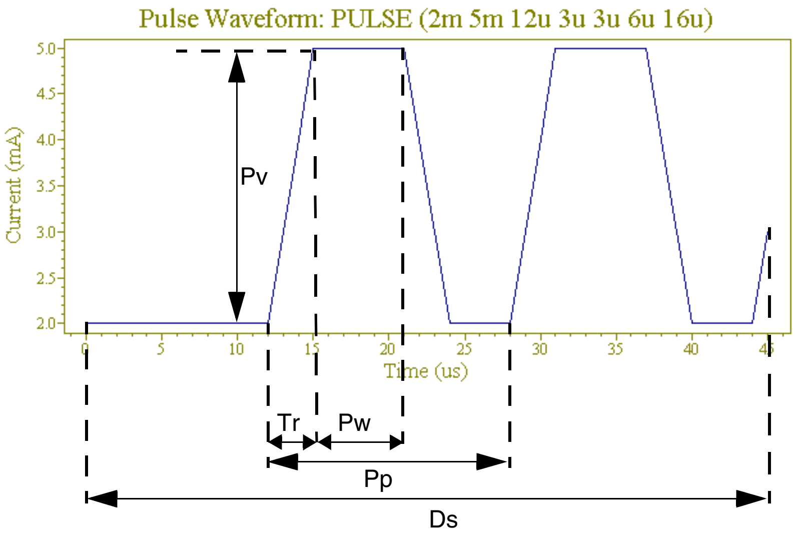
Area and fringe layer-to-substrate capacitance and resistance values for the routing layers are obtained from the **Setup Layers** dialog (see Layer Setup on page 1-155).

L-Edit reports the results of each net's delay in the **Delay Calculation Report** (see Delay Calculation Report on page 2-248).

Signal Integrity Analysis

Time domain simulation extracts a SPICE-format netlist of the selected nets for simulation with an external SPICE simulator. L-Edit generates SPICE-format descriptions of interconnect components from the design layout, components, and simulation commands to produce the complete SPICE-format netlist. Model information of interconnects, vias, drivers, and receivers is included from a user-created model file. L-Edit automatically produces the netlist and, if you have it installed, will invoke the Tanner EDA circuit simulation tool T-Spice with the netlist preloaded.

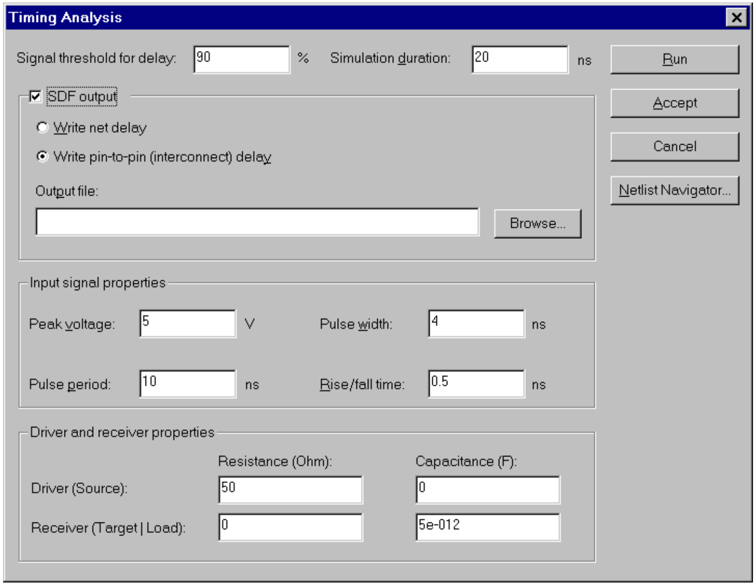
Both analyses use the following basic properties of the input signal:



Legend	*Parameter*
Pv	Peak voltage (V)
Pp	Pulse period (ns)
Pw	Pulse width (ns)
Tr	Rise/fall time (ns)
Ds	Simulation duration (ns)

Running Timing Analysis

Use **Tools > BPR > Timing Analysis** to open the **Timing Analysis** dialog. This dialog is used to enter information on the properties of the input signal and to initiate analysis.



Signal threshold for delay: 90 %

Simulation duration: 20 ns

☑ SDF output

○ Write net delay

● Write pin-to-pin (interconnect) delay

Output file:

Input signal properties

Peak voltage: 5 V

Pulse width: 4 ns

Pulse period: 10 ns

Rise/fall time: 0.5 ns

Driver and receiver properties

	Resistance (Ohm):	Capacitance (F):
Driver (Source):	50	0
Receiver (Target | Load):	0	5e-012

.

Signal threshold for delay Percentage of the **Peak voltage** at which the delay is calculated.

Simulation duration Length of the simulation.

SDF Output When this box is checked, analysis results will be saved in standard delay format (**.sdf**). For information on SDF syntax, see SDF Files on page 2-334.

Write net delay Writes net delays to the specified file.

Write pin-to-pin (interconnect) delay Writes interconnect (pin-to-pin) delays to the specified file.

Output file The SDF output file.

Peak voltage Maximum voltage of the specified signal.

Pulse period Duration of one cycle.

Pulse width Amount of time the signal remains at its peak voltage.

Rise/fall time Time required to reach peak voltage. The fall time is assumed to be equal to the rise time.

Driver (Source)

The output **Resistance (Ohm)** and **Capacitance (F)** of the source driver.

Receiver (Target | Load)

The input **Resistance (Ohm)** and **Capacitance (F)** of the target receiver or load.

Run

Initiates the analysis.

Netlist Navigator

Opens the **Netlist Navigator** (page 2-227) dialog.

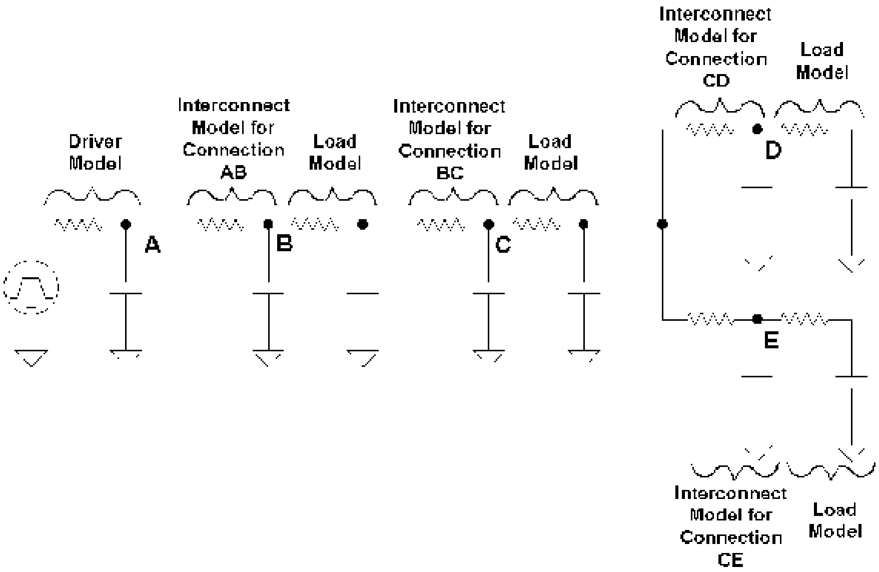
Selecting Nets for Analysis

There are two ways to select nets for analysis: In the **Netlist Navigator**, or by selecting the nets directly in the layout. Analysis will only be performed on nets that are selected.

Delay Calculations

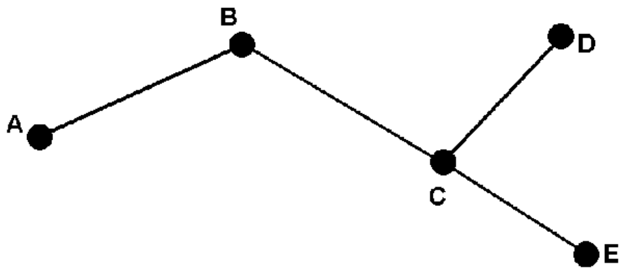
Transmission line equations are expressed as a function of time using the Telegrapher's equation. To simplify the equation, the delay calculator employs a moment matching approach using a second order moment approximation. The total delay is calculated from the moments for each routing segment.

Delay is calculated for complete and intermediate paths from the source to all other pins.


Driver Model
Interconnect Model for Connection AB
Load Model
Interconnect Model for Connection BC
Load Model
Interconnect Model for Connection CD
Load Model
A
B
C
D
E
Interconnect Model for Connection CE
Load Model

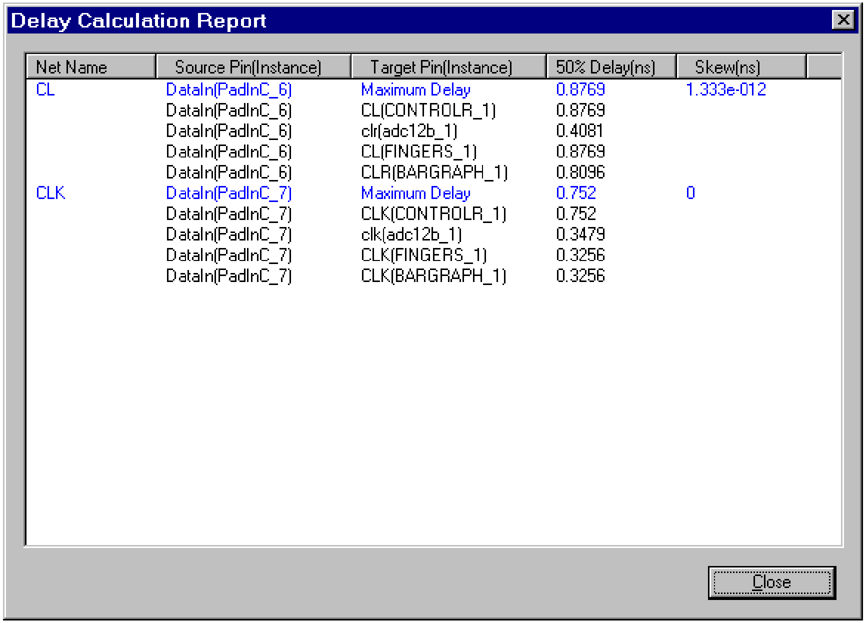
The model L-Edit uses to calculate delay.

The connectivity graph for the same circuit is shown below. Delay is calculated for paths AB, AC, AD, and AE.



Delay Calculation Report

The delay calculation report displays the results of the delay calculation analysis.



Net Name	Source Pin(Instance)	Target Pin(Instance)	50% Delay(ns)	Skew(ns)
CL	DataIn(PadInC_6)	Maximum Delay	0.8769	1.333e-012
	DataIn(PadInC_6)	CL(CONTROLR_1)	0.8769	
	DataIn(PadInC_6)	clr(adc12b_1)	0.4081	
	DataIn(PadInC_6)	CL(FINGERS_1)	0.8769	
	DataIn(PadInC_6)	CLR(BARGRAPH_1)	0.8096	
CLK	DataIn(PadInC_7)	Maximum Delay	0.752	0
	DataIn(PadInC_7)	CLK(CONTROLR_1)	0.752	
	DataIn(PadInC_7)	clk(adc12b_1)	0.3479	
	DataIn(PadInC_7)	CLK(FINGERS_1)	0.3256	
	DataIn(PadInC_7)	CLK(BARGRAPH_1)	0.3256	

Net Name Name of analyzed net.

Source Pin (Instance) Source pin and instance name. For the signal with the maximum delay, **Maximum Delay** will appear in this field.

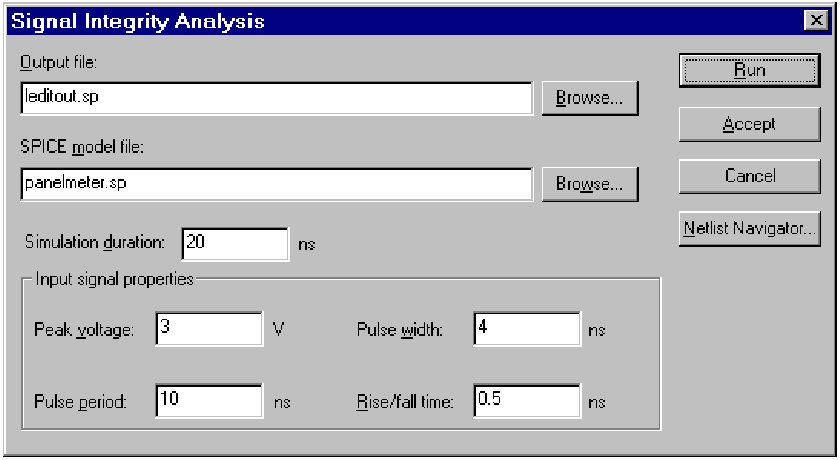
Target Pin (Instance) Target pin and instance name.

Delay (ns) Signal delay.

Skew (ns) The difference between the maximum delay and median delay for the leaf nodes in the net.

Running Signal Integrity

Use **Tools > BPR > Signal Integrity** to open the **Signal Integrity** dialog. This dialog is used to enter input signal properties and the SPICE model file to be referenced in the output file, and to initiate analysis.



Signal Integrity Analysis

Output file: leditout.sp

SPICE model file: panelmeter.sp

Simulation duration: 20 ns

Input signal properties

Peak voltage: 3 V

Pulse width: 4 ns

Pulse period: 10 ns

Rise/fall time: 0.5 ns

.

Output file The SPICE output file for time domain simulation.

SPICE model file A user-created SPICE file with interconnect (routing) models, via models, and driver and receiver models.

Simulation duration Length of the simulation.

Peak voltage Maximum voltage of the specified signal.

Pulse period Duration of one cycle.

Pulse width Amount of time the signal remains at its peak voltage.

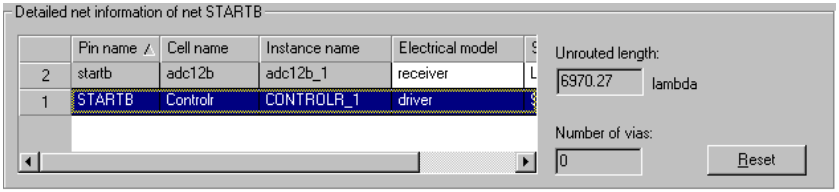
Rise/fall time Time required to reach peak voltage. The fall time is assumed to be equal to the rise time.

Run Initiate the analysis.

Netlist Navigator Opens the **Netlist Navigator** (page 2-227) dialog.

Signal Integrity Analysis

The **SPICE model file** contains model information in SPICE format for the drivers and receivers, their interconnects, and for each via used. The pin model names in this file must match the names of the models listed for each pin in the **Electrical model** column of the **Netlist Navigator**.



Detailed net information of net STARTB

	Pin name	Cell name	Instance name	Electrical model	S
2	startb	adc12b	adc12b_1	receiver	L
1	STARTB	Controlr	CONTROLR_1	driver	S

Unrouted length: 6970.27 lambda

Number of vias: 0

The resultant netlist describes the connectivity of the selected nets and includes the drivers and receivers of the pins of instances. Time domain simulation does not consider *cross-coupling*, or signal interaction between wires.

The result of time domain simulation is a SPICE format netlist that can be input to a SPICE circuit simulator for analysis. L-Edit will automatically invoke the circuit simulator T-Spice, if you have it installed, with the netlist loaded.

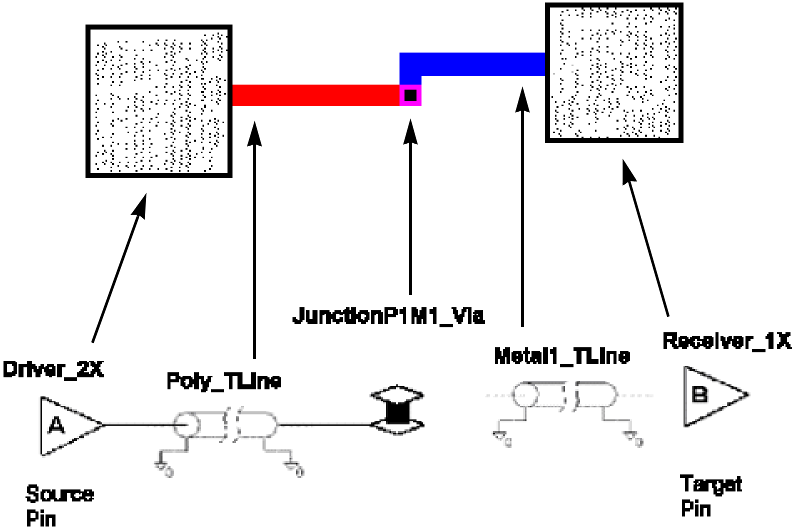
User-Supplied Models

You must supply a subcircuit model for each of the following extracted elements:

- An interconnect model for each routing layer defined for use with the BPR automatic or manual router. The subcircuit name should be **[*LayerName*]_TLine**. This subcircuit should have one input, one output, and a length parameter (for example, **.subckt Poly_TLine In Out l=1.0 Mm**).
- A via model for each via defined for use with the BPR automatic or manual router. The subcircuit name should be **[*ViaCellName*]_Via**. (If the via cell name or layer name has spaces in it, L-Edit will replace the spaces with underscores.) This subcircuit should have one input and one output (for example, **.subckt Via1 In Out)**.
- A model for each driver and receiver used in your design. These model names must match those you have entered in the **Netlist Navigator**. The driver subcircuit should have one input, one output, and a Gnd pin. For example, **.subckt driver In Out local Gnd**. The receiver subcircuit should have one output and a Gnd pin (for example, **.subckt receiver In local Gnd)**.

The following figure shows an example of the models used in time domain simulation. Each routing section is extracted as a subcircuit—in this case, **poly_TLine** and **Metal1_TLine**—with the length parameter set to the length of the segment.

The vias, drivers, and receivers are also extracted as subcircuits, as **Driver_2X**, **JunctionP1M1_Via**, and **Receiver_1X**.



Element Organization

A subcircuit model is created with all interconnect and via elements for each extracted net. Elements are organized in the output netlist as follows:

- Subcircuit elements for the extracted nets that specify the names of all of their pin connections.
- Driver/receiver subcircuit elements for each pin of the extracted nets.
- Print statements that instruct the simulator to plot the voltages at each pin of the extracted net or group of nets.
- The input voltage pulse.

Sample SPICE Model File

A sample user-provided model definition file is reproduced below, followed by explanatory comments.

```
* No Inductance yet.
.param lumptype = 2
.param NumLumps = 5

.subckt Metal1_TLine p1  p2  l=0.01
txMetal1  p1 0  p2 0 r=88.9k l=1p c=118.4p g=0 length='l'
+                    lumps='NumLumps' lumptype='lumptype'
.ends
```

```
.subckt Metal2_TLine p1  p2  l=0.01
txMetal2  p1 0  p2 0 r=77.8k l=1p c=86.9p g=0 length='l'
+                   lumps='NumLumps' lumptype='lumptype'
.ends

.subckt Metal3_TLine p1  p2  l=0.01
txMetal3  p1 0  p2 0 r=20.0k l=1p c=41.5p g=0 length='l'
+                   lumps='NumLumps' lumptype='lumptype'
.ends

.subckt ViaM1M2_Via n1 n2
rcontact n1 n2 0.61
.ends

.subckt ViaM2M3_Via n1 n2
rcontact n1 n2 0.62
.ends

.subckt Driver In Out myGND
M1 I1 In myGND myGND CMOSN L=0.6u W=8.4u AD=13.32p PD=20.4u
   AS=15.12p PS=20.4u
M2 I1 In DrvRecVDD DrvRecVDD CMOSP L=0.6u W=8.4u AD=15.12p
   PD=20.4u AS=25.92p PS=40.8u
M3 Out I1 myGND myGND CMOSN L=0.6u W=8.4u AD=13.32p PD=20.4u
   AS=15.12p PS=20.4u
M4 Out I1 DrvRecVDD DrvRecVDD CMOSP L=0.6u W=8.4u AD=15.12p
   PD=20.4u AS=25.92p PS=40.8u
.ends
```

```
.subckt Receiver In myGND
M1 Out In myGND myGND CMOSN L=0.6u W=8.4u AD=13.32p PD=20.4u
   AS=15.12p PS=20.4u
M2 Out In DrvRecVDD DrvRecVDD CMOSP L=0.6u W=8.4u AD=15.12p
   PD=20.4u AS=25.92p PS=40.8u
.ends

* N56S SPICE LEVEL3 PARAMETERS

.MODEL CMOSN NMOS LEVEL=3 PHI=0.700000 TOX=1.0000E-08
   XJ=0.200000U TPG=1
+ VTO=0.7812 DELTA=2.4510E-01 LD=4.0510E-08 KP=1.8847E-04
+ UO=545.8 THETA=2.5170E-01 RSH=2.1290E+01 GAMMA=0.6200
+ NSUB=1.3810E+17 NFS=7.0710E+11 VMAX=1.8610E+05
   ETA=2.2420E-02
+ KAPPA=9.6720E-02 CGDO=3.66E-10 CGSO=3.66E-10
+ CGBO=4.0161E-10 CJ=5.4E-04 MJ=0.6 CJSW=1.5000E-10
+ MJSW=0.32 PB=0.99
* Weff = Wdrawn - Delta_W
* The suggested Delta_W is 4.1360E-07
...
v_source driver.in  GND
+  PULSE (0 voltage 'period/4' 'riseTime' 'riseTime'
   '(period/2)-riseTime' 'period')

VDrvRecVDD DrvRecVDD 0 voltage

.GLOBAL DrvRecVDD
```

The **.param** section defines the lump type and number of lumps the transmission line models.

The first three subcircuit models are the interconnect models for routing layers **Metal1**, **Metal2**, and **Metal3**. You must define an input pin, an output pin and a length parameter. In the example shown, the input pin is **p1**, the output pin is **p2**, and the default length is **0.01** meters for each layer. L-Edit will calculate the actual length from the layout.

The second two subcircuit models define the via models used for **Metal1** to **Metal2** and **Metal2** to **Metal3** layer transitions, **ViaM1M2** and **ViaM2M3** respectively. You must specify an input and output pin for each via, in this case **n1** and **n2**.

An input pin (**In**), output pin (**Out**), and ground pin (**MyGND**) are required for the driver subcircuit model. The receiver model requires an input pin (**In**) and ground pin (**MyGND**) only.

Sample Output Netlist

A sample output netlist is reproduced below, followed by explanatory comments.

```
T-Spice header
.param riseTime = 5.0000e-001n
.param period = 1.0000e+001n
```

```
.param voltage = 3.3000e+000
.param pulseWidth = 4.0000e+000n
.param initialDelay = 2.5000e+000n
.include "C:\Bpr\BPR V1\panelmeter.sp"
*
* Extracted sub-circuit for net(s) STARTB
*
.subckt STARTB CONTROLR_1.STARTB adc12b_1.startb
x_s1 CONTROLR_1.STARTB 1 Metal2_TLine l=6.189400e-006
x_s2 1 2 Metal2_TLine l=2.052050e-006
x_s3 2 3 Metal2_TLine l=9.219000e-004
x_s4 3 4 Metal2_TLine l=5.481000e-004
x_s5 4 5 ViaM1M2_Via
x_s6 5 6 Metal1_TLine l=6.993000e-004
x_s7 6 7 ViaM1M2_Via
x_s8 7 8 Metal2_TLine l=5.481000e-004
x_s9 8 9 ViaM1M2_Via
x_s10 9 10 Metal1_TLine l=8.400000e-005
x_s11 10 11 ViaM1M2_Via
x_s12 11 12 Metal2_TLine l=4.200000e-006
x_s13 12 13 Metal2_TLine l=5.250000e-007
x_s14 13 14 Metal2_TLine l=2.520350e-006
x_s15 14 adc12b_1.startb Metal2_TLine l=9.312450e-006
.ends
*
* Sub-circuit instance for net(s) STARTB
*
x_STARTB CONTROLR_1.STARTB adc12b_1.startb STARTB
xSTARTB1 driver.in CONTROLR_1.STARTB GND driver
```

```
xSTARTB2 adc12b_1.startb GND receiver
.print tran v(driver.in) v(CONTROLR_1.STARTB)
   v(adc12b_1.startb)
.options prtdel=1.0000e-002n
.tran 5.0000e-001n 2.000000e+001n
```

The header section reports the signal values and SPICE model file entered in the **Signal Integrity Analysis** dialog.

The **.subckt/.ends** command block defines subcircuit **STARTB**, representing the interconnects and vias for net **STARTB** from the layout. The definition incorporates instances of two interconnects, **Metal2** and **Metal1**. The net has two connecting nodes *(**20** and **21**)*; these nodes represent the net's connections to the component pins.

The **.print** command ensures that when this netlist is simulated, the transient (time-domain) voltages at the two block pins will be reported to the output file.