Chapter 22

Performing Behavioral Modeling

Behavioral modeling refers to the substitution of more abstract, less computationally intensive circuit models for lower level descriptions of analog functions. These simpler models emulate the transfer characteristics of the circuit elements that they replace, but with increased efficiency, leading to substantial reduction in the actual simulation time per circuit. This reduction in elapsed time per simulation, when considering the whole of the design and simulation cycle, can lead to a tremendous increase in design efficiency, as well as possible reduction in the time necessary to take a design from a concept to a marketable product.

This chapter describes how to create behavioral models in the following topics:

Understanding the Behavioral Design Process

Star-Star-Hspice provides specific modeling elements that promote the use of behavioral and mixed signal techniques. These models include controllable sources that may be configured to emulate op-amps, single- or multi-input logic gates, or any system with a continuous algebraic transfer function. These functions may be in algebraic form or in the form of coordinate pairs. Digital stimulus files are useful features that allow you to enter a number of logic waveforms into the simulation deck without resorting to the awkward procedure of entering digital waveforms using piecewise linear sources. You can define clock rise times, fall times, periods, and voltage levels.

The typical design cycle of a circuit or system using Star-Star-Hspice behavioral models is described below.

- Perform full simulation of a subcircuit with pertinent inputs, characterizing its transfer functions.
- Determine which of the Star-Star-Hspice elements, singularly or in combination, accurately describe the transfer function.
- Reconfigure the subcircuit appropriately.
- After the behavioral model is verified, substitute the model into the larger system in place of the lower level subcircuit.

Using Behavioral Elements

Behavioral elements offer a higher level of abstraction and a faster processing time over the lower level description of an analog function. For system-level designers, function libraries of subcircuits containing these elements are used to describe parts such as op-amps, vendor specific output buffer drivers, TTL drivers, logic-to-analog and analog-to-logic simulator converters. For the integrated circuit designer, these elements offer a fast representation that is particularly useful in filter and signal processing design.

Behavioral elements are based on using an arbitrary algebraic equation as a transfer function to a voltage (E) or current (G) source. This function can include nodal voltages, element currents, time, or user defined parameters. A good example of this is a VCO where “control” is the input voltage node and “osc” is the oscillator output:

```
 Evco osc 0 VOL=’voff+gain*SIN(6.28*freq*(1+V(control))*TIME)’
```

Subcircuits provide a way to encapsulate a function. If you split the function definition from the use, you create a a hierarchy. If you pass parameters into the subcircuit, you create a parameterized cell. If you create a full transistor cell library and a behavioral representation library, you can deal with mixed signal functions within Star-Star-Hspice. You can calibrate the behavioral elements from a full transistor circuit using the built-in OPTIMIZE function.

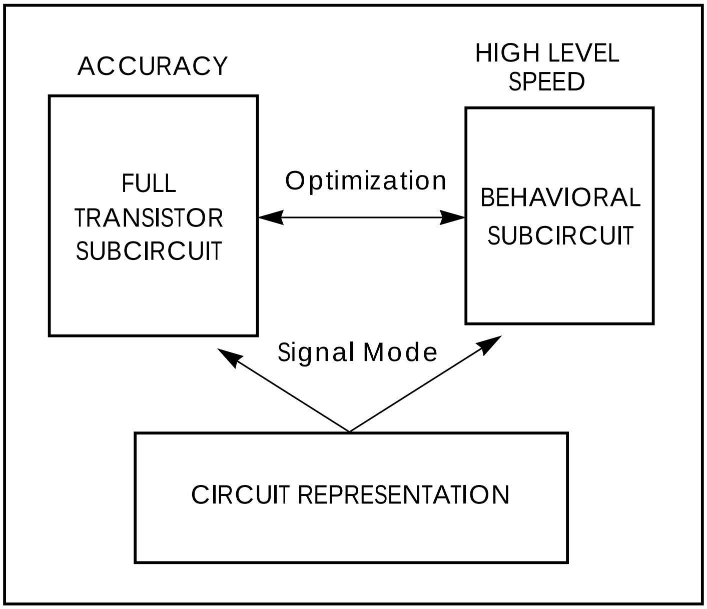


Figure 22-1: Netlisting by Signal Mode

Controlled Sources

Controlled sources model both analog and digital circuits at the behavioral level, allowing for fast mixed-signal simulation times and providing a means to model system level operations. Controlled sources model gate switching action for the behavioral modeling of digital circuits. For analog behavioral modeling, the controlled sources can be programmed as mathematical functions that are either linear or nonlinear, dependent on other nodal voltages and branch currents.

Digital Stimulus Files

Complex transition files are difficult to process using the piecewise linear sources. You can use the U element A2D and D2A conversion functions to simplify processing of transition files. The A2D function converts analog output to digital data, and the D2A function converts digital input data to analog. You can export output to logic or VHDL simulators as well.

Behavioral Examples

The examples of analog and digital elements in this chapter give some insight into how the behavioral elements operate.

Op-Amp Subcircuit Generators

Operational amplifiers are automatically designed using the subcircuit generator to meet given electrical specifications, such as PSRR, CMRR, and Vos. The generator produces component values for each of the elements in the design. The subcircuits produced by combining these values offer faster simulation times than conventional circuit level implementations.

Libraries

Use the Discrete Device Library of standard industry IC components to model board level designs that contain transistors, diodes, opamps, comparators, converters, IC pins, printed circuit board traces and coaxial cables. You can model drivers and receivers to analyze transmission line effects and power and signal line noise.

Using Subcircuits

Reusable cells are the key to saving labor in any CAD system, and this also applies to circuit simulation. To create a reusable circuit, it must be constructed as a subcircuit. Use parameters to expand the utility of a subcircuit. SPICE includes the basic subcircuit but does not provide for the consistent naming of nodes. Star-Star-Hspice provides a simple method for the naming of the subcircuit nodes and elements. To use this, you prefix the node or element with the subcircuit call name.

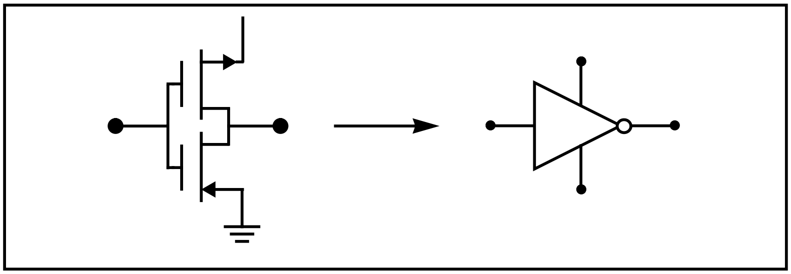


Figure 22-2: Subcircuit Representation

```
X1 IN OUT VD_LOCAL VS_LOCAL inv W=20
.MACRO INV IN OUT VDD VSS W=10 L=1 DJUNC=0
   MP OUT IN VDD VDD PCH W=W L=L DTEMP=DJUNC
   MN OUT IN VSS VSS NCH W='W/2' L=L DTEMP=DJUNC

.EOM
```

Note: To access the name of the MOSFET inside of the subcircuit INV called by X1, the names are X1.MP and X1.MN. So to print the current through the MOSFETs:

```
.PRINT I(X1.MP)
```

Hierarchical Parameters

The most basic subcircuit parameter is the M or multiply parameter. This is actually a keyword common to all elements including subcircuits, except for voltage sources. The multiply parameter multiplies the internal component values to give the effect of making parallel copies of the element or subcircuit. To simulate the effect of 32 output buffers switching simultaneously, you only need to place one subcircuit :

```
X1 in out buffer M=32
```

Multiply works hierarchically. A subcircuit within a subcircuit is multiplied by the product of both levels.

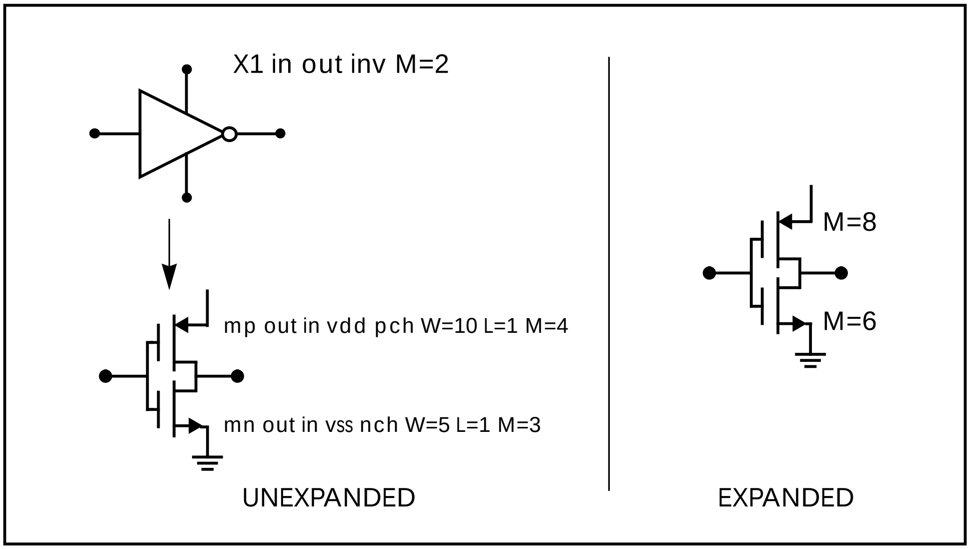


Figure 22-3: Using Hierarchical Parameters Simplifies Flip-flop Initialization

Example

```
X1 D Q Qbar CL CLBAR dlatch flip=0
macro dlatch
+ D Q Qbar CL CLBAR flip=vcc
.nodeset v(din)=flip
xinv1 din qbar inv
xinv2 Qbar Q inv
m1 q CLBAR din nch w=5 l=1
m2 D CL din nch w=5 l=1
.eom
```

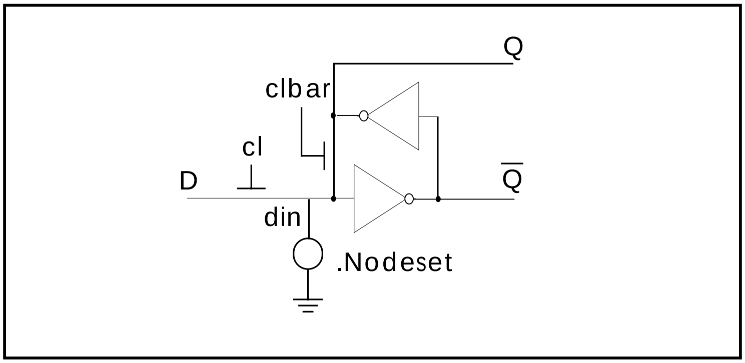


Figure 22-4: D Latch with Nodeset

There is no limit to the size or complexity of subcircuits; they may contain subcircuit references and any model or element statement. To specify subcircuit nodes in .PRINT or .PLOT statements, give the full subcircuit path and node name.

.SUBCKT or .MACRO Statement

The syntax is:

```
.SUBCKT subnam n1 < n2 n3 …> < parnam=val …>
```

or

```
.MACRO subnam n1 < n2 n3 … > < parnam=val …>
```

where:

subnam specifies reference name for the subcircuit model call

n1 … snode numbers for external reference; cannot be ground node (zero). Any element nodes appearing in the subcircuit but not included in this list are strictly local, with three exceptions:

1. the ground node (zero)

2. nodes assigned using BULK=node in the MOSFET or BJT models

3. nodes assigned using the .GLOBAL statement

parnam a parameter name set to a value. For use only in the subcircuit, overridden by an assignment in the subcircuit call or by a value set in a .PARAM statement.

Examples of Subcircuit Test

```
*FILE SUB2.SP TEST OF SUBCIRCUITS
.OPTIONS LIST ACCT
*
V1 1 0 1
.PARAM P5=5 P2=10
*
.SUBCKT SUB1 1 2 P4=4
R1 1 0 P4
R2 2 0 P5
X1 1 2 SUB2 P6=7
X2 1 2 SUB2
.ENDS
*
```

```
.MACRO SUB2 1 2 P6=11
R1 1 2 P6
R2 2 0 P2
.EOM
*
X1 1 2 SUB1 P4 =6
X2 3 4 SUB1 P6=15
X3 3 4 SUB2
*
.MODEL DA D  CJA=CAJA CJP=CAJP VRB=-20 IS=7.62E-18
+   PHI=.5 EXA=.5 EXP=.33
*
.PARAM CAJA=2.535E-16 CAJP=2.53E-16
.END
```

The above example defines two subcircuits: SUB1 and SUB2. These are resistor divider networks whose resistance values have been parameterized. They are called with the X1, X2, and X3 statements. Since the resistor value parameters are different in each call, these three calls produce different subcircuits.

.ENDS or .EOM Statement

The syntax is:

```
.ENDS <SUBNAM>
```

or

```
.EOM <SUBNAM>
```

Examples

```
.ENDS OPAMP
.EOM MAC3
```

This statement must be the last for any subcircuit definition. The subcircuit name, if included, indicates which definition is being terminated. Subcircuit references (calls) may be nested within subcircuits.

Subcircuit Call Statement

The syntax is:

```
Xyyy n1 <n2 n3 …> subnam <parnam=val …> <M=val>
```

where: specifies:

Xyyy subcircuit element name. Must begin with an “X”, which may be followed by up to 15 alphanumeric characters.

n1 … node names for external reference

subnam subcircuit model reference name

parnam a parameter name set to a value (val) for use only in the subcircuit. It overrides a parameter value assigned in the subcircuit definition, but is overridden by a value set in a .PARAM statement.

M multiplier. Makes the subcircuit appear as M subcircuits in parallel. This is useful in characterizing circuit loading. No additional calculation time is needed to evaluate multiple subcircuits.

Example

```
X1 2 4 17 31 MULTI WN=100 LN=5
```

The above example calls a subcircuit model named MULTI. It assigns the parameters WN=100 and LN=5 to the parameters WN and LN given in the .SUBCKT statement (not shown). The subcircuit name is X1. All subcircuit names must begin with X.

Example

```
.SUBCKT YYY NODE1 NODE2 VCC=5V
.IC NODEX=VCC
R1 NODE1 NODEX 1
R2 NODEX NODE2 1
.EOM
XYYY 5 6 YYY VCC=3V
```

The above example defines a subcircuit named YYY. The subcircuit consists of two 1 ohm resistors in series. The subcircuit node, NODEX, is initialized with the .IC statement through the passed parameter VCC.

Note: A warning message is generated if a nonexistent subcircuit node is initialized. This can occur if an existing .ic file (initial conditions) is used to initialize a circuit modified since the .ic file was created.

Undefined Subcircuit Search

When a subcircuit call is in a data file that does not contain the subcircuit description, Star-Star-Hspice automatically searches the:

1. present directory for the file
2. directories specified in any .OPTION SEARCH= “directory_path_name” statement
3. directory where the Discrete Device Library is located.

Star-Star-Hspice searches for the model reference name file with an *.inc* suffix. For example, if an undefined subcircuit such as “X 1 1 2 INV” is included in the data file, Star-Star-Hspice searches the system directories for the file called *inv.inc* and when found, places it in the calling data file.

Using Voltage and Current Controlled Elements

In Star-Star-Hspice there are four voltage and current controlled elements, known as G, E, H and F elements. You can use these controlled elements to model the following:

- MOS and bipolar transistors
- Tunnel diodes
- SCRs

and analog functions such as

- Operational amplifiers
- Summers
- Comparators
- Voltage controlled oscillators
- Modulators
- Switched capacitor circuits

The controlled elements can be either linear or nonlinear functions of controlling node voltages or branch currents, depending on whether you used the polynomial or piecewise linear functions.

The functions of the G, E, F, and H controlled elements are different. The G element can be a voltage or current controlled current source, a voltage controlled resistor, a piecewise linear voltage controlled capacitor, an ideal delay element, or a piecewise linear multi-input AND, NAND, OR, or NOR gate.

The E elements can be a voltage or current controlled voltage source, an ideal op-amp, an ideal transformer, an ideal delay element, or a piecewise linear voltage controlled multi-input AND, NAND, OR, or NOR gate.

The H element can be a current controlled voltage source, an ideal delay element, or a piecewise linear current controlled multi-input AND, NAND, OR, or NOR gate.

The F element can be a current controlled current source, an ideal delay element, or a piecewise linear current controlled multi-input AND, NAND, OR, or NOR gate.

Polynomial and piecewise linear functions are discussed below. Element statements for linear or nonlinear functions are described in the following sections.

Polynomial Functions

The controlled element statement allows the definition of the controlled output variable (current, resistance, or voltage) as a polynomial function of one or more voltages or branch currents. There are three polynomial equations that can be selected through the POLY(ndim) parameter in the E, F, G, or H element statement.

POLY(1)	one-dimensional equation
POLY(2)	two-dimensional equation
POLY(3)	three-dimensional equation

The POLY(1) polynomial equation specifies a polynomial equation as a function of one controlling variable, POLY(2) as a function of two controlling variables, and POLY(3) as a function of three controlling variables.

Along with each polynomial equation are polynomial coefficient parameters (P0, P1 … Pn) that can be set to explicitly define the equation.

One-Dimensional Function

If the function is one-dimensional, that is, a function of one branch current or node voltage, the function value FV is determined by the following expression:

$$= P0 + (P1 \cdot FA) + (P2 \cdot FA^2) + (P3 \cdot FA^3) + (P4 \cdot FA^4) + (P5 \cdot FA^5) + \ldots$$

FV	the controlled voltage or current from the controlled source
P0. . .Pn	coefficients of polynomial equation
FA	the controlling branch current or nodal voltage

Note: If the polynomial is one-dimensional and exactly one coefficient is specified, Star-Star-Hspice assumes it to be P1 (P0 = 0.0), in order to facilitate the input of linear controlled sources.

The following controlled source statement is an example of a one-dimensional function:

```
E1 5 0 POLY(1) 3 2 1 2.5
```

The above voltage controlled voltage source is connected to nodes 5 and 0. The single dimension polynomial function parameter, POLY(1), informs Star-Star-Hspice that E1 is a function of the difference of one nodal voltage pair, in this case, the voltage difference between nodes 3 and 2, hence FA=V(3,2). The dependent source statement then specifies that P0=1 and P1=2.5. From the one-dimensional polynomial equation above, the defining equation for E1 is:

$$E1 = 1 + 2.5 \cdot V(3,2)$$

Two-Dimensional Function

Where the function is two-dimensional, that is, a function of two node voltages or two branch currents, FV is determined by the following expression:

$$FV = P0 + (P1 \cdot FA) + (P2 \cdot FB) + (P3 \cdot FA^2) + (P4 \cdot FA \cdot FB) + (P5 \cdot FB^2) \\ + (P6 \cdot FA^3) + (P7 \cdot FA^2 \cdot FB) + (P8 \cdot FA \cdot FB^2) + (P9 \cdot FB^3) + \ldots$$

For a two-dimensional polynomial, the controlled source is a function of two nodal voltages or currents. To specify a two-dimensional polynomial, set POLY(2) in the controlled source statement.

For example, generate a voltage controlled source that gives the controlled voltage, E1, as:

$$E1 = 3 \cdot V(3,2) + 4 \cdot V(7,6)^2$$

To implement this function, use the following controlled source element statement:

```
E1 1 0 POLY(2) 3 2 7 6 0 3 0 0 0 4
```

This specifies a controlled voltage source connected between nodes 1 and 0 that is controlled by two differential voltages: the voltage difference between nodes 3 and 2 and the voltage difference between nodes 7 and 6, that is, FA=V(3,2) and FB=V(7,6). The polynomial coefficients are P0=0, P1=3, P2=0, P3=0, P4=0, and P5=4.

Three-Dimensional Function

For a three-dimensional polynomial function with arguments FA, FB, and FC, the function value FV is determined by the following expression:

$$\begin{aligned} &= P0 + (P1 \cdot FA) + (P2 \cdot FB) + (P3 \cdot FC) + (P4 \cdot FA^2) \\ &\quad + (P5 \cdot FA \cdot FB) + (P6 \cdot FA \cdot FC) + (P7 \cdot FB^2) + (P8 \cdot FB \cdot FC) \\ &\quad + (P9 \cdot FC^2) + (P10 \cdot FA^3) + (P11 \cdot FA^2 \cdot FB) + (P12 \cdot FA^2 \cdot FC) \\ &\quad + (P13 \cdot FA \cdot FB^2) + (P14 \cdot FA \cdot FB \cdot FC) + (P15 \cdot FA \cdot FC^2) \\ &\quad + (P16 \cdot FB^3) + (P17 \cdot FB^2 \cdot FC) + (P18 \cdot FB \cdot FC^2) \\ &\quad + (P19 \cdot FC^3) + (P20 \cdot FA^4) + \dots \end{aligned}$$

For example, generate a voltage controlled source that gives the voltage as:

$$E1 = 3 \cdot V(3,2) + 4 \cdot V(7,6)^2 + 5 \cdot V(9,8)^3$$

From the above defining equation and the three-dimensional polynomial equation:

$$FA = V(3,2)$$

$$FB = V(7,6)$$

$$FC = V(9,8)$$

$$P1 = 3$$

$$P7 = 4$$

$$P19 = 5$$

Substituting these values into the voltage controlled voltage source statement:

```
E1 1 0 POLY(3) 3 2 7 6 9 8 0 3 0 0 0 0 0 4 0 0 0 0 0 0 0 0 0
+  0 0 5
```

The above specifies a controlled voltage source connected between nodes 1 and 0 that is controlled by three differential voltages: the voltage difference between nodes 3 and 2, the voltage difference between nodes 7 and 6, and the voltage difference between nodes 9 and 8, that is, FA=V(3,2), FB=V(7,6), and FC=V(9,8). The statement gives the polynomial coefficients as P1=3, P7=4, P19=5, and the rest are zero.

Piecewise Linear Function

The one-dimensional piecewise linear (PWL) function allows designers to model some special element characteristics, such as those of tunnel diodes, silicon controlled rectifiers. and diode breakdown regions. You can describe the piecewise linear function by specifying measured data points. Although the device characteristic is described by data points, Star-Star-Hspice automatically smooths the corners to ensure derivative continuity and, as a result, better convergence.

A parameter DELTA is provided to control the curvature of the characteristic at the corners. The smaller the DELTA, the sharper the corners are. The maximum value allowed for DELTA is half the smallest of the distances between breakpoints. Specify a DELTA that provides satisfactory sharpness of the function corners. You can specify up to 100 breakpoint pairs. You must specify at least two point pairs (with each point consisting of an x and a y coefficient).

The functions NPWL and PPWL can be used for modeling bidirectional switch or transfer gates using G elements. The NPWL and PPWL functions behave like NMOS and PMOS transistors.

The piecewise linear function also is used to model multi-input AND, NAND,OR, and NOR gates. In this case only one input determines the state of the output. In AND and NAND gates, the input with the smallest value is used in the piecewise linear function to determine the corresponding output of the gates. In the OR and NOR gates, the input with the largest value is used to determine the corresponding output of the gates.

Dependent Current Sources — G Elements

Voltage Controlled Current Source (VCCS)

The syntax is:

Linear

```
Gxxx n+ n- <VCCS> in+ in- transconductance <MAX=val> <MIN=val>
<SCALE=val>
+         <M=val> <TC1=val> <TC2=val> <ABS=1> <IC=val>
```

Polynomial

```
Gxxx  n+ n- <VCCS> POLY(ndim) in1+ in1- ... <inndim+ inndim-> MAX=val>
<MIN=val>
+         <SCALE=val> <M=val> <TC1=val> <TC2=val> <ABS=1>   p0 <p1…>
<IC=vals>
```

Piecewise Linear

```
Gxxx  n+ n- <VCCS> PWL(1) in+ in- <DELTA=val> <SCALE=val> <M=val>
<TC1=val>
+         <TC2=val> x1,y1 x2,y2 ... x100,y100 <IC=val> <SMOOTH=val>
```

```
Gxxx  n+ n- <VCCS> NPWL(1) in+ in- <DELTA=val> <SCALE=val> <M=val>
<TC1=val>
+         <TC2=val> x1,y1 x2,y2 ... x100,y100 <IC=val> <SMOOTH=val>
```

```
Gxxx  n+ n- <VCCS> PPWL(1) in+ in- <DELTA=val> <SCALE=val> <M=val>
<TC1=val>
+         <TC2=val> x1,y1 x2,y2 ... x100,y100 <IC=val> <SMOOTH=val>
```

Multi-Input Gates

```
Gxxx  n+ n- <VCCS> gatetype(k) in1+ in1- ... ink+ ink- <DELTA=val> <TC1=val>
<TC2=val>
+         <SCALE=val> <M=val> x1,y1  ... x100,y100<IC=val>
```

Delay Element

```
Gxxx n+ n- <VCCS> DELAY in+ in- TD=val <SCALE=val> <TC1=val> <TC2=val>
+         NPDELAY=val
```

Behavioral Current Source

The syntax is:

Gxxx n+ n- CUR=’equation’ <MAX>=val> <MIN=val>

Voltage Controlled Resistor (VCR)

The syntax is:

Linear

Gxxx n+ n- VCR in+ in- transfactor <MAX=val> <MIN=val> <SCALE=val> <M=val>
+ <TC1=val> <TC2=val> <IC=val>

Polynomial

Gxxx n+ n- VCR POLY(ndim) in1+ in1- ... <inndim+ inndim-> <MAX=val> <MIN=val>
+ <SCALE=val> <M=val> <TC1=val> <TC2=val> p0 <p1…> <IC=vals>

Piecewise Linear

Gxxx n+ n- VCR PWL(1) in+ in- <DELTA=val> <SCALE=val> <M=val> <TC1=val>
+ <TC2=val> x1,y1 x2,y2 ... x100,y100 <IC=val> <SMOOTH=val>

Gxxx n+ n- VCR NPWL(1) in+ in- <DELTA=val> <SCALE=val> <M=val> <TC1=val>
+ <TC2=val> x1,y1 x2,y2 ... x100,y100 <IC=val> <SMOOTH=val>

Gxxx n+ n- VCR PPWL(1) in+ in- <DELTA=val> <SCALE=val> <M=val> <TC1=val>
+ <TC2=val> x1,y1 x2,y2 ... x100,y100 <IC=val> <SMOOTH=val>

Multi-Input Gates

Gxxx n+ n- VCR gatetype(k) in1+ in1- ... ink+ ink- <DELTA=val>
+ <TC1=val> <TC2=val> <SCALE=val> <M=val> x1,y1 ... x100,y100 <IC=val>

Voltage Controlled Capacitor (VCCAP)

The syntax is:

```
Gxxx  n+ n- VCCAP PWL(1) in+ in-   <DELTA=val> <SCALE=val> <M=val>
<TC1=val>
+          <TC2=val> x1,y1 x2,y2 ... x100,y100 <IC=val> <SMOOTH=val>
```

The two functions NPWL and PPWL allow the interchange of the ‘n+’ and ‘n-’ nodes while keeping the same transfer function. This action can be summarized as follows:

NPWL Function

For node ‘in- ‘connected to ‘n-’;

If v(n+,n-) > 0, then the controlling voltage would be v(in+,in-). Otherwise, the controlling voltage is v(in+,n+)

For node ‘in-’connected to ‘n+’;

If v(n+,n-) < 0, then the controlling voltage would be v(in+,in-). Otherwise, the controlling voltage is v(in+,n+)

PPWL Function

For node’ in-’connected to ‘n-’;

If v(n+,n-) < 0, then the controlling voltage would be v(in+,in1-). Otherwise, the controlling voltage is v(in+,n+)

For node ’in-’connected to ‘n+‘;

If v(n+,n-) > 0, then the controlling voltage would be v(in+,in-). Otherwise, the controlling voltage is v(in+,n+)

G Element Parameters

ABS Output is absolute value if ABS=1.

CUR=equation current output which flows from n+ to n-. The “equation”, which is defined by the user, can be a function of node voltages, branch currents, TIME, temperature (TEMPER), and frequency (HERTZ).

DELAY keyword for the delay element. The delay element is the same as voltage controlled current source except it is associated by a propagation delay TD. This element facilitates the adjustment of propagation delay in the subcircuit model process. DELAY is an Star-Star-Hspice keyword and should not be used as a node name.

DELTA used to control the curvature of the piecewise linear corners. Defaults to 1/4 of the smallest of the distances between breakpoints. The maximum is 1/2 of the smallest of the distances between breakpoints.

Gxxx voltage controlled element name. Must begin with “G”, which may be followed by up to 15 alphanumeric characters.

gatetype(k) may be AND, NAND, OR, or NOR. The value of k is the number of inputs of the gate. The x and y terms represent the piecewise linear variation of output as a function of input. In the multi-input gates only one input determines the state of the output.

IC initial condition. The initial estimate of the value(s) of the controlling voltage(s). If IC is not specified, Default=0.0.

in +/- positive or negative controlling nodes. Specify one pair for each dimension.

M number of replications of the element in parallel

MAX maximum current or resistance value. The default is undefined, and sets no maximum value.

MIN minimum current or resistance value. The default is undefined, and sets no minimum value.

n+/- positive or negative node of controlled element

NPDELAY sets the number of data points to be used in delay simulations. The default value is the larger of 10 or the smaller of TD/tstep and tstop/tstep. That is,

$$NPDELAY_{default} = max\left[\frac{min\langle TD, tstop\rangle}{tstep}, 10\right]$$

The values of tstep and tstop are specified in the .TRAN statement.

NPWL models the symmetrical bidirectional switch or transfer gate, NMOS

p0, p1 … the polynomial coefficients. When one coefficient is specified, Star-Star-Hspice assumes it to be p1, with p0=0.0, and the element is linear. When more than one polynomial coefficient is specified by p0, p1, p2, ..., the element is nonlinear. See “Polynomial Functions” on page -14.

POLY polynomial dimension. If POLY(ndim) is not specified, a one-dimensional polynomial is assumed. Ndim must be a positive number.

PWL piecewise linear function keyword

PPWL models the symmetrical bidirectional switch or transfer gate, PMOS

SCALE element value multiplier

SMOOTH For piecewise linear dependent source elements, SMOOTH selects the curve smoothing method.

A curve smoothing method simulates exact data points you provide. This method can be used to make Star-Star-Hspice simulate specific data points that correspond to measured data or data sheets, for example.

Choices for SMOOTH are 1 or 2:

1	Selects the smoothing method used in Star-Hspice releases prior to Release H93A. Use this method if you need to maintain compatibility with simulations done using releases older than H93A.
2	Selects the smoothing method that uses data points you provide. This is the default for HPICE releases starting with H93A.

TC1,TC2 first and second order temperature coefficients. The SCALE is updated by temperature:

$$SCALEeff = SCALE \cdot (1 + TC1 \cdot \Delta t + TC2 \cdot \Delta t^2)$$

TD time delay keyword

transconductance voltage to current conversion factor

*transfacto*r voltage to resistance conversion factor

VCCAP the keyword for voltage controlled capacitance element. VCCAP is an Star-Hspice keyword and should not be used as a node name.

VCCS the keyword for voltage controlled current source. VCCS is an Star-Hspice keyword and should not be used as a node name.

VCR the keyword for voltage controlled resistor element. VCR is an Star-Hspice keyword and should not be used as a node name.

x1,... controlling voltage across nodes in+ and in-. The x values must be in increasing order.

y1,... corresponding element values of x

Examples

Switch

A voltage controlled resistor represents a basic switch characteristic. The resistance between nodes 2 and 0 varies linearly from 10meg to 1m ohms when voltage across nodes 1 and 0 varies between 0 and 1 volt. Beyond the voltage limits, the resistance remains at 10meg and 1m ohms respectively.

```
 Gswitch 2 0 VCR PWL(1) 1 0  0v,10meg 1v,1m
```

Switch-level MOSFET

A switch level n-channel MOSFET can be modelled by the N-piecewise linear resistance switch. The resistance value does not change when the node d and s positions are switched.

```
Gnmos d s VCR NPWL(1) g s LEVEL=1 0.4v,150g 1v,10meg
2v,50k 3v,4k 5v,2k
```

Voltage Controlled Capacitor

The capacitance value across nodes (out,0) varies linearly from 1p to 5p when voltage across nodes (ctrl,0) varies between 2v and 2.5v. Beyond the voltage limits, the capacitance value remains constant at 1 picofarad and 5 picofarads respectively.

```
Gcap out 0 VCCAP PWL(1) ctrl 0  2v,1p 2.5v,5p
```

Zero Delay Gate

A two-input AND gate can be implemented using an expression and a piecewise linear table. The inputs are voltages at nodes a and b, and the output is the current flow from node out to 0. The current is multiplied by the SCALE value, which, in this example, is specified as the inverse of the load resistance connected across the nodes (out,0).

```
Gand out 0 AND(2) a 0 b 0 SCALE='1/rload' 0v,0a 1v,.5a
4v,4.5a 5v,5a
```

Delay Element

A delay is a low-pass filter type delay similar to that of an opamp. A transmission line, on the other hand, has an infinite frequency response. A glitch input to a G delay is attenuated similarly to a buffer circuit. In this example, the output of the delay element is the current flow from node *out* to node *1* with a value equal to the voltage across nodes (*in*, *0*) multiplied by SCALE value and delayed by TD value.

```
Gdel out 0 DELAY in 0 TD=5ns SCALE=2 NPDELAY=25
```

Diode Equation

A forward bias diode characteristic from node 5 to ground can be modelled with a run time expression. The saturation current is 1e-14 amp, and the thermal voltage is 0.025v.

```
Gdio 5 0 CUR=’1e-14*(EXP(V(5)/0.025)-1.0)’
```

Diode Breakdown

Diode breakdown region to forward region can be modelled. When voltage across diode goes beyond the piecewise linear limit values (-2.2v, 2v), the diode current remains at the corresponding limit values (-1a, 1.2a).

```
Gdiode 1 0 PWL(1) 1 0 -2.2v,-1a -2v,-1pa .3v,.15pa.6v,10ua
1v,1a + 2v,1.2a
```

Triodes

Both the following voltage controlled current sources implement a basic triode. The first uses the poly(2) operator to multiply the anode and grid voltages together and scale by .02. The next example uses the explicit behavioral algebraic description.

```
gt i_anode cathode poly(2) anode,cathode grid,cathode 0 0
0 0 .02 gt i_anode cathode cur=’20m*v(anode,cathode)
*v(grid,cathode)’
```

Dependent Voltage Sources — F Elements

Current Controlled Current Source (CCCS)

The syntax is:

Linear

```
Fxxx n+ n-  <CCCS>  vn1  gain <MAX=val> <MIN=val> <SCALE=val>
<TC1=val> <TC2=val>
+         <M=val> <ABS=1> <IC=val>
```

Polynomial

```
Fxxx n+ n- <CCCS> POLY(ndim) vn1 <... vnndim> <MAX=val> <MIN=val>
<TC1=val>
+         <TC2=val> <SCALE=vals> <M=val> <ABS=1> p0 <p1…> <IC=vals>
```

Piecewise Linear

```
Fxxx n+  n-  <CCCS>  PWL(1)  vn1 <DELTA=val> <SCALE=val><TC1=val>
<TC2=val>
+         <M=val>   x1,y1  ...  x100,y100 <IC=val>
```

Multi-Input Gates

```
Fxxx  n+ n- <CCCS> gatetype(k) vn1, ... vnk <DELTA=val> <SCALE=val>
<TC1=val>
+         <TC2=val> <M=val> <ABS=1> x1,y1   ...   x100,y100    <IC=val>
```

Delay Element

```
Fxxx n+  n-  <CCCS>  DELAY  vn1  TD=val <SCALE=val> <TC1=val><TC2=val>
+         NPDELAY=val
```

Note: G elements with algebraics make CCCS elements obsolete. However, CCCS elements may still be used for backward compatibility with existing designs.

F Element Parameters

ABS Output is absolute value if ABS=1.

CCCS keyword for current controlled current source. CCCS is an Star-Hspice keyword and should not be used as a node name

DELAY keyword for the delay element. The delay element is the same as a current controlled current source except it is associated by a propagation delay TD.This element facilitates the adjustment of propagation delay in the subcircuit model process. DELAY is an Star-Hspice keyword and should not be used as a node name.

DELTA used to control the curvature of the piecewise linear corners. Defaults to 1/4 of the smallest of the distances between breakpoints. The maximum is 1/2 of the smallest of the distances between breakpoints.

Fxxx current controlled current source element name. Must begin with “F”, which may be followed by up to 15 alphanumeric characters.

gain current gain

gatetype(k) may be AND, NAND, OR, or NOR. The value of k is the number of inputs of the gate. The x and y terms represent the piecewise linear variation of output as a function of input. In the multi-input gates only one input determines the state of the output. The above keywords should not be used as node names.

IC initial condition: the initial estimate of the value(s) of the controlling current(s) in amps. Default=0.0.

M number of replications of the element in parallel

MAX maximum output current value. The default is undefined, and sets no maximum value.

MIN minimum output current value. The default is undefined, and sets no minimum value.

n+/- positive or negative controlled source connecting nodes

NPDELAY sets the number of data points to be used in delay simulations. The default value is the larger of 10 or the smaller of TD/tstep and tstop/tstep. That is,

$$NPDELAY_{default} = max\left[\frac{min\langle TD, tstop\rangle}{tstep}, 10\right]$$

The values of tstep and tstop are specified in the .TRAN statement.

p0, p1 … the polynomial coefficients. When one coefficient is specified, Star-Star-Hspice assumes it to be p1, with p0=0.0, and the element is linear. When more than one polynomial coefficient is specified by p0, p1, p2, ..., the element is nonlinear. See “Polynomial Functions” on page -14.

POLY polynomial dimension. If POLY(ndim) is not specified, a one-dimensional polynomial is assumed. Ndim must be a positive number.

PWL piecewise linear function keyword

SCALE element value multiplier

TC1,TC2 first and second order temperature coefficients. The SCALE is updated by temperature:

$$SCALEeff = SCALE \cdot (1 + TC1 \cdot \Delta t + TC2 \cdot \Delta t^2)$$

TD time delay keyword

vn1 … names of voltage sources through which the controlling current flows. One name must be specified for each dimension.

x1,... controlling current through vn1 source. The x values must be in increasing order.

y1,... corresponding output current values of x

Examples

```
$ Current controlled current sources - F elements,
F1 13 5 VSENS MAX=+3 MIN=-3 5
F2 12 10 POLY VCC 1MA 1.3M
Fd 1 0 DELAY vin TD=7ns SCALE=5
Filim 0 out PWL(1) vsrc -1a,-1a 1a,1a
```

The first example describes a current controlled current source connected between nodes 13 and 5. The current that controls the value of the controlled source flows through the voltage source named VSENS (to use a current controlled current source, a dummy independent voltage source is often placed into the path of the controlling current). The defining equation is:

$$I(F1) = 5 \cdot I(VSENS)$$

The current gain is 5, the maximum current flow through F1 is 3 A, and the minimum current flow is -3 A. If I(VSENS) = 2 A, I(F1) would be set to 3 amps and not 10 amps as would be suggested by the equation. A user-defined parameter may be specified for the polynomial coefficient(s), as shown below.

```
.PARAM VU = 5
F1 13 5 VSENS MAX=+3 MIN=-3 VU
```

The second example describes a current controlled current source with the value:

$$I(F2)=1e\text{-}3 + 1.3e\text{-}3 \cdot I(VCC)$$

Current flow is from the positive node through the source to the negative node. The direction of positive controlling current flow is from the positive node through the source to the negative node of vnam (linear), or to the negative node of each voltage source (nonlinear).

The third example is a delayed current controlled current source. The fourth example is a piecewise linear current controlled current source.

Dependent Voltage Sources — E Elements

Voltage Controlled Voltage Source (VCVS)

The syntax is:

Linear

```
Exxx n+ n- <VCVS> in+ in- gain  <MAX=val> <MIN=val> <SCALE=val>
<TC1=val>
+         <TC2=val><ABS=1> <IC=val>
```

Polynomial

```
Exxx  n+ n-  <VCVS>  POLY(ndim)  in1+ in1- ...  inndim+  inndim-<TC1=val>
<TC2=val>
+         <SCALE=val><MAX=val><MIN=val> <ABS=1> p0 <p1…> <IC=vals>
```

Piecewise Linear

```
Exxx  n+ n- <VCVS>  PWL(1) in+ in-    <DELTA=val> <SCALE=val> <TC1=val>
<TC2=val>
+         x1,y1 x2,y2  ...   x100,y100  <IC=val>
```

Multi-Input Gates

```
Exxx  n+ n- <VCVS>  gatetype(k) in1+ in1- ... inj+  inj-  <DELTA=val> <TC1=val>
<TC2=val>
+         <SCALE=val>  x1,y1   ...   x100,y100 <IC=val>
```

Delay Element

```
Exxx n+ n- <VCVS> DELAY in+ in- TD=val <SCALE=val> <TC1=val> <TC2=val>
+         <NPDELAY=val>
```

Behavioral Voltage Source

The syntax is:

```
Exxx n+ n- VOL=’equation’ in+ in- <MAX>=val> <MIN=val>
```

Ideal Op-Amp

The syntax is:

```
Exxx  n+ n-  OPAMP  in+ in-
```

Ideal Transformer

The syntax is:

```
Exxx n+ n- TRANSFORMER in+ in- k
```

E Element Parameters

ABS Output is absolute value if ABS=1.

DELAY keyword for the delay element. The delay element is the same as voltage controlled voltage source, except it is associated by a propagation delay TD.This element facilitates the adjustment of propagation delay in the subcircuit modeling process. DELAY is an Star-Hspice keyword and should not be used as a node name.

DELTA used to control the curvature of the piecewise linear corners. Defaults to 1/4 of the smallest of the distances between breakpoints. The maximum is 1/2 of the smallest of the distances between breakpoints.

Exxx voltage controlled element name. Must begin with "E", which may be followed by up to 15 alphanumeric characters.

gain voltage gain

gatetype(k) may be AND, NAND, OR, or NOR. The value of k is the number of inputs of the gate. The x and y terms represent the piecewise linear variation of output as a function of input. In the multi-input gates only one input determines the state of the output.

IC initial condition: the initial estimate of the value(s) of the controlling voltage(s). Default=0.0.

in +/- positive or negative controlling nodes. Specify one pair for each dimension.

j ideal transformer turn ratio: V(in+,in-) = $j \cdot$ V(n+,n-)

MAX maximum output voltage value. The default is undefined, and sets no maximum value.

MIN minimum output voltage value. The default is undefined, and sets no minimum value.

n+/- positive or negative node of controlled element

NPDELAY sets the number of data points to be used in delay simulations. The default value is the larger of 10 or the smaller of TD/tstep and tstop/tstep. That is,

$$NPDELAY_{default} = max\left[\frac{min\langle TD, tstop\rangle}{tstep}, 10\right]$$

The values of tstep and tstop are specified in the .TRAN statement.

OPAMP the keyword for ideal op-amp element. OPAMP is an Star-Hspice keyword and should not be used as a node name.

p0, p1 … the polynomial coefficients. When one coefficient is specified, Star-Hspice assumes it to be p1, with p0=0.0, and the element is linear. When more than one polynomial coefficient is specified by p0, p1, p2, ..., the element is nonlinear. See “Polynomial Functions” on page -14.

POLY polynomial dimension. If POLY(ndim) is not specified, a one-dimensional polynomial is assumed. Ndim must be a positive number.

PWL piecewise linear function keyword

SCALE element value multiplier

TC*1*,*TC2* first and second order temperature coefficients. The SCALE is updated by temperature:

$$\mathrm{SCALEeff} = \mathrm{SCALE} \cdot (1 + TC1 \cdot \Delta t + TC2 \cdot \Delta t^2)$$

TD time delay keyword

TRANS- the keyword for ideal transformer. TRANSFORMER is an Star-Hspice keyword
FORMER and should not be used as a node name.

VCVS the keyword for voltage controlled voltage source. VCVS is an Star-Hspice keyword and should not be used as a node name.

x1,... controlling voltage across nodes in+ and in-. The x values must be in increasing order.

y1,... corresponding element values of x

Examples

Ideal Op-Amp

A voltage amplifier with supply limits can be built with the voltage controlled voltage source. The output voltage across nodes 2,3 = v(14,1) * 2. The voltage gain parameter, 2, is also given. The MAX and MIN parameters specify a maximum E1 voltage of 5V and a minimum E1 voltage output of -5V. If, for instance, V(14,1) = -4V, E1 would be set to -5V and not -8V as the equation would produce.

```
Eopamp 2 3 14 1 MAX=+5 MIN=-5  2.0
```

A user-defined parameter may be used in the following format to specify a value for polynomial coefficient parameters:

```
.PARAM CU = 2.0
E1 2 3 14 1 MAX=+5 MIN=-5  CU
```

Voltage Summer

An ideal voltage summer specifies the source voltage as a function of three controlling voltage(s): V(13,0), V(15,0) and V(17,0). It describes a voltage source with the value:

V(13,0) + *V*(15,0) + *V*(17,0)

This example represents an ideal voltage summer. The three controlling voltages are initialized for a DC operating point analysis to 1.5, 2.0, and 17.25 V, respectively.

```
EX 17 0 POLY(3) 13 0 15 0 17 0 0 1 1 1 IC=1.5,2.0,17.25
```

Polynomial Function

The voltage controlled source can also output a nonlinear function using the one-dimensional polynomial. Since the POLY parameter is not specified, a one-dimensional polynomial is assumed, i.e., a function of one controlling voltage. The equation corresponds to the element syntax. Behavioral equations replace this older method.

$$E2\ (3,4) = 10.5 + 2.1\ {*}V(21,17) + 1.75\ {*}V(21,17)^2$$

```
E2 3 4 POLY
21 17 10.5 2.1 1.75
```

Zero Delay Inverter Gate

A simple inverter with no delay can be built with a piecewise linear transfer function.

```
Einv out 0 PWL(1) in 0 .7v,5v 1v,0v
```

Ideal Transformer

With the turn ratio 10 to 1, the voltage relationship is V(out)=V(in)/10.

```
Etrans out 0 TRANSFORMER in 0 10
```

Voltage Controlled Oscillator (VCO)

The keyword VOL is used to define a single-ended input that controls the output of a VCO.

In the following example, the frequency of the sinusoidal output voltage at node “out” is controlled by the voltage at node “control”. Parameter “v0” is the DC offset voltage and “gain” is the amplitude. The output is a sinusoidal voltage with a frequency of freq∗control.

```
Evco out 0 VOL=’v0+gain∗SIN(6.28∗freq∗v(control)∗TIME)’
```

Dependent Voltage Sources – H Elements

Current Controlled Voltage Source (CCVS)

The syntax is:

Linear

```
Hxxx n+  n-  <CCVS> vn1  transresistance  <MAX=val> <MIN=val>
<SCALE=val> <TC1=val>
+         <TC2=val> <ABS=1> <IC=val>
```

Polynomial

```
Hxxx n+ n-  <CCVS>  POLY(ndim)  vn1  <... vnndim> <MAX=val>MIN=val>
<TC1=val>
+         <TC2=val> <SCALE=val> <ABS=1> p0 <p1…> <IC=vals>
```

Piecewise Linear

```
Hxxx n+  n-  <CCVS>  PWL(1)  vn1 <DELTA=val> <SCALE=val> <TC1=val>
<TC2=val>
+         x1,y1   ...   x100,y100 <IC=val>
```

Multi-Input Gates

```
Hxxx  n+  n- gatetype(k) vn1, ... vnk  <DELTA=val> <SCALE=val> <TC1=val>
<TC2=val>
+         x1,y1  ...   x100,y100 <IC=val>
```

Delay Element

```
Hxxx n+ n-  <CCVS>  DELAY vn1  TD=val <SCALE=val> <TC1=val> <TC2=val>
+         <NPDELAY=val>
```

Note: E elements with algebraics make CCVS elements obsolete. However, CCVS elements may still be used for the sake of backward compatibility.

H Element Parameters

ABS Output is absolute value if ABS=1.

CCVS the keyword for current controlled voltage source. CCVS is an Star-Hspice keyword and should not be used as a node name.

DELAY keyword for the delay element. The delay element is the same as a current controlled voltage source except it is associated by a propagation delay TD.This element facilitates the adjustment of propagation delay in the subcircuit model process. DELAY is an Star-Hspice keyword and should not be used as a node name.

DELTA used to control the curvature of the piecewise linear corners. Defaults to 1/4 of the smallest of the distances between breakpoints. The maximum is 1/2 of the smallest of the distances between breakpoints.

gatetype(k) may be AND, NAND, OR, or NOR. The value of k is the number of inputs of the gate. The x and y terms represent the piecewise linear variation of output as a function of input. In the multi-input gates only one input determines the state of the output.

Hxxx current controlled voltage source element name. Must begin with “H”, which may be followed by up to 15 alphanumeric characters.

IC initial condition. This is the initial estimate of the value(s) of the controlling current(s) in amps. Default=0.0.

MAX maximum voltage value. The default is undefined, which sets no maximum value.

MIN minimum voltage value. The default is undefined, which sets no minimum value.

n+/- positive or negative controlled source connecting nodes

NPDELAY sets the number of data points to be used in delay simulations. The default value is the larger of 10 or the smaller of TD/tstep and tstop/tstep. That is,

$$NPDELAY_{default} = max\left[\frac{min\langle TD, tstop\rangle}{tstep}, 10\right]$$

The values of tstep and tstop are specified in the .TRAN statement.

p0, p1 . . . the polynomial coefficients. When one coefficient is specified, Star-Hspice assumes it to be p1, with p0=0.0, and the element is linear. When more than one polynomial coefficient is specified by p0, p1, p2, ..., the element is nonlinear. See “Polynomial Functions” on page -14.

POLY polynomial dimension. If POLY(ndim) is not specified, a one-dimensional polynomial is assumed. Ndim must be a positive number.

PWL piecewise linear function keyword

SCALE element value multiplier

TC1,TC2 first and second order temperature coefficients. The SCALE is updated by temperature:

$$\mathrm{SCALEeff} = \mathrm{SCALE} \cdot (1 + TC1 \cdot \Delta t + TC2 \cdot \Delta t^{2})$$

TD time delay keyword

transresistance current to voltage conversion factor

vn1 … names of voltage sources through which the controlling current flows. One name must be specified for each dimension.

x1,... controlling current through vn1 source. The x values must be in increasing order.

y1,... corresponding output voltage values of x

Example

```
HX 20 10 VCUR MAX=+10 MIN=-10 1000
```

The example above selects a linear current controlled voltage source. The controlling current flows through the dependent voltage source called VCUR. The defining equation of the CCVS is:

$$HX = 1000 \cdot VCUR$$

The defining equation states that the voltage output of HX is 1000 times the value of current flowing through CUR. If the equation produces a value of HX greater than +10V or less than -10V, HX, because of the MAX= and MIN= parameters, would be set to either 10V or -10V, respectively. CUR is the name of the independent voltage source that the controlling current flows through. If the controlling current does not flow through an independent voltage source, a dummy independent voltage source must be inserted.

```
.PARAM CT=1000
HX 20 10 VCUR MAX=+10 MIN=-10 CT
HXY 13 20 POLY(2) VIN1 VIN2 0 0 0 0 1 IC=0.5, 1.3
```

The example above describes a dependent voltage source with the value:

$$V = I(VIN1) \cdot I(VIN2)$$

This two-dimensional polynomial equation specifies FA1=VIN1, FA2=VIN2, P0=0, P1=0, P2=0, P3=0, and P4=1. The controlling current for flowing through VIN1 is initialized at.5mA. For VIN2, the initial current is 1.3mA.

The direction of positive controlling current flow is from the positive node, through the source, to the negative node of vnam (linear). The polynomial (nonlinear) specifies the source voltage as a function of the controlling current(s).

Referencing Digital Files

The U element can reference digital input and digital output models. If Star-Star-Hspice is being used by Viewlogic's Viewsim mixed mode simulator, the digital input comes from Viewsim. The state information comes from a digital file if Star-Star-Hspice is being run in standalone mode. Digital outputs are handled in a similar fashion. In this digital file mode, the input file is '*<designname>.d2a*' and the output file is named '*<designname>.a2d*'.

A2D and D2A statements accept the "\" backslash character as a line-continuation character to allow more than 255 characters in a line. This is needed because the first line of a digital file, which contains the signal name list, is often longer than the maximum line length accepted by some text editors.

A digital D2A file must not have a blank first line. If the first line of a digital file is blank, Star-Star-Hspice issues an error message.

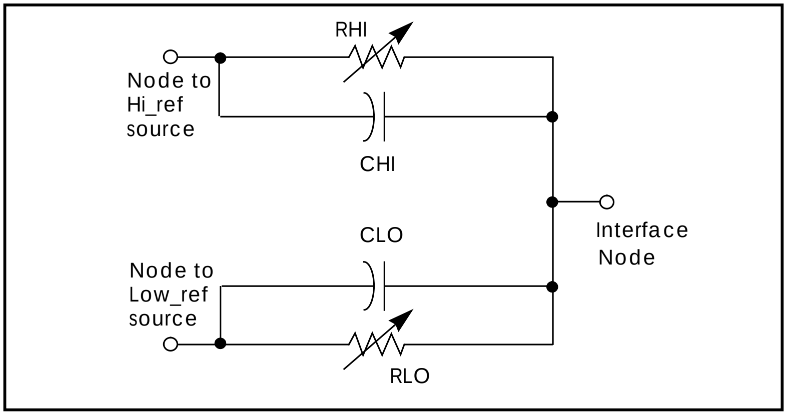


Figure 22-5: Digital- to-Analog Converter Element

Digital Input Element and Model

The syntax is:

U<name> <(interface) node> <(lo_ref)node> <(hi_ref)node> <(model) name>
+ SIGNAME = <(digital signal) name>] [IS = (initial state)]

Example

```
UC carry-in VLD2A VHD2A D2A SIGNAME=1 IS=0
VLD2A VLD2A 0 DC lo
VHD2A VHD2A 0 DC hi
```

The syntax is:

```
.MODEL <(model) name> DINPUT [(model parameters)]
```

Examples

```
.MODEL D2A U LEVEL=5 TIMESTEP=0.1NS,
+ S0NAME=0 S0TSW=1NS S0RLO = 15, S0RHI = 10K,
+ S2NAME=x S2TSW=3NS S2RLO = 1K, S2RHI = 1K
+ S3NAME=z S3TSW=5NS S3RLO = 1MEG,S3RHI = 1MEG
+ S4NAME=1 S4TSW=1NS S4RLO = 10K, S4RHI = 60
```

The following example demonstrates the use of the “\” line continuation character to format an input file for text editing. The file contains a signal list for a 64-bit bus.

```
...
a00 a01 a02 a03 a04 a05 a06 a07 \
a08 a09 a10 a11 a12 a13 a14 a15 \
...                              * Continuation of signal
names
a56 a57 a58 a59 a60 a61 a62 a63  * End of signal names
...                              * Remainder of file
```

Digital to Analog Input Model Parameters

Names(Alias)	Units	Default	Description
CLO	farad	0	capacitance to low level node

Names(Alias)	Units	Default	Description
CHI	farad	0	capacitance to high level node
SONAME			state “O” character abbreviation
SOTSW	sec		state “O” switching time
SORLO	ohm		state “O” resistance to low level node
SORHI	ohm		state “O” resistance to high level node
S1NAME			state “1” character abbreviation
S1TSW	sec		state “1” switching time
S1RLO	ohm		state “1”resistace to low level node
S1RHI	ohm		state “1” resistance to high level node
S2NAME			state “2” character abbreviation
S2TSW	sec		state “2” switching time
S2RLO	ohm		state “2” resistance to low level node
S2RHI	ohm		state “2” resistance to high level ode
S19NAME			state “19” character abbreviation
S19TSW	sec		state “19” switching time
S19RLO	ohm		state “19” resistance to low level node
S19RHI	ohm		state “19” resistance to high level node
TIMESTEP	sec		digital input file step size (digital files only)

Analog to Digital Output

General Form

U<name> <(interface)node> <(reference) node> <(model) name>

+ [SIGNAME = <(digital signal) name>]

Examples

```
vref VREFA2D 0 DC 0.0V
uco carry-out_2 VREFA2D a2d signame=12
```

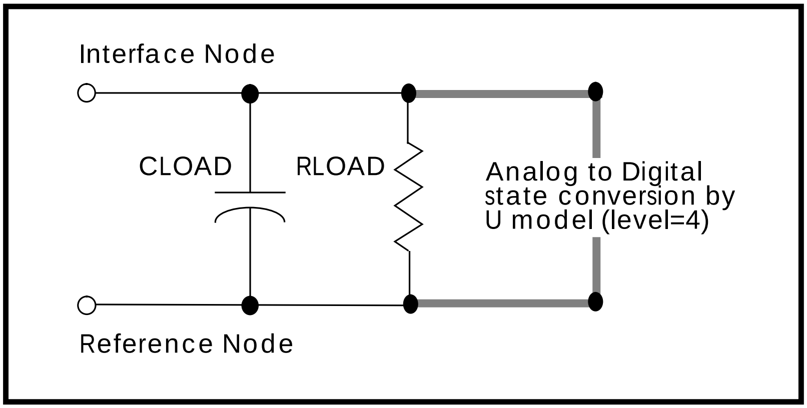


Figure 22-6: – Analog to Digital Converter Element

The syntax is:

.MODEL < name> U LEVEL=4 [(model parameters)]

Examples

```
* DEFAULT DIGITAL OUTPUT MODEL (no “X” value)
.MODEL A2D U LEVEL=4 TIMESTEP=0.1NS TIMESCALE=1
+ S0NAME=0 S0VLO=-1 S0VHI= 2.7
+ S4NAME=1 S4VLO= 1.4 S4VHI=9.0
+ CLOAD=0.05pf
```

Analog to Digital Output Model Parameters

Name(Alias)	Units	Default	Description
RLOAD	ohm	1/gmin	output resistor
CLOAD	farad	0	output capacitor
CHGONLY		0	0: write each timestep, 1: write upon change
SONAME			state “0” character abbreviation
SOVLO	volt		state “0” low level voltage
SOVHI	volt		state “0” high level voltage
S1NAME			state “1” character abbreviation
S1VLO	volt		state “1” low level voltage
S1VHI	volt		state “1” high level voltage
S2NAME			state “2” character abbreviation
S2VLO	volt		state “2” low level voltage
S2VHI	volt		state “2” high level voltage
S19NAME			state “19” character abbreviation
S19VLO	volt		state “19” low level voltage
S19VHI	volt		state “19” high level voltage
TIMESTEP	sec	1E-9	digital input file step-size

Two Bit Adder with Digital I/O

The following two bit MOS adder uses the digital input file. In the following plot, nodes ‘A[0], A[1], B[0], B[1], and CARRY-IN’ all come from a digital file input. SPICE outputs a digital file.

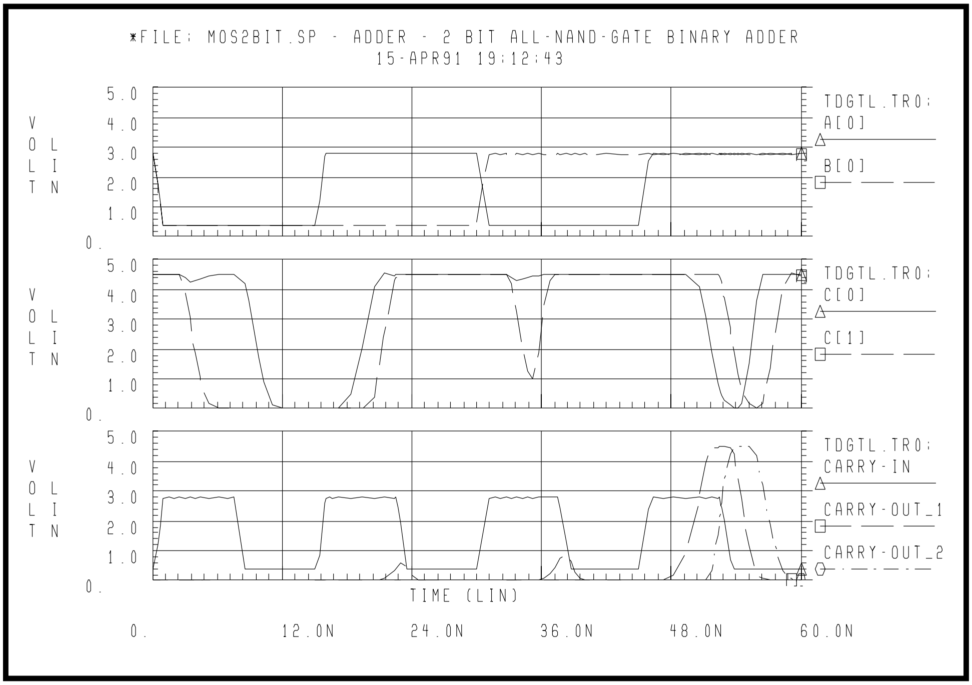
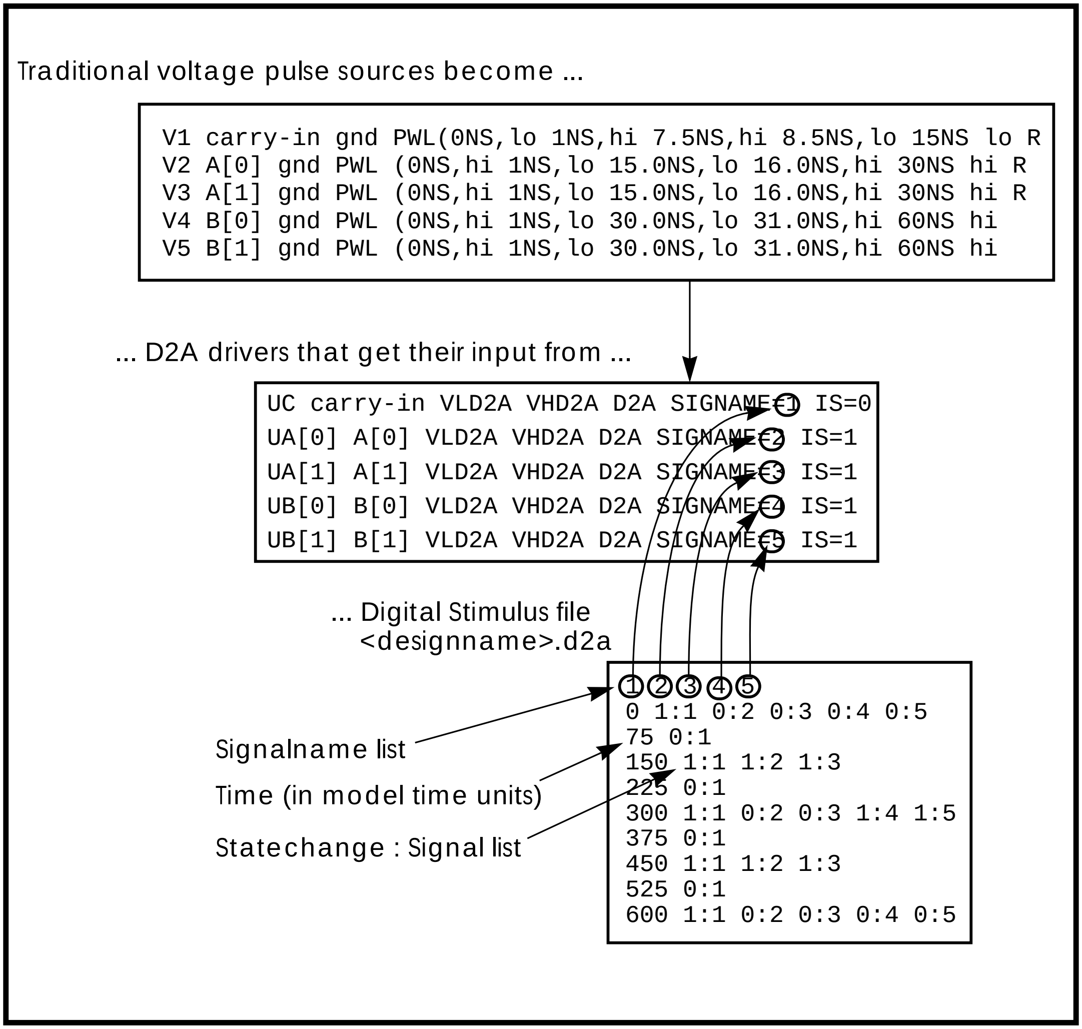


Figure 22-7: Digital Stimulus File Input

Replacing Sources With Digital Inputs


Traditional voltage pulse sources become ...
... D2A drivers that get their input from ...
... Digital Stimulus file
<designname>.d2a
Signalname list
Time (in model time units)
Statechange : Signal list

```
V1 carry-in gnd PWL(0NS,lo 1NS,hi 7.5NS,hi 8.5NS,lo 15NS lo R
V2 A[0] gnd PWL (0NS,hi 1NS,lo 15.0NS,lo 16.0NS,hi 30NS hi R
V3 A[1] gnd PWL (0NS,hi 1NS,lo 15.0NS,lo 16.0NS,hi 30NS hi R
V4 B[0] gnd PWL (0NS,hi 1NS,lo 30.0NS,lo 31.0NS,hi 60NS hi
V5 B[1] gnd PWL (0NS,hi 1NS,lo 30.0NS,lo 31.0NS,hi 60NS hi
```

```
UC carry-in VLD2A VHD2A D2A SIGNAME=1 IS=0
UA[0] A[0] VLD2A VHD2A D2A SIGNAME=2 IS=1
UA[1] A[1] VLD2A VHD2A D2A SIGNAME=3 IS=1
UB[0] B[0] VLD2A VHD2A D2A SIGNAME=4 IS=1
UB[1] B[1] VLD2A VHD2A D2A SIGNAME=5 IS=1
```

```
12345
0 1:1 0:2 0:3 0:4 0:5
75 0:1
150 1:1 1:2 1:3
225 0:1
300 1:1 0:2 0:3 1:4 1:5
375 0:1
450 1:1 1:2 1:3
525 0:1
600 1:1 0:2 0:3 0:4 0:5
```

Figure 22-8: Digital File Signal Correspondence

Example of MOS 2 Bit Adder

```
FILE: MOS2BIT.SP - ADDER - 2 BIT ALL-NAND-GATE BINARY ADDER
*
.OPTIONS ACCT NOMOD FAST scale=1u gmindc=100n post
.param lmin=1.25 hi=2.8v lo=.4v vdd=4.5
.global vdd

.TRAN .5NS 60NS
.MEAS PROP-DELAY TRIG V(carry-in) TD=10NS VAL=’vdd*.5’ RISE=1
+ TARG V(c[1]) TD=10NS VAL=’vdd*.5’ RISE=3
*
.MEAS PULSE-WIDTH TRIG V(carry-out_1) VAL=’vdd*.5’ RISE=1
+ TARG V(carry-out_1) VAL=’vdd*.5’ FALL=1
*
.MEAS FALL-TIME TRIG V(c[1]) TD=32NS VAL=’vdd*.9’ FALL=1
+ TARG V(c[1]) TD=32NS VAL=’vdd*.1’ FALL=1

VDD vdd gnd DC vdd
X1 A[0] B[0] carry-in C[0] carry-out_1 ONEBIT
X2 A[1] B[1] carry-out_1 C[1] carry-out_2 ONEBIT
```

Subcircuit Definitions

```
.subckt NAND in1 in2 out wp=10 wn=5
   M1 out in1 vdd vdd P W=wp L=lmin ad=0
   M2 out in2 vdd vdd P W=wp L=lmin ad=0
   M3 out in1 mid gnd N W=wn L=lmin as=0
   M4 mid in2 gnd gnd N W=wn L=lmin ad=0
   CLOAD out gnd ‘wp*5.7f’
.ends

.subckt ONEBIT in1 in2 carry-in out carry-out
   X1 in1 in2 #1_nand NAND
   X2 in1 #1_nand 8 NAND
   X3 in2 #1_nand 9 NAND
   X4 8 9 10 NAND
   X5 carry-in 10 half1 NAND
   X6 carry-in half1 half2 NAND
   X7 10 half1 13 NAND
   X8 half2 13 out NAND
   X9 half1 #1_nand carry-out NAND
.ENDS ONEBIT
```

Stimulus

```
UC carry-in VLD2A VHD2A D2A SIGNAME=1 IS=0
UA[0] A[0] VLD2A VHD2A D2A SIGNAME=2 IS=1
UA[1] A[1] VLD2A VHD2A D2A SIGNAME=3 IS=1
UB[0] B[0] VLD2A VHD2A D2A SIGNAME=4 IS=1
UB[1] B[1] VLD2A VHD2A D2A SIGNAME=5 IS=1
*
uc0 c[0] vrefa2d a2d signame=10
uc1 c[1] vrefa2d a2d signame=11
uco carry-out_2 vrefa2d a2d signame=12
uci carry-in vrefa2d a2d signame=13
```

Models

```
.MODEL N NMOS LEVEL=3 VTO=0.7 UO=500 KAPPA=.25 KP=30U
+ ETA=.01 THETA=.04 VMAX=2E5 NSUB=9E16 TOX=400 GAMMA=1.5
+ PB=0.6 JS=.1M XJ=0.5U LD=0.1U NFS=1E11 NSS=2E10
+ RSH=80 CJ=.3M MJ=0.5 CJSW=.1N MJSW=0.3
+ acm=2 capop=4
*
.MODEL P PMOS LEVEL=3 VTO=-0.8 UO=150 KAPPA=.25 KP=15U
+ ETA=.015 THETA=.04 VMAX=5E4 NSUB=1.8E16 TOX=400 GAMMA=.672
+ PB=0.6 JS=.1M XJ=0.5U LD=0.15U NFS=1E11 NSS=2E10
+ RSH=80 CJ=.3M MJ=0.5 CJSW=.1N MJSW=0.3
+ acm=2 capop=4
```

Default Digital Input Interface Model

```
.MODEL D2A U LEVEL=5 TIMESTEP=0.1NS,
+ S0NAME=0 S0TSW=1NS S0RLO = 15, S0RHI = 10K,
+ S2NAME=x S2TSW=5NS S2RLO = 1K, S2RHI = 1K
+ S3NAME=z S3TSW=5NS S3RLO = 1MEG,S3RHI = 1MEG
+ S4NAME=1 S4TSW=1NS S4RLO = 10K, S4RHI = 60
VLD2A VLD2A 0 DC lo
VHD2A VHD2A 0 DC hi
```

Default Digital Output Model (no “X” value)

```
.MODEL A2D U LEVEL=4 TIMESTEP=0.1NS TIMESCALE=1
+ S0NAME=0 S0VLO=-1 S0VHI= 2.7
+ S4NAME=1 S4VLO= 1.4 S4VHI=6.0
+ CLOAD=0.05pf
VREFA2D VREFA2D 0 DC 0.0V
.END
```

Modeling with Digital Behavioral Components

This section provides example of how to model with digital behavioral components.

Behavioral AND and NAND Gates

In this example, a two-input AND gate is modeled by a G element. A two-input NAND gate is modeled by an E element.

Example of AND/NAND Gates

```
behave.sp and/nand gates using g, e elements
.options post=2
.op
.tran .5n 20n
.probe v(in1) v(in2) v(andout) v(in1) v(in2) v(nandout)
g 0 andout and(2) in1 0 in2 0
+ 0.0 0.0ma
+ 0.5 0.1ma
+ 1.0 0.5ma
+ 4.0 4.5ma
+ 4.5 4.8ma
+ 5.0 5.0ma
*
e nandout 0 nand(2) in1 0 in2 0
+ 0.0 5.0v
+ 0.5 4.8v
+ 1.0 4.5v
+ 4.0 0.5v
+ 4.5 0.2v
+ 5.0 0.0v
*
vin1 in1 0 0 pwl(0,0 5ns,5)
vin2 in2 0 5 pwl(0,5 10ns,5 15ns,0)
rin1 in1 0 1k
rin2 in2 0 1k
rand andout 0 1k
rnand nandout 0 1k
.end
```

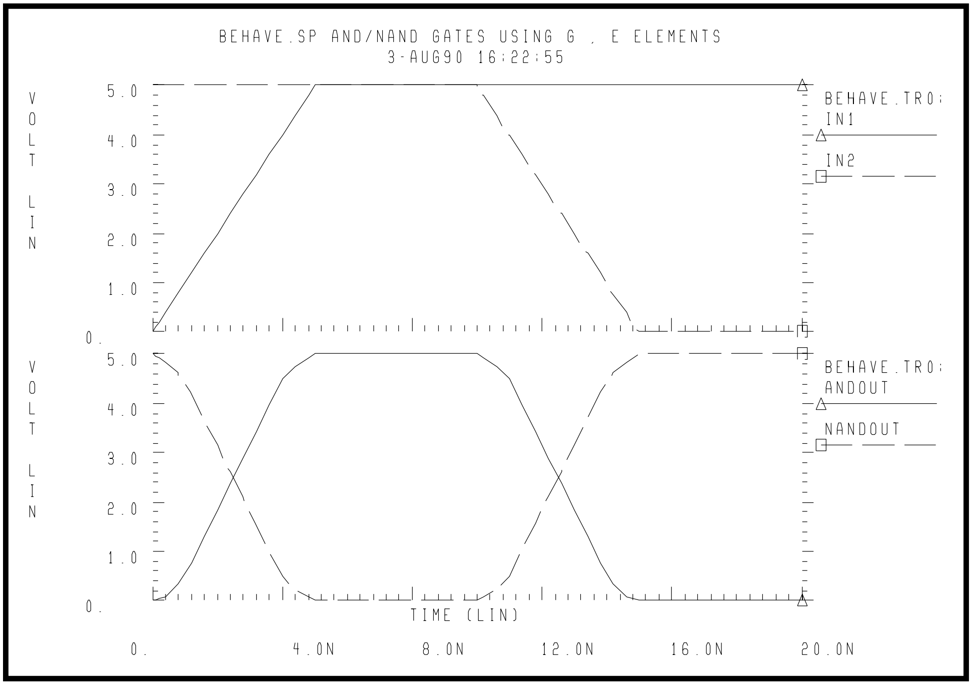


Figure 22-9: NAND/AND Gates

Behavioral D-Latch

In this example, a D flip-flop is modeled by one input NAND gates and NPWL/ PPWL functions.

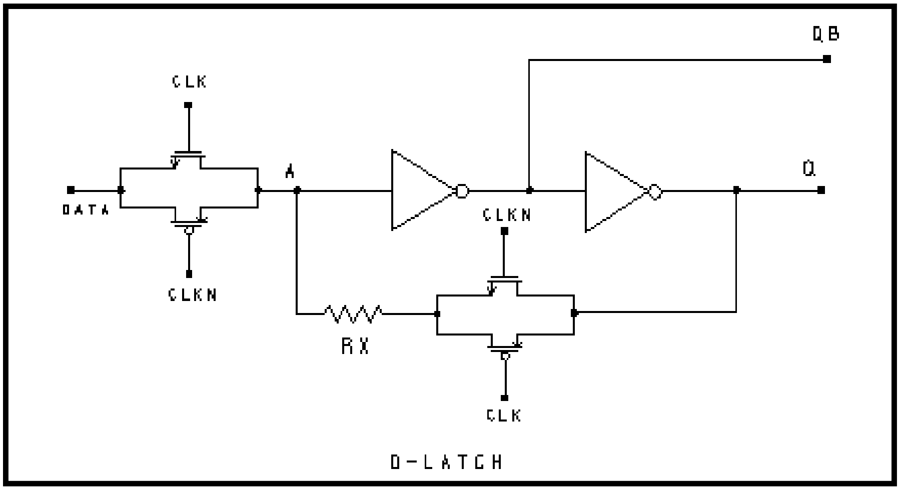


Figure 22-10: D-Latch Circuit

Example of a D-Latch

```
dlatch.sp--- cmos d-latch
.option post
.tran .2n 60ns
.probe tran clock=v(clck)data=v(d) q=v(q)
.ic v(q)=0
```

Waveforms

```
vdata d 0 pulse(0,5 2n,1n,1n 19n,40n)
vclk clck 0 pulse(0,5 7n,1n,1n 10n,20n)
vclkn clckn 0 pulse(5,0 7n,1n,1n 10n,20n)
xdlatch d clck clckn q qb dlatch cinv=.2p
```

Subcircuit Definitions for Behavioral N-Channel MOSFET

```
* DRAIN GATE SOURCE
.SUBCKT nmos 1 2 3 capm=.5p
cd 1 0 capm
cs 3 0 capm
gn 3 1 VCR NPWL(1) 2 3
+ 0. 495.8840G
+ 200.00000M 456.0938G
+ 400.00000M 141.6902G
+ 600.00000M 7.0624G
+ 800.00000M 258.9313X
+ 1.00000 6.4866X
+ 1.20000 842.9467K
+ 1.40000 321.6882K
+ 1.60000 170.8367K
+ 1.80000 106.4944K
+ 2.00000 72.7598K
+ 2.20000 52.4632K
+ 2.40000 38.5634K
+ 2.60000 8.8056K
+ 2.80000 5.2543K
+ 3.00000 4.3553K
+ 3.20000 3.8407K
+ 3.40000 3.4950K
+ 3.60000 3.2441K
+ 3.80000 3.0534K
+ 4.00000 2.9042K
+ 4.20000 2.7852K
+ 4.40000 2.6822K
+ 4.60000 2.5k
+ 5.0 2.3k
.ENDS nmos
```

Behavioral P-Channel MOSFET

```
* DRAIN GATE SOURCE
.SUBCKT pmos 1 2 3 capm=.5p
cd 1 0 capm
cs 3 0 capm
gp 1 3 VCR PPWL(1) 2 3
+ -5.0000 2.3845K
+ -4.8000 2.4733K
+ -4.6000 2.5719K
+ -4.4000 2.6813K
+ -4.2000 2.8035K
+ -4.0000 2.9415K
+ -3.8000 3.1116K
+ -3.6000 3.3221K
+ -3.4000 3.5895K
+ -3.2000 3.9410K
+ -3.0000 4.4288K
+ -2.8000 5.1745K
+ -2.6000 6.6041K
+ -2.4000 29.6203K
+ -2.2000 42.4517K
+ -2.0000 58.3239K
+ -1.8000 83.4296K
+ -1.6000 128.1517K
+ -1.4000 221.2640K
+ -1.2000 471.8433K
+ -1.0000 1.6359X
+ -800.00M 41.7023X
+ -600.00M 1.3394G
+ -400.00M 38.3449G
+ -200.00M 267.7325G
+ 0. 328.7122G
.ENDS pmos
*
.subckt tgate in out clk clkn ctg=.5p
xmn in clk out nmos capm=ctg
xmp in clkn out pmos capm=ctg
.ends tgate

.SUBCKT inv in out capout=1p
cout out 0 capout
rout out 0 1.0k
gn 0 out nand(1) in 0 scale=1
+ 0. 4.90ma
+ 0.25 4.88ma
+ 0.5 4.85ma
+ 1.0 4.75ma
+ 1.5 4.42ma
+ 3.5 1.00ma
+ 4.000 0.50ma
+ 4.5 0.2ma
+ 5.0 0.1ma
.ENDS inv

.subckt dlatch data clck clckn q qb cinv=1p
xtg1 data a clck clckn tgate ctg='cinv/2'
xtg2 q ax clckn clck tgate ctg='cinv/2'
rx ax a 5
xinv1 a qb inv capout=cinv
xinv2 qb q inv capout=cinv
.ends dlatch

.end
```

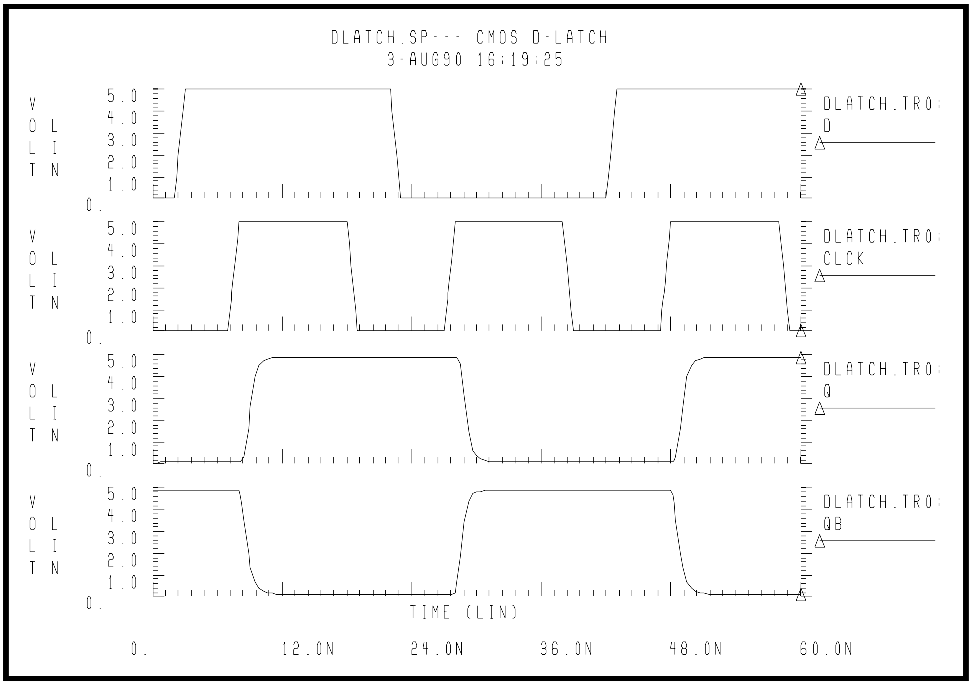


Figure 22-11: D-Latch Response

Behavioral Double-Edge Triggered Flip-Flop

In this example a double edged triggered flip-flop is modeled by using the D_LATCH subcircuit from previous example and several NAND gates.

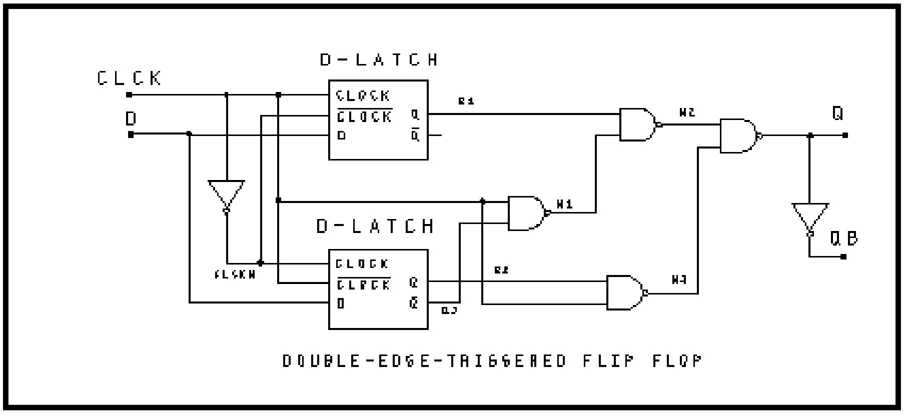


Figure 22-12: Double Edge Triggered Flip-Flop Schematic

Example of a Double Edge Triggered Flip-Flop

```
det_dff.sp--- double edge triggered flip-flop
.option post=2
.tran .2n 100ns
.probe tran clock=v(clck) data=v(d) q=v(q)
```

Waveforms

```
vdata d 0 pulse(0,5 2n,1n,1n 28n,50n)
vclk clck 0 pulse(0,5 7n,1n,1n 10n,20n)
```

Main Circuit

```
xclkn clck clckn inv cinv=.1p
xd1 d clck clckn q1 qb1 dlatch cinv=.2p
xd2 d clckn clck q2 qb2 dlatch cinv=.2p
xnand1 clck qb2 n1 nand2 capout=.5p
xnand2 q1 n1 n2 nand2 capout=.5p
xnand3 q2 clck n3 nand2 capout=.5p
xnand4 n2 n3 q nand2 capout=.5p
xinv q qb inv capout=.5p
```

Subcircuit Definitions

```
*Note: Subcircuit definitions for NMOS, PMOS, and INV are
given in the
* D-Latch examples; therefore they are not repeated here.
*
.SUBCKT nand2 in1 in2 out capout=2p
cout out 0 capout
rout out 0 1.0k
gn 0 out nand(2) in1 0 in2 0 scale=1
+ 0. 4.90ma
+ 0.25 4.88ma
+ 0.5 4.85ma
+ 1.0 4.75ma
+ 1.5 4.42ma
+ 3.5 1.00ma
+ 4.000 0.50ma
+ 4.5 0.2ma
+ 5.0 0.1ma
.ENDS nand2
*
.subckt dlatch data clck clckn q qb cinv=1p
xtg1 data a clck clckn tgate ctg='cinv/2'
xtg2 q ax clckn clck tgate ctg='cinv/2'
rx ax a 10
xinv1 a qb inv capout=cinv
xinv2 qb q inv capout=cinv
.ends dlatch
.end
```

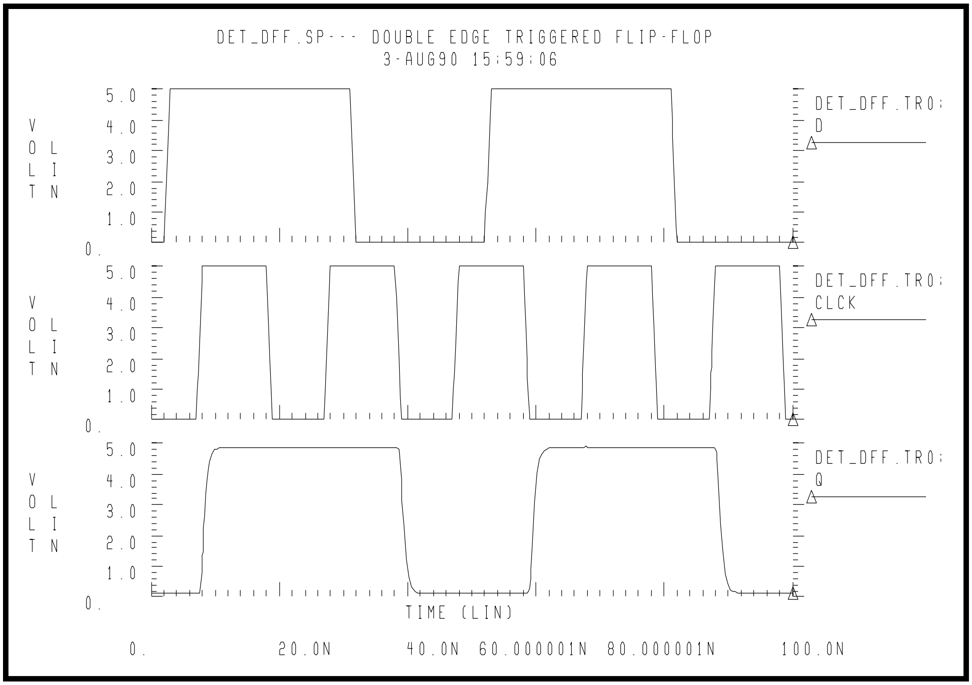


Figure 22-13: Double Edge Triggered Flip-Flop Response

Calibrating Digital Behavioral Components

This section describes how to calibrate with digital behavioral components.

Building Behavioral Lookup Tables

The following simulation demonstrates an ACL family output buffer with 2 ns delay and 1.8 ns rise and fall time. The ground and VDD supply currents and the internal ground bounce due to the package are also shown.

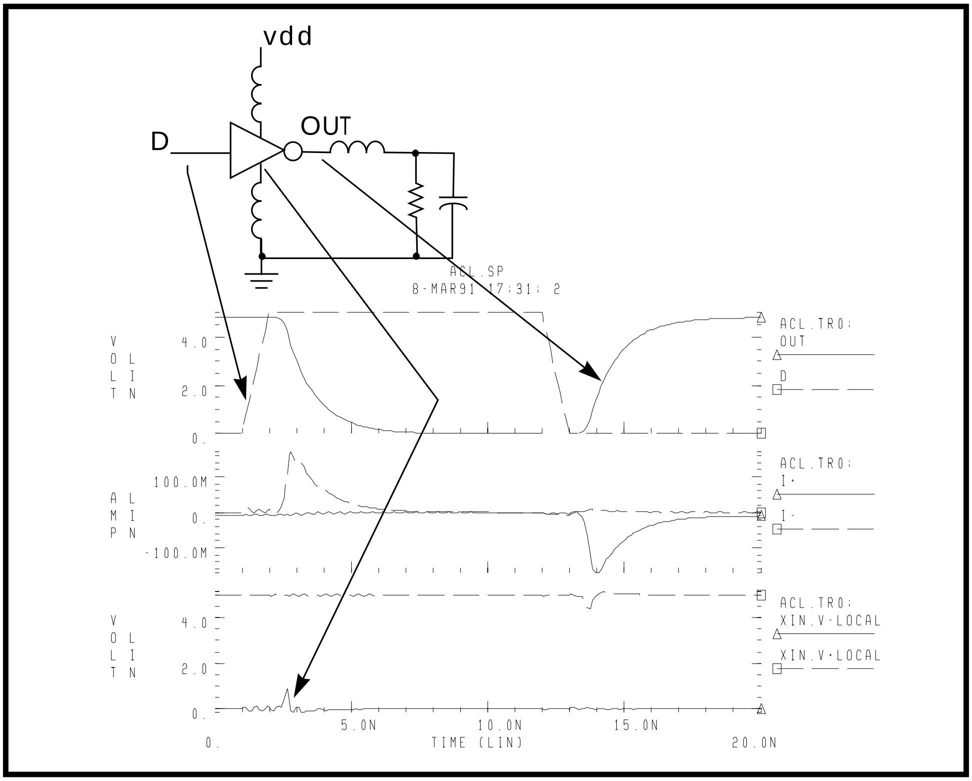


Figure 22-14: ACL Family Output Buffer

Star-Star-HspiceStar-Star-Hspice can automatically measure the datasheet quantities such as TPHL, risetime, maximum power dissipation, and ground bounce using the following commands.

```
.MEAS tphl trig v(D) val='.5*vdd' rise=1
+ targ v(out) val='.5*vdd' fall=1
.MEAS risetime trig v(out) val='.1*vdd' rise=1
+ targ v(out) val='.9*vdd' rise=1
.MEAS max_power max power
.MEAS bounce max v(xin.v_local)
```

The inverter is composed of capacitors, diodes, one-dimensional lookup table MOSFETs, and a special low-pass delay element. The low-pass delay element has the property that attenuates pulses that are narrower than the delay value.

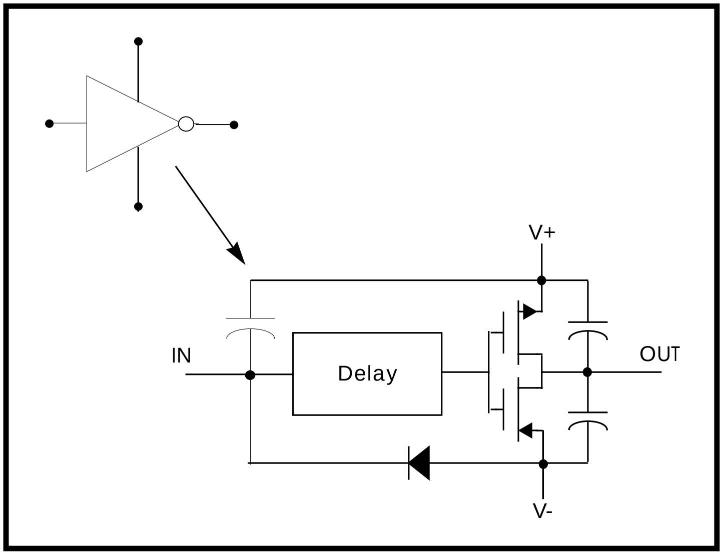


Figure 22-15: Inverter

Subcircuit Definition

```
.subckt inv in out v+ v-
cout+ out_l v+ 2p
cout- out_l v- 2p
xmp out_l inx v+ pmos
xmn out_l inx v- nmos
e inx v- delay in v- td=1n
din v- in dx
.model dx d cjo=2pf
chi in v+ .5pf
.ends inv
```

The behavioral MOSFETs are represented by one dimensional lookup tables. The equivalent n-channel lookup table is shown below.

Behavioral N-Channel MOSFET

Drain Gate Source

```
.subckt nmos 1 2 3
gn 3 1 VCR npwl(1) 2 3 scale=0.008
* VOLTAGE      RESISTANCE
+ 0.           495.8840g
+ 200.00000m   456.0938g
+ 400.00000m   141.6902g
+ 600.00000m   7.0624g
+ 800.00000m   258.9313meg
+ 1.00000      6.4866meg
+ 1.20000      842.9467k
+ 1.40000      21.6882k
+ 1.60000      170.8367k
+ 1.80000      106.4944k
+ 2.00000      72.7598k
+ 2.20000      52.4632k
+ 2.40000      38.5634k
+ 2.60000      8.8056k
+ 2.80000      5.2543k
+ 3.00000      4.3553k
+ 3.40000      3.4950k
+ 3.80000      2.0534k
+ 4.20000      2.7852k
+ 4.60000      2.5k
+ 5.0          2.3k
.ends nmos
```

The table above describes a voltage versus resistance table. It shows, for example, that the resistance at 5 volts is 2.3 kohm.

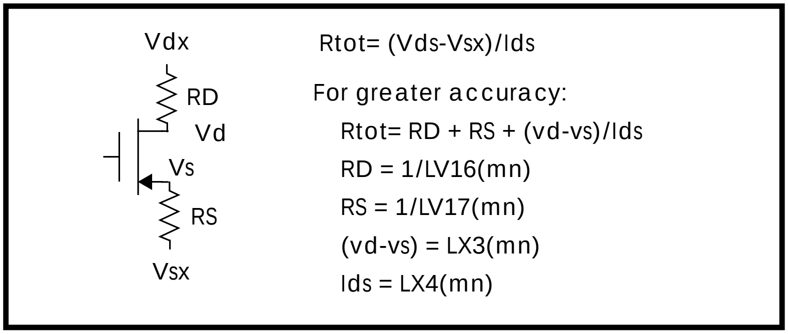
Creating a Behavioral Inverter Lookup Table

You can create an inverter lookup table in two simple steps. First simulate an actual transistor level inverter using a DC sweep of the input and print the output V/I for the output pullup and pulldown transistors. Next, copy the printed output into the volt controlled resistor lookup table element.

The following test file, *inv_vin_vout.sp* calculates RN (the effective pulldown resistor transfer function) and RP (the pullup transfer function).

RN is calculated as Vout/I(mn) where mn is the pulldown transistor. RP is calculated as (VCC-Vout)/I(mp) where mp is the pullup transfer function.

The actual calculation uses a more accurate way of obtaining the transistor series resistance as follows:



Rtot= (Vds-Vsx)/Ids

For greater accuracy:

Rtot= RD + RS + (vd-vs)/Ids

RD = 1/LV16(mn)

RS = 1/LV17(mn)

(vd-vs) = LX3(mn)

Ids = LX4(mn)

Figure 22-16: VIN versus VOUT

The first graph below shows VIN versus VOUT and the second graph shows the computed transfer resistances RP and RN as a function of VIN.

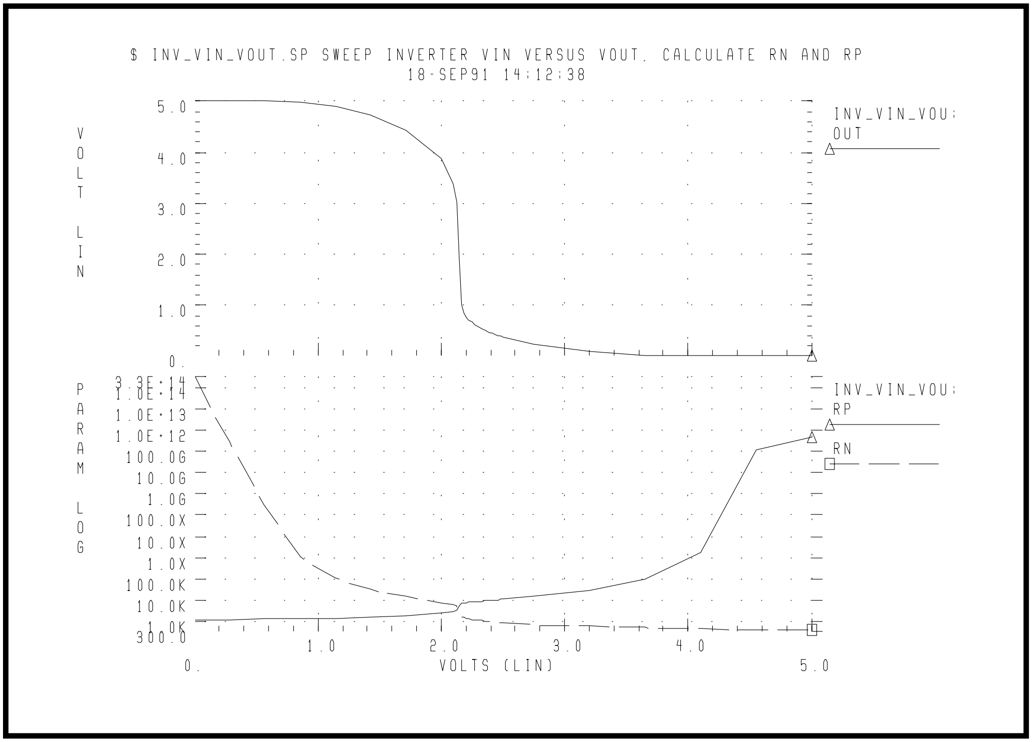


Figure 22-17: RP and RN as a Function of VIN

The Star-Hspice file used to calculate RP and RN is

```
$ inv_vin_vout.sp sweep inverter vin versus vout, calculate
rn and rp
```

The triple range DC sweep allows coarse grid before and after:

```
* use dc sweep with 3 ranges; 0-1.5v, 1.6-2.5, 2.6 5
.dc vin lin 8 0 2.0 lin 20 2.1 2.5 lin 6 2.75 5
$$ rn=par('v(out)/i(x1.mn)')
.print rn=
+ par('1/lv16(x1.mn)+1/lv17(x1.mn)+abs(lx3(x1.mn)/
lx4(x1.mn))')
.print rp=par('(-vcc+v(out))/i(x1.mp)')
.param sigma=0 vcc=5
.global vcc
vcc vcc 0 vcc
```

```
vin in 0 pwl 0,0 0.2n,5

x1 in out inv
.macro inv in out
mn out in 0 0 nch w=10u l=1u
mp out in vcc vcc pch w=10u l=1u
.eom
```

The tabular listing produced by Star-Hspice is

```
****** dc transfer curves tnom= 25.000 temp= 25.000
******
volt           rn

0.             3.312e+14
285.71429m     317.3503g
571.42857m     304.0682x
857.14286m     1.1222x
1.14286        107.6844k
1.42857        32.1373k
1.71429        14.6984k
2.00000        7.7108k
2.10000        5.8210k
2.12105        5.1059k
2.14211        3.2036k
2.16316        1.6906k
2.18421        1.4421k
2.20526        1.3255k
2.22632        1.2470k
2.24737        1.1860k
2.26842        1.1360k
2.28947        1.0935k
2.31053        1.0565k
2.33158        1.0238k
2.35263        994.3804
2.37368        967.7559
2.39474        943.4266
2.41579        921.0413
2.43684        900.3251
2.45789        881.0585
2.47895        863.0632
2.50000        846.1922
2.75000        701.5119
```

```
 3.20000        560.6908
 3.65000        479.8893
 4.10000        426.4486
 4.55000        387.7524
 5.00000        357.4228
```

Optimizing Behavioral CMOS Inverter Performance

Calibrate behavioral models by running Star-Hspice on the full transistor version of a cell and then optimizing the behavioral model to this data.

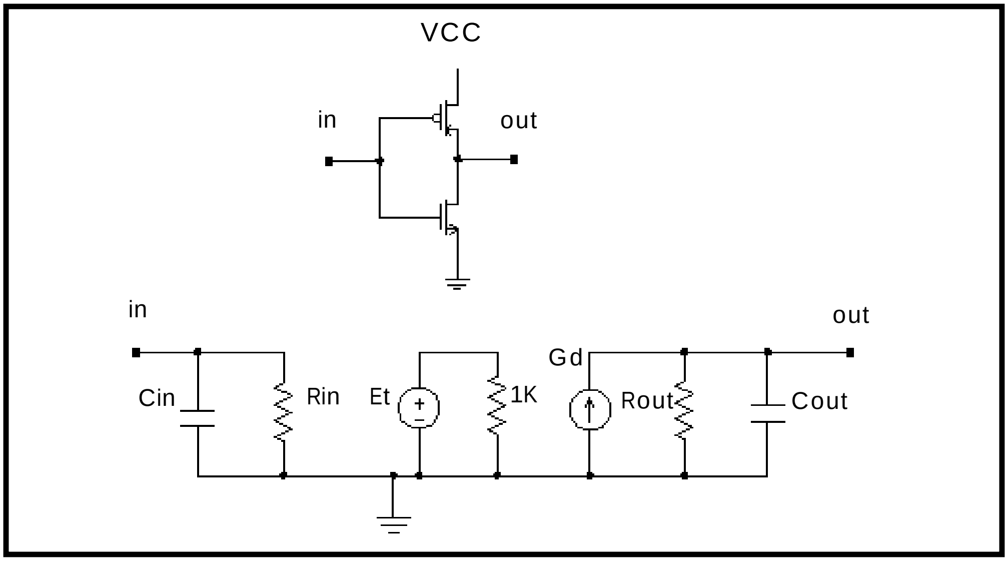


Figure 22-18: CMOS Inverter and its Equivalent Circuit

In this example, Star-Hspice simulates the CMOS inverter using the LEVEL 3 MOSFET model. The input and output resistances are obtained by performing a .TF transfer function analysis (.TF V(out) Vin). The transfer function table of the inverter is obtained by performing the .DC analysis sweeping input voltage (.DC Vin 0 5 .1). This table is then used in the PWL element to represent the transfer function of the inverter. The rise and fall time of the inverter in the equivalent circuit is adjusted by a voltage controlled PWL capacitance across the output resistance. The propagation delay is obtained by the delay element across the output rc circuit. The input capacitance is adjusted by using the inverter in a ring oscillator. All the adjustment in this example is done using the Star-Hspice optimization analysis. The data file and the results are shown below.

Example CMOS Inverter Subcircuit

```
INVB_OP.SP---OPTIMIZATION OF CMOS MACROMODEL INVERTER
.OPTIONS POST PROBE NOMOD METHOD=GEAR
.GLOBAL VCC VCCM
.PARAM VCC=5  ROUT=2.5K CAPIN=.5P
+ TDELAY=OPTINV(1.0N,.5N,3N)
+ CAPL=OPTINV(.2P,.1P,.6P)
+ CAPH=OPTINV(.2P,.1P,.6P)
.TRAN .25N 120NS
+ SWEEP OPTIMIZE=OPTINV RESULTS=RISEX,FALLX,PROPFX,PROPRX
+ MODEL=OPT1
.MODEL OPT1 OPT  ITROPT=30 RELIN=1.0E-5  RELOUT=1E-4
.MEAS TRAN PROPFM  TRIG V(INM)  VAL='.5*VCC' RISE=2
+                  TARG V(OUTM) VAL='.5*VCC' FALL=2
.MEAS TRAN PROPFX  TRIG V(IN)   VAL='.5*VCC' RISE=2
+                  TARG V(OUT)  VAL='.5*VCC' FALL=2
+                  GOAL='PROPFM' WEIGHT=0.8
.MEAS TRAN PROPRM  TRIG V(INM)  VAL='.5*VCC' FALL=2
+                  TARG V(OUTM) VAL='.5*VCC' RISE=2
.MEAS TRAN PROPRX  TRIG V(IN)   VAL='.5*VCC' FALL=2
+                  TARG V(OUT)  VAL='.5*VCC' RISE=2
+                  GOAL='PROPRM' WEIGHT=0.8
.MEAS TRAN FALLM  TRIG V(OUTM) VAL='.9*VCC' FALL=2
+                 TARG V(OUTM) VAL='.1*VCC' FALL=2
.MEAS TRAN FALLX  TRIG V(OUT)  VAL='.9*VCC' FALL=2
+                 TARG V(OUT)  VAL='.1*VCC' FALL=2
+                 GOAL='FALLM'
.MEAS TRAN RISEM  TRIG V(OUTM) VAL='.1*VCC' RISE=2
+                 TARG V(OUTM) VAL='.9*VCC' RISE=2
.MEAS TRAN RISEX  TRIG V(OUT)  VAL='.1*VCC' RISE=2
+                 TARG V(OUT)  VAL='.9*VCC' RISE=2
+                 GOAL='RISEM'
.TRAN 0.5N 120N
.PROBE V(out) V(outm)
VC VCC 0 VCC
VCCM VCCM 0 VCC
X1  IN  OUT  INV
X1M INM OUTM INVM
```

```
VIN   IN   GND PULSE(0,5 1N,5N,5N 20N,50N)
VINM  INM  GND PULSE(0,5 1N,5N,5N 20N,50N)
```

Subcircuit Definition

```
.SUBCKT INV IN OUT
RIN IN 0 1E12
CIN IN 0 CAPIN
ET  1 0  PWL(1)  IN 0
+    1.00000       5.0
+    1.50000       4.93
+    2.00000       4.72
+    2.40000       4.21
+    2.50000       3.77
+    2.60000       0.90
+    2.70000       0.65
+    3.00000       0.30
+    3.50000       0.092
+    4.00000       0.006
+    4.60000       0.
RT    1 0    1K
GD   0  OUT DELAY  1 0  TD=TDELAY   SCALE='1/ROUT'
GCOUT  OUT  0 VCCAP PWL(1) IN 0   1V,CAPL 2V,CAPH
ROUT  OUT  0   ROUT
.ENDS
```

Inverter Using Model

```
.SUBCKT INVM IN OUT
        XP1 OUT IN VCCM VCCM  MP
        XN1 OUT IN GND  GND   MN
.ENDS
.MODEL N NMOS LEVEL=3 TOX=850E-10 LD=.85U NSUB=2E16 VTO=1
+GAMMA=1.4 PHI=.9 UO=823  VMAX=2.7E5 XJ=0.9U KAPPA=1.6
+ETA=.1 THETA=.18 NFS=1.6E11 RSH=25 CJ=1.85E-4 MJ=.42 PB=.7
+CJSW=6.2E-10 MJSW=.34 CGSO=5.3E-10 CGDO=5.3E-10
+CGBO=1.75E-9

.MODEL P PMOS LEVEL=3 TOX=850E-10 LD=.6U
+NSUB=1.4E16 VTO=-.86 GAMMA=.65 PHI=.76 UO=266
+VMAX=.8E5 XJ=0.7U KAPPA=4 ETA=.25 THETA=.08 NFS=2.3E11
+RSH=85 CJ=1.78E-4 MJ=.4 PB=.6 CJSW=5E-10 MJSW=.22
 +CGSO=5.3E-10 CGDO=5.3E-10 CGBO=.98E-9
SUBCKT MP   1     2     3       4
```

```
M1 1 2 3 4 P W=45U L=5U AD=615P AS=615P
+PD=65U PS=65U NRD=.4 NRS=.4
.ENDS MP
.SUBCKT MN   1     2    3        4
M1 1 2 3 4 N W=17U L=5U AD=440P AS=440P
+PD=80U PS=80U NRD=.85 NRS=.85
.ENDS MN
.END
```

Result

```
OPTIMIZATION  RESULTS
   RESIDUAL SUM OF SQUARES     =  4.589123E-03
   NORM OF THE GRADIENT        =  1.155285E-04
   MARQUARDT SCALING PARAMETER =   130.602
   NO. OF FUNCTION EVALUATIONS =       51
   NO. OF ITERATIONS           =       15
OPTIMIZATION COMPLETED
MEASURED RESULTS < RELOUT=  1.0000E-04 ON LAST ITERATIONS
```

Optimized Parameters OPTINV

```
*                                    %NORM-SEN    %CHANGE
.PARAM TDELAY        =   1.3251N  $   37.6164   -48.6429U
.PARAM CAPL          = 390.2613F  $   37.2396    60.2596U
.PARAM CAPH          = 364.2716F  $   25.1440    62.1922U
```

Optimize Results Measure Names and Values

```
* RISEX          =   2.7018N
* FALLX          =   2.5388N
* PROPFX         =   2.0738N
* PROPRX         =   2.1107N
```

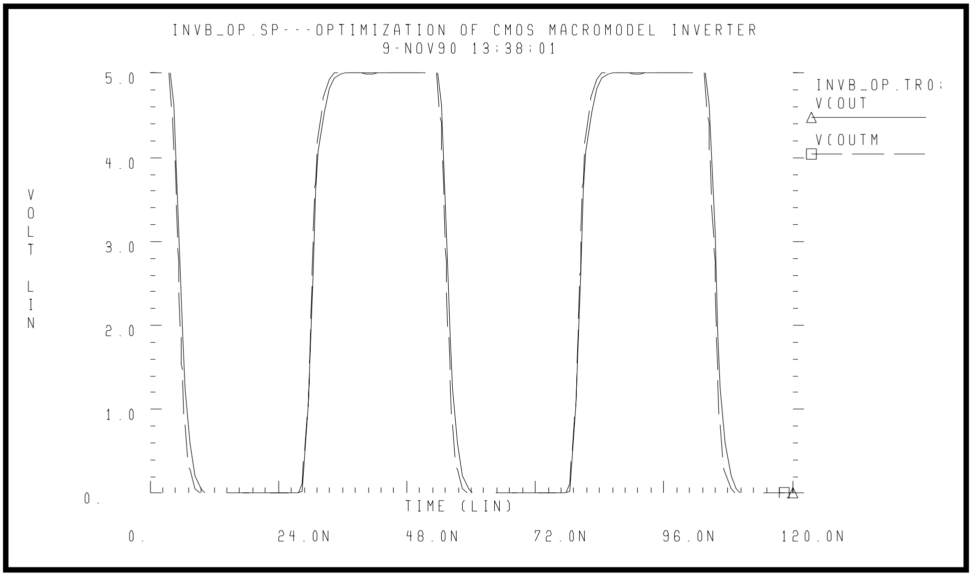


Figure 22-19: CMOS Inverter Response

Optimizing Behavioral Ring Oscillator Performance

To optimize behavioral ring oscillator performance, review the examples in this section.

Example Five-Stage Ring Oscillator

```
RING5BM.SP-5 STAGE RING OSCILLATOR--MACROMODEL CMOS INVERTER
.IC  V(IN)=5  V(OUT1)=0  V(OUT2)=5  V(OUT3)=0
.IC V(INM)=5 V(OUT1M)=0 V(OUT2M)=5 V(OUT3M)=0
.GLOBAL  VCCM
.OPTIONS NOMOD POST=2 PROBE  METHOD=GEAR DELMAX=0.5N
.PARAM VCC=5  $ CAPIN=0.92137P
.PARAM TDELAY=1.32N CAPL=390.26F CAPH=364.27F ROUT=2.5K
+ CAPIN=OPTOSC(0.8P,0.1P,1.0P)
.TRAN 1NS 150NS UIC
+ SWEEP OPTIMIZE=OPTOSC RESULTS=PERIODX MODEL=OPT1
.MODEL OPT1 OPT RELIN=1E-5 RELOUT=1E-4 DIFSIZ=.02 ITROPT=25
.MEAS TRAN PERIODM  TRIG V(OUT3M) VAL='.8*VCC' RISE=2
+                   TARG V(OUT3M) VAL='.8*VCC' RISE=3
.MEAS TRAN PERIODX  TRIG V(OUT3)  VAL='.8*VCC' RISE=2
```

```
+                   TARG V(OUT3)  VAL='.8*VCC' RISE=3
+                   GOAL='PERIODM'
.TRAN 1NS 150NS UIC
.PROBE V(OUT3) V(OUT3M)
X1  IN   OUT1  INV
X2  OUT1 OUT2  INV
X3  OUT2 OUT3  INV
X4  OUT3 OUT4  INV
X5  OUT4 IN    INV
CL  IN   0     1P
VCCM VCCM 0 VCC
X1M INM OUT1M INVM
X2M OUT1M OUT2M INVM
X3M OUT2M OUT3M INVM
X4M OUT3M OUT4M INVM
X5M OUT4M INM INVM
CLM INM 0 1P
*Subcircuit definitions given in the previous example are not
repeated here.
.END
```

Result

```
Optimization  Results
   RESIDUAL SUM OF SQUARES     =  4.704516E-10
   NORM OF THE GRADIENT        =  2.887249E-04
   MARQUARDT SCALING PARAMETER =   32.0000
   NO. OF FUNCTION EVALUATIONS =       52
   NO. OF ITERATIONS           =       20
OTIMIZATION COMPLETED
MEASURED RESULTS < RELOUT=  1.0000E-04 ON LAST ITERATIONS

**** OPTIMIZED PARAMETERS OPTOSC
*                                    %NORM-SEN    %CHANGE
.PARAM CAPIN         = 921.4155F  $  100.0000    8.5740U

*** OPTIMIZE RESULTS MEASURE NAMES AND VALUES
* PERIODX         =  40.3180N
```

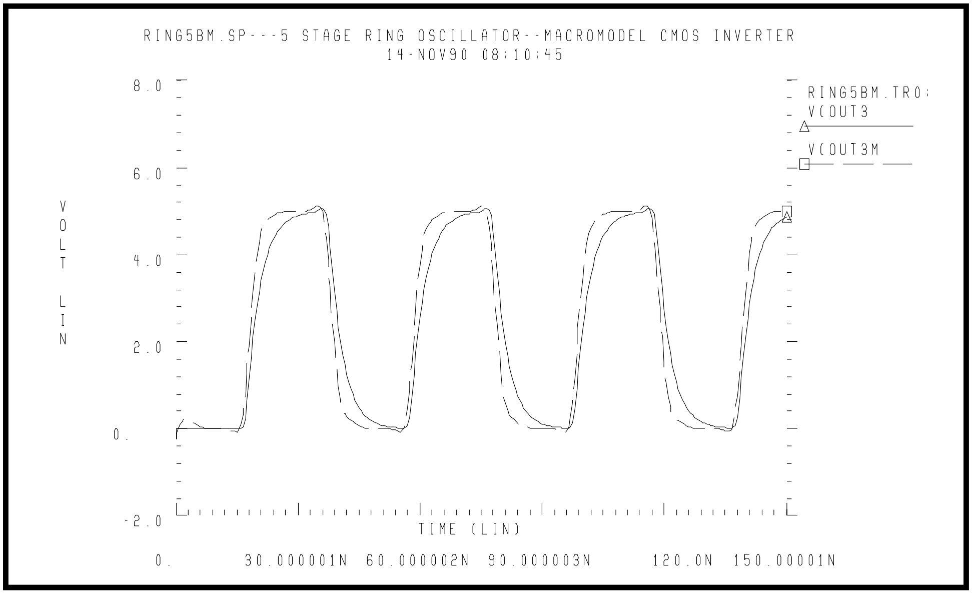


Figure 22-20: Ring Oscillator Response

Using Analog Behavioral Elements

The following components are examples of analog behavioral building blocks. Each demonstrates a basic Star-Hspice feature:

■ integrator	ideal op-amp E element source
■ differentiator	ideal op-amp E element source
■ ideal transformer	ideal transformer E element source
■ tunnel diode	lookup table G element source
■ silicon controlled rectifier	lookup table H element source
■ triode vacuum tube	algebraic G element source
■ AM modulator	algebraic G element source
■ data sampler	algebraic E element source

Behavioral Integrator

The integrator circuit is modelled by an ideal op-amp and uses a VCVS to adjust the output voltage. The output of integrator is given by:

$$Vout = -\frac{gain}{Ri \cdot Ci} \cdot \int_0^t Vin \cdot dt + Vout(0)$$

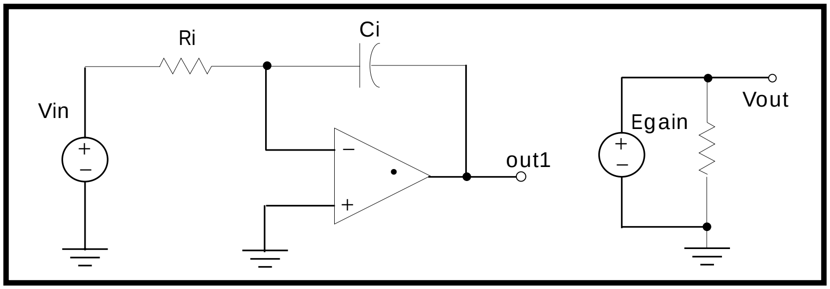


Figure 22-21: Integrator Circuit

Example of Integrator Circuit

```
Integ.sp  integrator circuit
```

Control and Options

```
.TRAN 1n 20n
.OPTIONS  POST PROBE  DELMAX =.1n
.PROBE Vin=V(in) Vout=V(out)
```

Subcircuit Definition

```
.SUBCKT integ  in out  gain=-1 rval=1k cval=1p
EOP  out1  0  OPAMP   in-  0
Ri in  in-  rval
Ci in- out1 cval
Egain out 0 out1 0 gain
Rout  out 0 1e12
.ENDS
```

Circuit

```
Xint in out  integ  gain=-0.4
Vin in  0  PWL(0,0 5n,5v 15n,-5v 20n,0)
.END
```

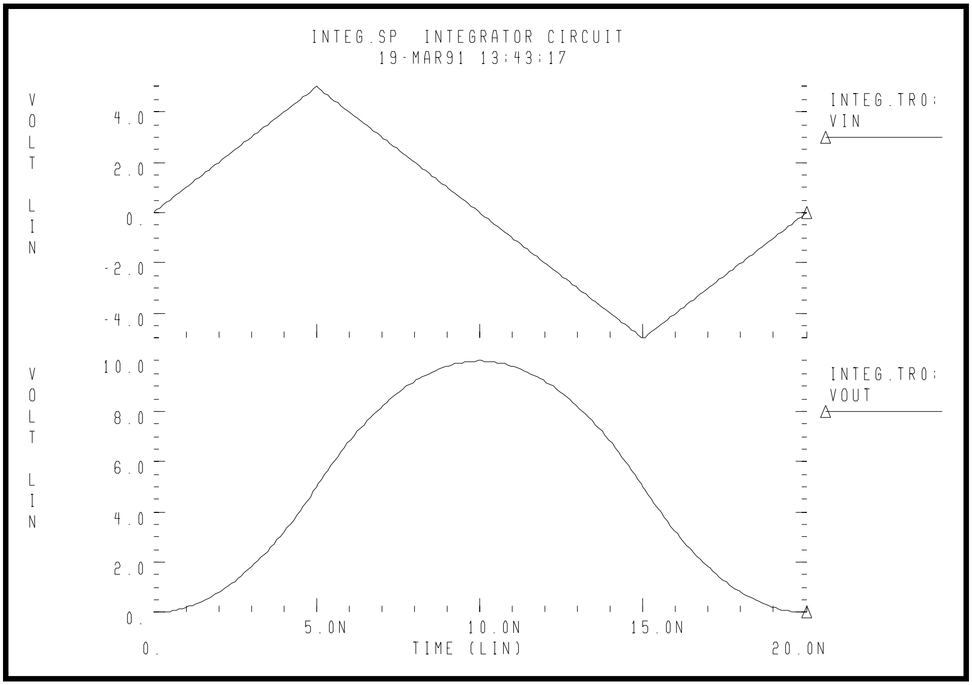


Figure 22-22: Response of Integrator to a Triangle Waveform

Behavioral Differentiator

A differentiator is modelled by an ideal op-amp and a VCVS for adjusting the magnitude and polarity of the output. The differentiator response is given by:

$$Vout = -gain \cdot Rd \cdot Cd \cdot \frac{d}{dt}Vin$$

For high frequency signal the output of a differentiator can have overshoot at the edges. You can smooth this out using a simple RC filter.

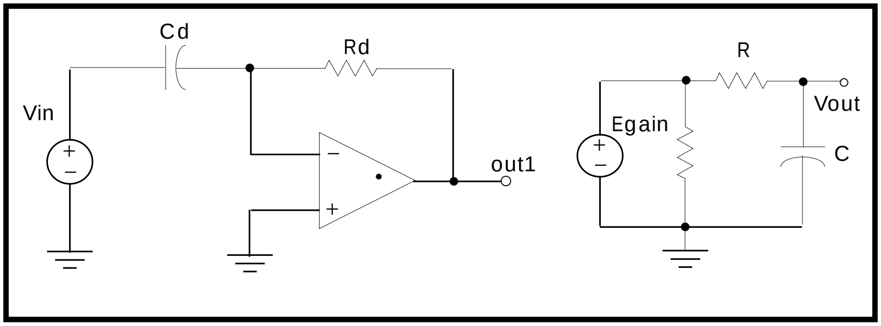


Figure 22-23: Differentiator Circuit

Example of a Differentiator Circuit

```
Diff.sp  differentiator circuit
* V(out)=Rval * Cval * gain * (dV(in)/dt)
```

Control and Options

```
.TRAN 1n 20n
.PROBE  Vin=V(in)  Vout=V(out)
.OPTIONS PROBE POST
```

Differentiator Subcircuit Definition

```
.SUBCKT diff in out  gain=-1 rval=1k cval=1pf
EOP  out1  0  OPAMP   in-  0
Cd in  in-  cval
Rd in- out1 rval
Egain out2 0 out1 0 gain
Rout  out2 0 1e12
*rc filter to smooth the output
R out2 out 75
C out  0   1pf
.ENDS
```

Circuit

```
Xdiff in out  diff  rval=5k
Vin in  0  PWL(0,0 5n,5v 15n,-5v 20n,0)

.END
```

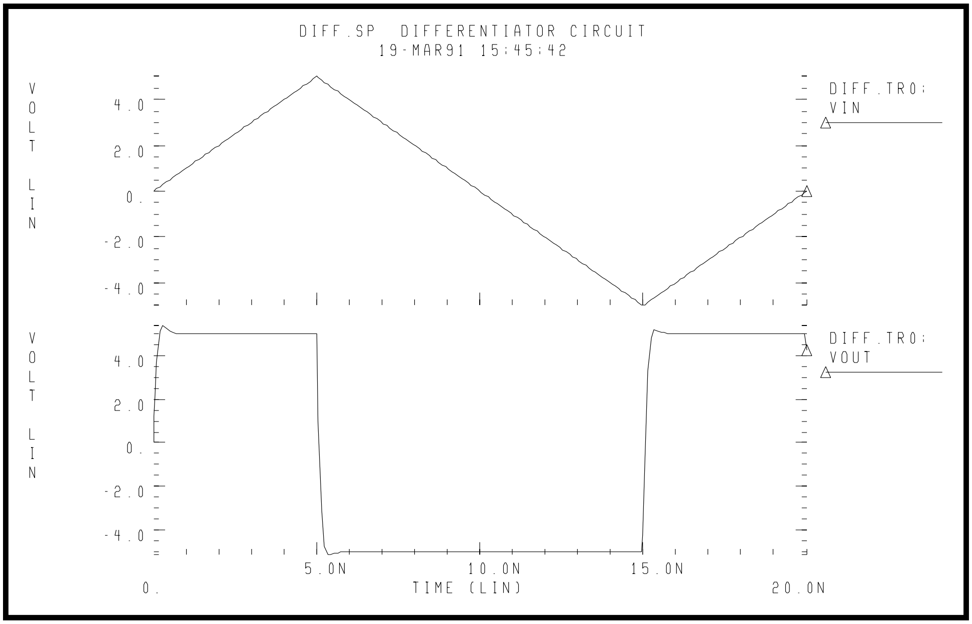


Figure 22-24: Response Of a Differentiator to a Triangle Waveform

Ideal Transformer

The following example uses the ideal transformer to convert 8 ohms impedance of a loudspeaker to 800 ohms impedance, which is a proper load value for a power amplifier, $Rin=n^2 \cdot RL$.

```
MATCHING IMPEDANCE BY USING IDEAL TRANSFORMER
E  OUT 0 TRANSFORMER  IN 0   10
RL OUT 0 8
VIN IN 0 1
.OP
.END
```

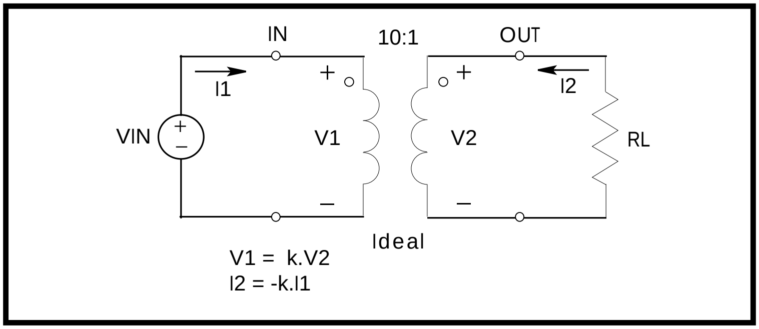


Figure 22-25: Ideal Transformer Example

Behavioral Tunnel Diode

In the following example, a tunnel diode is modeled by a PWL VCCS. The current characteristics are obtained for two DELTA values (50 µv and 10 µv). The IV characteristics corresponding to DELTA=10 µv have sharper corners. The derivative of current with respect to voltage (GD) is also displayed. The GD value around breakpoints changes in a linear fashion.

Example of Tunnel Diode

```
tunnel.sp-- modeling tunnel diode characteristic by pwl vccs
* pwl function is tested for two different delta values. The
* smaller delta will create the sharper corners.
.options post=2
vin 1 0 pvd
.dc pvd 0 550m 5m sweep delta poi 2 50mv 5mv
.probe dc id=lx0(g) gd=lx2(g)
g 1 0 pwl(1) 1 0 delta=delta
+ -50mv,-5ma 50mv,5ma 200mv,1ma 400mv,.05ma
+ 500mv,2ma 600mv,15ma
.end
```

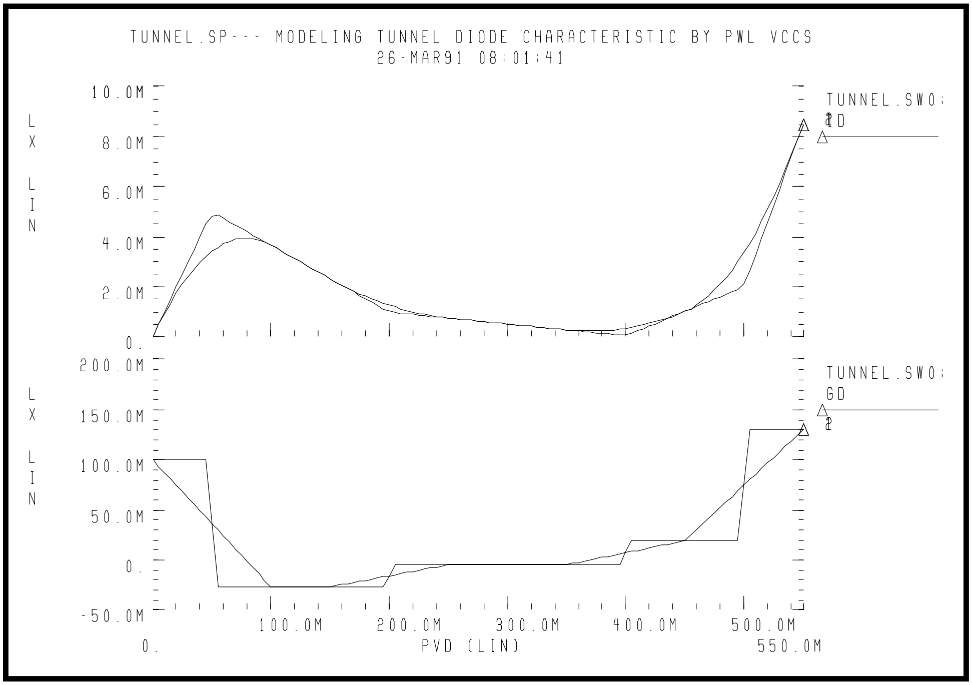


Figure 22-26: Tunnel Diode Characteristic

Behavioral Silicon Controlled Rectifier

The silicon controlled rectifier (SCR) characteristic can be easily modeled using a PWL CCVS because there is a unique voltage value for any current value.

Example of Silicon Controlled Rectifier (SCR)

```
pwl6.sp--- modeling SCR by pwl ccvs
*The silicon controlled rectifier (SCR) characteristic
*is modelled by a piecewise linear current controlled
*voltage source (PWL_CCVS), because for any current value
*there is a unique voltage value.
*
*use iscr as y-axis and v(1) as x-axis
*
.options post=2
```

```
iscr 0 2 0
vdum 2 1 0
.dc iscr 0 1u 1n
.probe vscr=lx0(h) transr=lx3(h)
h 1 0 pwl(1) vdum -5na,-2v 5na,2v 15na,.1v 1ua,.3v 10ua,.5
+ delta=5na
.end
```

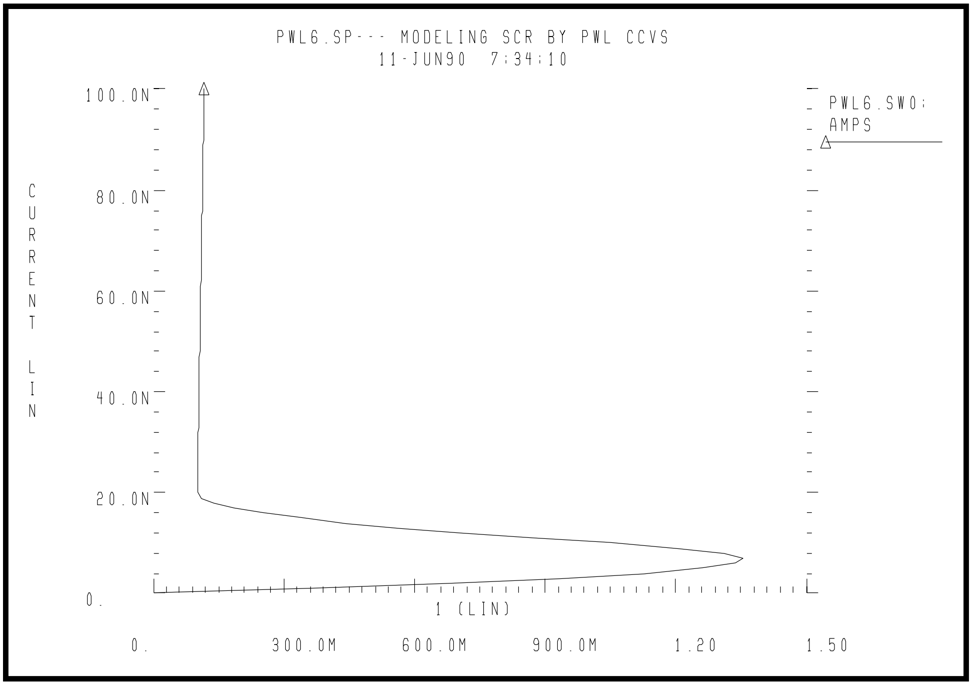


Figure 22-27: Silicon Controlled Rectifier

Behavioral Triode Vacuum Tube Subcircuit

The following example shows how to include the behavioral elements in a subcircuit to give very good triode tube action. The basic power law equation (current source *gt)* is modified by the voltage source *ea* to give better response in saturation.

Example Triode Model

```
triode.sp triode model family of curves using behavioral
elements
```

Control and Options

```
.options post acct
.dc va 20v 60v 1v vg 1v 10v 1v
.probe ianode=i(xt.ra) v(anode) v(grid) eqn=lv6(xt.gt)
.print v(xt.int_anode) v(xt.i_anode) inode=i(xt.ra)
eqn=lv6(xt.gt)
```

Circuit

```
vg grid 0 1v
va anode 0 20v
vc cathode 0 0v
xt anode grid cathode triode
```

Subcircuit Definition

```
.subckt triode anode grid cathode
* 5 ohm anode resistance
* ea creates saturation region conductance
ra anode i_anode 5
ea int_anode cathode pwl(1) i_anode cathode delta=.01
+ 20,0 27,.85 28,.95 29,.99 30,1 130,1.2
gt i_anode cathode
+
cur='20m*v(int_anode,cathode)*pwr(max(v(grid,cathode),0),1.5)'
cga grid i_anode 30p
cgc grid cathode 20p
cac i_anode cathode 5p
.ends
*
.end
```

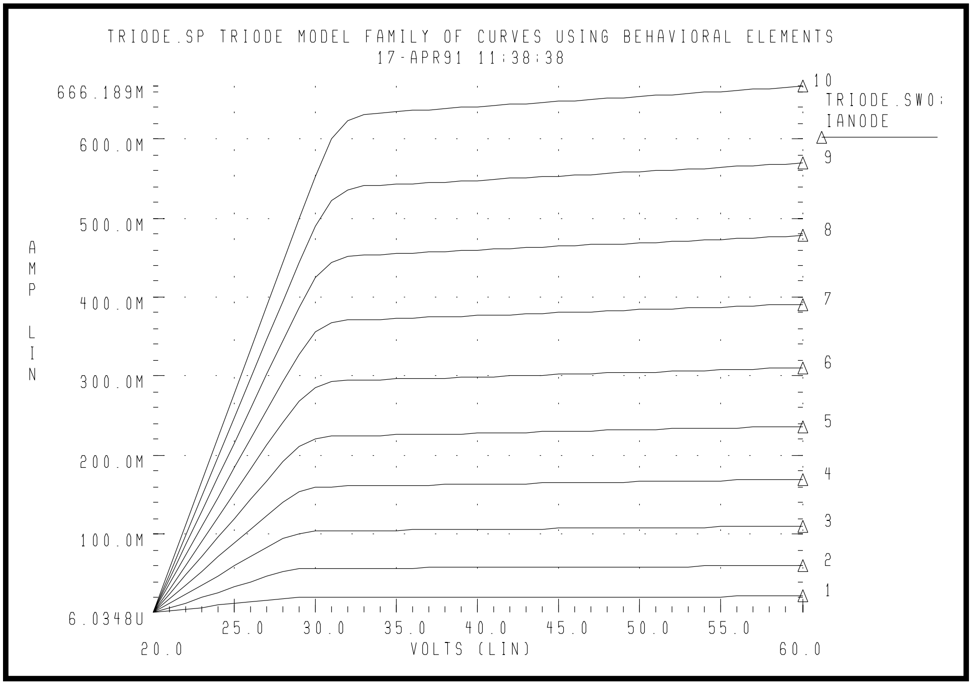


Figure 22-28: Triode Family of Curves

Behavioral Amplitude Modulator

This example uses a G element as an amplitude modulator with pulse waveform carrier.

Example of Amplitude Modulator

```
amp_mod.sp amplitude modulator with pulse waveform carrier
.OPTIONS POST
.TRAN .05m 40m
.PROBE V(1) V(2) V(3)
Vs 1 0 SIN(0,1,100)
Vc 2 0 PULSE(1,-1,0,1n,1n,.5m,1m)
Rs 1 0 1+
Rc 2 0 1
G 0 3 CUR='(1+.5*V(1))*V(2)'
Re 3 0 1
.END
```

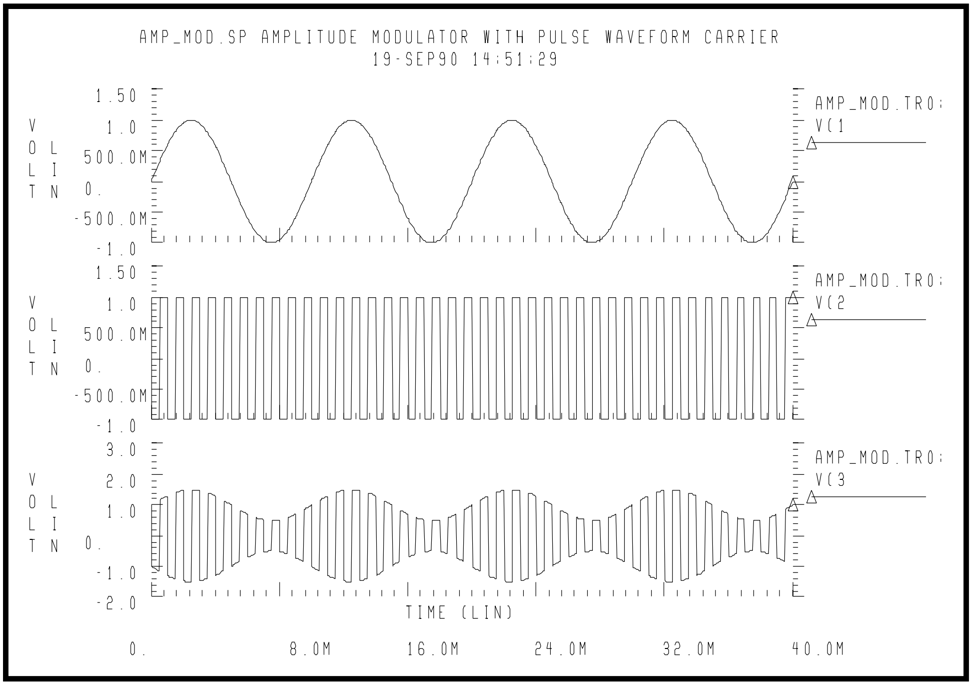


Figure 22-29: Amplitude Modulator Waveforms

Behavioral Data Sampler

A behavioral data sample follows.

Example Behavioral Sampling with E Element

```
sampling.sp sampling a sine wave.
.OPTIONS POST
.TRAN .05m 40m
.PROBE V(1) V(2) V(3)
Vc 1 0 SIN(0,5,100)
Vs 2 0 PULSE(0,1,0,1n,1n,.5m,1m)
Rc 1 0 1
Rs 2 0 1
E 3 0 VOL='V(1)*V(2)'
Re 3 0 1
.END
```

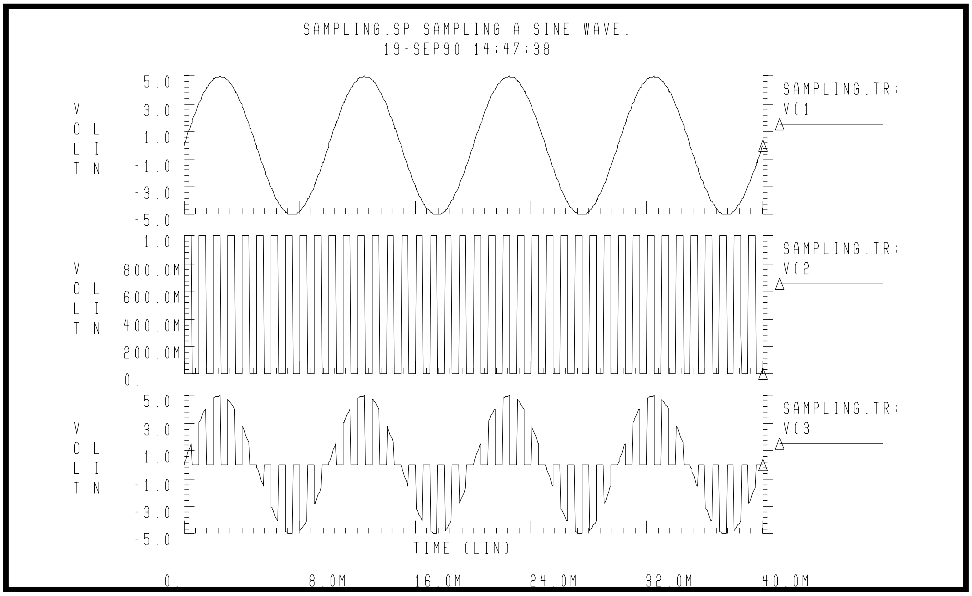


Figure 22-30: Sampled Data

Using Op-Amps, Comparators, and Oscillators

Thsi section describes the benefits of using Star-Hspice’s op-amps, comparators, and oscillators when performing simulation.

Star-Hspice Op-Amp Model Generator

Star-Hspice uses the model generator for the automatic design and simulation of both board level and IC op-amp designs. You can take the existing electrical specifications for a standard industrial operational amplifier, enter the specifications in the op-amp model statement, and Star-Hspice automatically generates the internal components of the op-amp to meet the specifications. You can then call the design from a library for a board level simulation.

The Star-Hspice op-amp model is a subcircuit that is about 20 times faster to simulate than an actual transistor level op-amp. You can adjust the AC gain and phase to within 20 percent of the actual measured values and set the transient slew rates accurately. This model does not contain high order frequency response poles and zeros and may significantly differ from actual amplifiers in predicting high frequency instabilities. Normal amplifier characteristics, including input offsets, small signal gain, and transient effects are represented in this model.

The op-amp subcircuit generator consists of two parts, a model and one or more elements. Each element is in the form of a subcircuit call. The model generates an output file of the op-amp equivalent circuit for collection in libraries. The file name is the name of the model (mname) with an *.inc* extension.

Once the output file is generated, other Star-Hspice input files may reference this subcircuit using a .SUBCKT call to the model name. The .SUBCKT call automatically searches the present directory for the file, then the directories specified in any .OPTION SEARCH =’directory_path_name’, and finally the directory where the DDL (Discrete Device Library) is located.

The amplifier element references the amplifier model.

Convergence

If DC convergence problems are encountered with op-amp models created by the model generator, use the .IC or .NODESET statement to set the input nodes to the voltage halfway between the VCC and VEE. This balances the input nodes and stabilizes the model.

Op-Amp Element Statement Format

COMP=0 (internal compensation)

The syntax is:

```
xa1 in- in+ out vcc vee modelname AV=val
```

COMP=1 (external compensation)

General form

```
xa1 in- in+ out comp1 comp2 vcc vee modelname AV=val
```

in-	the inverting input
in+	the noninverting input
out	the output, single ended
vcc	the positive supply
vee	the negative supply
modelname	the subcircuit reference name

Op-Amp .MODEL Statement Format

The syntax is:

```
.MODEL mname AMP parameter=value …
```

mname	model name. Elements reference the model by this name.
AMP	identifies an amplifier model
parameter	any model parameter described below
value	value assigned to a parameter

Example

```
 X0  IN-  IN+ OUT0  VCC VEE ALM124
 .MODEL ALM124 AMP
 +       C2=  30.00P     SRPOS=   .5MEG      SRNEG= .5MEG
 +       IB=  45N         IBOS=   3N           VOS=  4M
 +     FREQ=  1MEG      DELPHS=   25          CMRR=  85
 +     ROUT=  50            AV=   100K         ISC=  40M
 +    VOPOS=  14.5       VONEG=   -14.5        PWR=  142M
 +      VCC=  16           VEE=   -16         TEMP=  25.00
 +     PSRR=  100          DIS=   8.00E-16     JIS=  8.00E-16
```

Op-Amp Model Parameters

The model parameters for op-amps are shown below. The defaults for these parameters depend on the DEF parameter setting. Defaults for each of the three DEF settings are shown in the following table.

Names(Alias)	Units	Default	Description
AV (AVD)	volt/volt		amplifier gain in volts out per volt in. It is the DC ratio of the voltage in to the voltage out. Typical gains are from 25k to 250k. If the frequency comes out too low, try increasing the negative and positive slew rates or decreasing DELPHS.
AV1K	volt/volt		amplifier gain at 1 kilohertz. This is a convenient method of estimating the unity gain bandwidth. The gain can be expressed in actual voltage gain or in dB. Decibel is now a standard unit conversion for Star-Hspice. If AV1K is set, then FREQ is ignored. A typical value for AV1K is AV1K=(unity gain freq)/ 1000.

Names(Alias)	Units	Default	Description
C2	farad		internal feedback compensation capacitance. If the amplifier is internally compensated and no capacitance value is given, assume 30 pF. If the gain is high (above 500k), the internal compensation capacitor is probably different (typically 10 pF). If the amplifier is externally compensated, (COMP=1) set C2 to about 0.5 pF as the residual internal capacitance.
CMRR	volt/volt		common mode rejection ratio. This is usually between 80 and 110 dB. This can be entered as 100 dB or as 100000.

COMP			compensation level selector. This modifies the number of nodes in the equivalent to include external compensation nodes if set to one. See C2 for external compensation settings.
			COMP=0 internal compensation (Default)
			COMP=1 external compensation

DEF			default model selector. Allows choice of three default settings. 0= generic (0.6 MHz bandwidth) (Default) 1= ua741 (1.2 MHz bandwidth) 2= mc4560 (3 MHz bandwidth)

Names(Alias)	Units	Default	Description
DELPHS	degree s		excess phase at the unity gain frequency. Also called the phase margin. DELPHS is measured in degrees. Typical excess phases range from 5° to 50°. To determine DELPHS, subtract the phase at unity gain from 90°; this gives the phase margin. Use the same chart as used for the FREQ determination above. DELPHS interacts with FREQ (or AV1K). Values of DELPHS tend to lower the unity gain bandwidth, particularly values greater than 20°. The model does not have enough poles to always give correct phase and frequency response. It is usually best to pick the DELPHS closest to measured value that does not reduce unity gain bandwidth more than 20%.
DIS	amp	1e-16	diode and BJT saturation current
FREQ (GBW,*BW*)	Hz		unity gain frequency. Measured in hertz and typical frequencies range from 100 kHz to 3 MHz. If not specified, measure open loop frequency response at 0 dB voltage gain and the actual compensation capacitance. Typical compensation is 30 pF and single pole compensation configuration. If AV1K is greater than zero, the unity gain frequency is calculated from AV1K and FREQ is ignored.
IB	amp		input bias current. The amount of current required to bias the input differential transistors. This is generally a fundamental electrical characteristic. Typical values are between 20 and 400 nA.
IBOS	amp		input bias offset current, also called input offset current. This is the amount of unbalanced current between the input differential transistors. Generally a fundamental electrical characteristic. Typical values are 10% to 20% of the IB.

Names(Alias)	Units	Default	Description
ISC	amp		input short circuit current – not always specified. Typical values are between 5 and 25 mA. ISC can also be determined from output characteristics (current sinking) as the maximum output sink current. ISC and ROUT interact with each other, if ROUT is too large for a given value of ISC, ROUT is automatically reduced.
JIS	amp		JFET saturation current. Default=1e-16 and need not be changed.
LEVIN			input level type selector. Allows only BJT differential pair creation. LEVIN=1 BJT differential input stage.
LEVOUT			output level type selector. Allows only single-ended output stage creation. LEVOUT=1 single-ended output stage.
MANU			manufacturer's name. This can be added to the model parameter list to identify the source of the model parameters. The name is printed in the final equivalent circuit.
PWR (PD)	watt		total power dissipation value for the amplifier. This includes the calculated value for the op-amp input differential pair. If high slew rate and very low power is specified a warning is issued and the input differential pair alone gives the power dissipation.
RAC (r0ac, roac)	ohm		high frequency output resistance. This typically is about 60% of ROUT. RAC usually ranges between 40 to 70 ohms for op-amps with video drive capabilities.

Names(Alias)	Units	Default	Description
ROUT	ohm		low frequency output resistance. This can be determined using the closed loop output impedance graph. The impedance at about 1kHz, using the maximum gain, is close to ROUT. Gains of 1,000 and above show the effective DC impedance, generally in the frequency region between 1k and 10 kHz. Typical values for ROUT are 50 to 100 ohms.
SRNEG *(SRN)*	volt		negative going output slew rate. This is found from the graph of the voltage follower pulse response. This is generally a 4 or 5 volt output change with 10 to 20 volt supplies. Measures the negative going change in voltage and the amount of time for the change.
SRPOS (SRP)	volt		positive going output slew rate. This is found from the graph of the voltage follower pulse response. This is generally a 4 or 5 volt output change with 10 to 20 volt supplies. Measures the positive going change in voltage and the amount of time for the change. Typical slew rates are in the range of 70k to 700k.
TEMP	°C		temperature in degrees Celsius. This usually is set to the temperature at which the model parameters were measured, which typically is 25 °C.
VCC	volt		positive power supply reference voltage for VOPOS. The amplifier VOPOS was measured with respect to VCC.
VEE	volt		negative power supply voltage. The amplifier VONEG was measured with respect to VCC.
VONEG (*VON)*	volt		maximum negative output voltage. This is less than VEE (the negative power-supply voltage) by the internal voltage drop.

Names(Alias)	Units	Default	Description
VOPOS (VOP)	volt		maximum positive output voltage. This is less than VCC, the positive power supply voltage, by the internal voltage drop.
VOS	volt		input offset voltage. This is the voltage required between the input differential transistors to zero the output voltage. This is generally a fundamental electrical characteristic. Typical values for bipolar amplifiers are in the range 0.1 mV to 10 mV. VOS is measured in volts. VOS can cause a failure to converge for some amplifiers. If this occurs, try setting VOS to 0 or use the initial conditions described above for convergence.

Op-Amp Model Parameter Defaults

Parameter	Description	Defaults		
		DEF=0	DEF=1	DEF=2
AV	amplifier voltage gain	160k	417k	200k
AV1K	amplifier voltage gain at 1 kHz	-	1.2 k	3 k
C2	feedback capacitance	30 p	30 p	10 p

Parameter	Description	Defaults		
		DEF=0	DEF=1	DEF=2
CMRR	common mode rejection ratio	96 db	106 db	90 db
		63.1k	199.5k	31.63k
COMP	compensation level selector	0	0	0
DEF	default level selector	0	1	2
DELPHS	delta phase at unity gain	25 °	17 °	52 °
DIS	diode saturation current	8e-16	8e-16	8e-16
FREQ	unity gain frequency	600 k	-	-
IB	input bias current	30 n	250 n	40 n
IBOS	input bias offset current	1.5 n	0.7 n	5 n
ISC	output short circuit current	25 mA	25 mA	25 mA
LEVIN	input circuit level selector	1	1	1
LEVOUT	output circuit level selector	1	1	1
MANU	manufacturer’s name	-	-	-
PWR	power dissipation	72 mW	60 mW	50 mW
RAC	AC output resistance	0	75	70
ROUT	DC output resistance	200	550	100
SRPOS	positive output slew rate	450 k	1 meg	1 meg
SRNEG	negative output slew rate	450 k	800 k	800 k
TEMP	temperature of model	25 deg	25 deg	25 deg
VCC	positive supply voltage for VOPOS	20	15	15
VEE	negative supply voltage for VONEG	-20	-15	-15
VONEG	maximum negative output	-14	-14	-14

Parameter	Description	Defaults		
		DEF=0	DEF=1	DEF=2
VOPOS	maximum positive output	14	14	14
VOS	input offset voltage	0	0.3 m	0.5 m

Op-Amp Subcircuit Example

AUTOSTOP Option

This example uses the .OPTION AUTOSTOP option to shorten simulation time. Once Star-Hspice makes the measurements specified by the .MEASURE statement, the associated transient analysis and AC analysis stops whether or not the full sweep range for each has been covered.

AC Resistance

AC=10000G parameter in the Rfeed element statement installs a 10000 GΩ feedback resistor for the AC analysis in place of the 10 kΩ feedback resistor – used in the DC operating point and transient analysis – which is open-circuited for the AC measurements.

Simulation Results

The simulation results give the DC operating point analysis for an input voltage of 0 v and power supply voltages of 15v. The DC offset voltage is 3.3021 mv, which is less than that specified for the original vos specification in the op-amp .MODEL statement. The unity gain frequency is given as 907.885 kHz, which is within 10% of the 1 MHz specified in the .MODEL statement with the parameter FREQ. The required time rate for a 1 volt change in the output (from the .MEASURE statement) is 2.3 μs (from the SRPOS simulation result listing) providing a slew rate of 0.434 Mv/s. This compares to within about 12% of the 0.5 Mv/s given by the SRPOS parameter in the .MODEL statement. The negative slew rate is almost exactly 0.5 Mv/s, which is within 1% of the slew rate specified in the .MODEL statement.

Example Amplifier Model

```
$$ FILE  ALM124.SP
.OPTION NOMOD AUTOSTOP  SEARCH=' '
.OP VOL
.AC DEC 10 1HZ  10MEGHZ
.MODEL PLOTDB   PLOT XSCAL=2 YSCAL=3
.MODEL PLOTLOGX PLOT XSCAL=2
.GRAPH AC MODEL=PLOTDB VM(OUT0)
.GRAPH AC MODEL=PLOTLOGX VP(OUT0)
.TRAN 1U 40US 5US .15MS
.GRAPH V(IN) V(OUT0)
.MEASURE TRAN 'SRPOS'     TRIG V(OUT0) VAL=2V RISE=1
+                         TARG V(OUT0) VAL=3V RISE=1
.MEASURE TRAN 'SRNEG'     TRIG V(OUT0) VAL=-2V FALL=1
+                         TARG V(OUT0) VAL=-3V FALL=1
.MEASURE AC   'UNITFREQ'  TRIG AT=1
+                         TARG VDB(OUT0) VAL=0 FALL=1
.MEASURE AC   'PHASEMARGIN'  FIND VP(OUT0)
+                         WHEN VDB(OUT0)=0
.MEASURE AC   'GAIN(DB)'      MAX VDB(OUT0)
.MEASURE AC   'GAIN(MAG)'     MAX VM(OUT0)
VCC   VCC GND  +15V
VEE   VEE GND  -15V
VIN IN  GND  AC=1 PWL 0US   0V 1US   0V 1.1US +10V 15US +10V
+  15.2US -10V 100US -10V
.MODEL ALM124 AMP
+       C2=  30.00P   SRPOS=     .5MEG       SRNEG=  .5MEG
+       IB=  45N       IBOS=   3N              VOS= 4M
+     FREQ=  1MEG    DELPHS=   25             CMRR= 85
+     ROUT=  50          AV=   100K            ISC=  40M
+    VOPOS=  14.5     VONEG=  -14.5            PWR= 142M
+      VCC=  16         VEE=  -16             TEMP= 25.00
+     PSRR=  100        DIS=    8.00E-16       JIS=  8.00E-16
*
```

Unity Gain Resistor Divider Mode

```
*
Rfeed  OUT0  IN- 10K AC=10000G
RIN     IN  IN- 10K
RIN+    IN+ GND 10K
X0  IN-  IN+ OUT0  VCC VEE ALM124
```

```
ROUT0  OUT0 GND    2K
COUT0  OUT0 GND  100P

.END

***** OPERATING POINT STATUS IS VOLTAGE   SIMULATION TIME
IS     0.
  NODE     =VOLTAGE      NODE    =VOLTAGE      NODE  =VOLTAGE
+0:IN       =   0.      0:IN+  =-433.4007U 0:IN-  =  3.3021M
+0:OUT0     =   7.0678M 0:VCC  =  15.0000  0:VEE  = -15.0000

unitfreq    = 907.855K  TARG   = 907.856K  TRIG   =   1.000
PHASEMARGIN =  66.403
gain(db)    =  99.663   AT     =   1.000
FROM        =   1.000   TO     =  10.000X
gain(mag)   =  96.192K  AT     =   1.000
FROM        =   1.000   TO     =  10.000X
srpos       =   2.030U  TARG   =  35.471U  TRIG   =  33.442U
srneg       =   1.990U  TARG   =   7.064U  TRIG   =   5.074U
```

741 Op-Amp from Controlled Sources

The µA741 op-amp is modeled by PWL controlled sources. The output is limited to ±15 volts by a piecewise linear CCVS (source “h”).

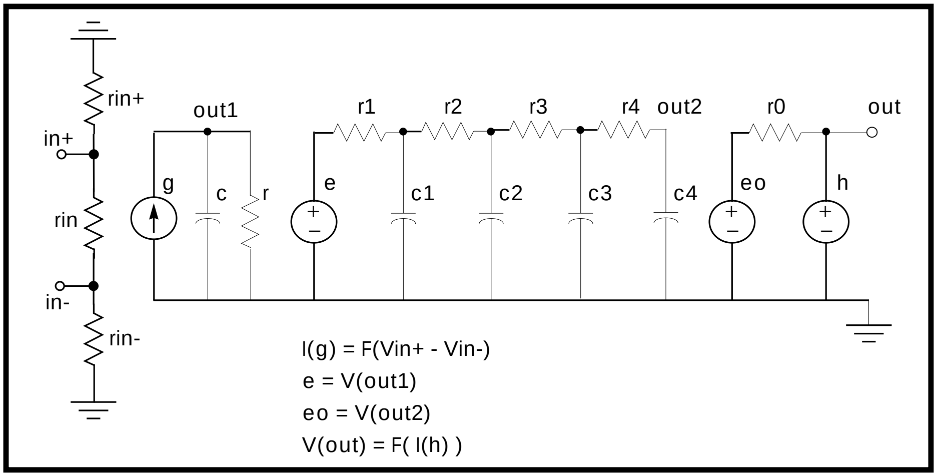


Figure 22-31: Op-Amp Circuit

Example Op-Amp

```
0p_amp.sp --- operational amplifier
*
.options post=2
.tran .001ms 2ms
.ac dec 10 .1hz 10me’
*.graph tran vout=v(output)
*.graph tran vin=v(input)
*.graph ac model=grap voutdb=vdb(output)
*.graph ac model=grap vphase=vp(output)
.probe tran vout=v(output) vin=v(input)
.probe ac voutdb=vdb(output) vphase=vp(output)
.model grap plot xscal=2
```

Main Circuit

```
xamp input 0 output opamp
vin input 0 sin(0,1m,1k) ac 1
* subcircuit definitions
* input subckt
.subckt opin in+ in- out
rin  in+ in- 2meg
rin+ in+ 0 500meg
rin- in- 0 500meg
g 0 out pwl(1) in+ in- -68mv,-68ma 68mv,68ma delta=1mv
c out 0 .136uf
r out 0 835k
.ends
```

RC Circuit With Pole At 9 MHz

```
.subckt oprc in out
e out1 0 in 0 1
r1 out1 out2 168
r2 out2 out3 1.68k
r3 out3 out4 16.8k
r4 out4 out 168k
c1 out2 0 100p
c2 out3 0 10p
c3 out4 0 1p
c4 out 0 .1p
r out 0 1e12
.ends
```

Output Limiter to 15 v

```
.subckt opout in out
eo out1 0 in 0 1
ro out1 out 75
vdum out dum 0
h dum 0 pwl(1) vdum delta=.01ma -.1ma,-15v .1ma,15v
.ends
* op-amp subckt
.subckt opamp in+ in- out
xin in+ in- out1 opin
xrc out1 out2 oprc
xout out2 out opout
.ends
.end
```

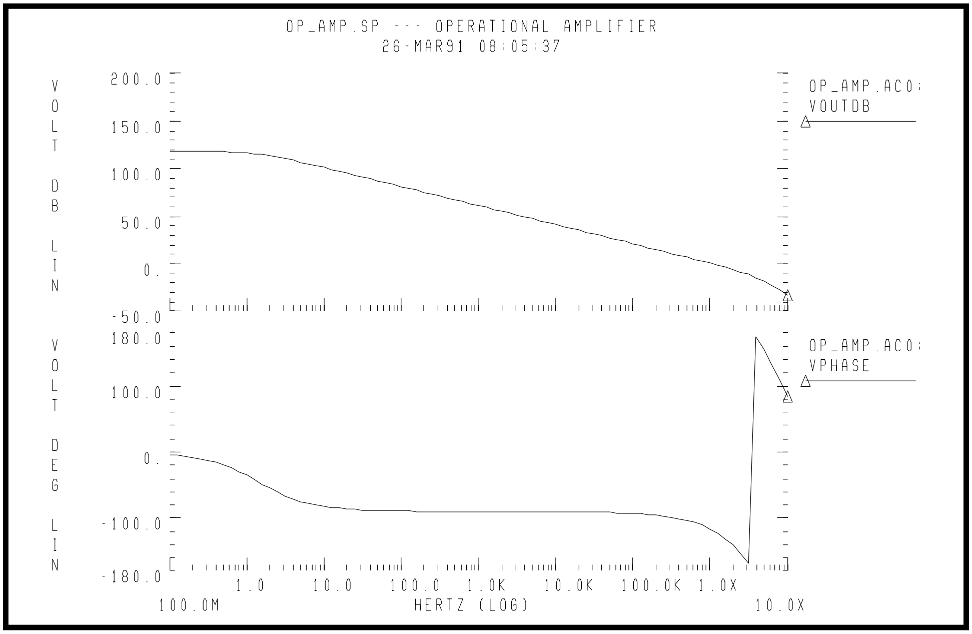


Figure 22-32: AC Analysis Response

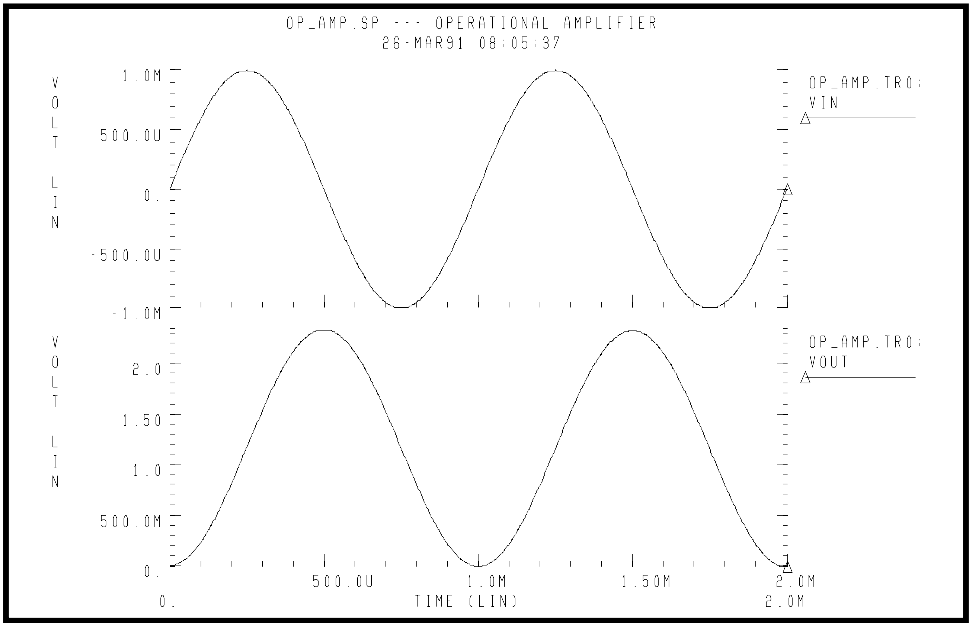


Figure 22-33: Transient Analysis Response[1]

Inverting Comparator with Hysteresis

An inverting comparator is modelled by a piecewise linear VCVS.

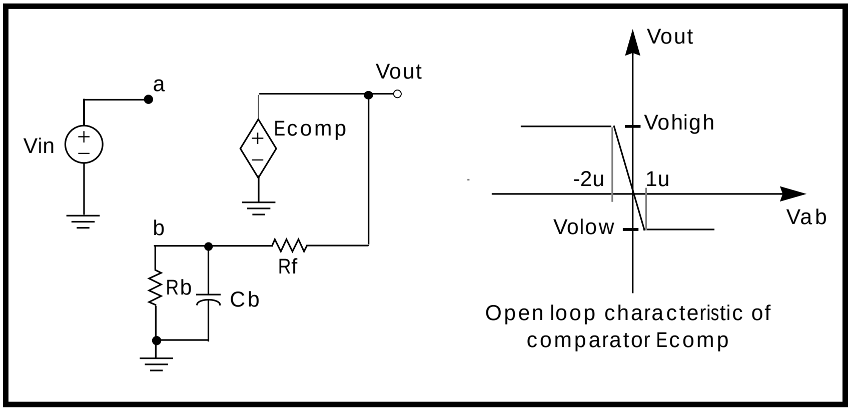


Figure 22-34: Inverting Comparator with Hysteresis

Two reference voltages corresponding to volow and vohigh of Ecomp characteristic are:

$$Vreflow = \frac{Volow \cdot Rb}{Rb + Rf}$$

$$Vrefhigh = \frac{Vohigh \cdot Rb}{Rb + Rf}$$

When Vin exceeds Vrefhigh, the output Vout goes to Volow. For Vin less than Vreflow, the output goes to Vohigh.

Example Inverting Comparator with Hysteresis

```
Compar.sp Inverting comparator with hysterisis
.OPTIONS POST PROBE
.PARAM  vohigh=5v volow=-2.5v rbval=1k rfval=9k
Ecomp  out 0 PWL(1) a  b   -2u,vohigh  1u,volow
Rb  b  0   rbval
Rf  b  out rfval
Cb  b  0 1ff
Vin a  0 PWL(0,-4 1u,4 2u,-4)
.TRAN .1n  2u
.PROBE Vin=V(a) Vab=V(a,b) Vout=V(out)
.END
```

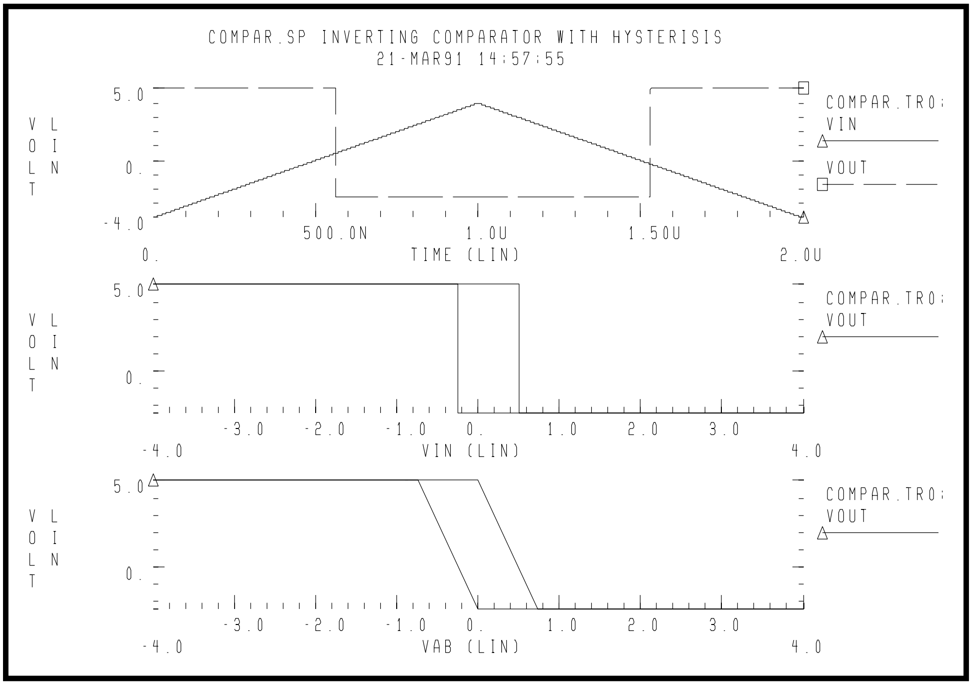


Figure 22-35: Response of Comparator

Voltage Controlled Oscillator

In this example, a one-input NAND functioning as an inverter models a five stage ring oscillator. PWL capacitance is used to switch the load capacitance of this inverter from 1pF to 3 pF. As the simulation results indicate, the oscillation frequency decreases as the load capacitance increases.

Example Voltage Controlled Oscillator

```
vcob.sp voltage controlled oscillator using pwl functions
.OPTION POST
.GLOBAL ctrl
.TRAN 1n 100n
.IC V(in)=0 V(out1)=5
.PROBE TRAN V(in) V(out1) V(out2) V(out3) V(out4)
X1 in out1 inv
X2 out1 out2 inv
X3 out2 out3 inv
X4 out3 out4 inv
X5 out4 in inv
Vctrl ctrl 0 PWL(0,0 35n,0 40n,5)
```

Subcircuit Definition

```
.SUBCKT inv in out rout=1k
* The following G element is functioning as PWL capacitance.
Gcout out 0 VCCAP PWL(1) ctrl 0  DELTA=.01
+ 4.5 1p
+ 4.6 3p
Rout out 0 rout
Gn 0 out NAND(1) in 0 SCALE='1.0k/rout'
+ 0. 5.00ma
+ 0.25 4.95ma
+ 0.5 4.85ma
+ 1.0 4.75ma
+ 1.5 4.42ma
+ 3.5 1.00ma
+ 4.000 0.50ma
+ 4.5 0.20ma
+ 5.0 0.05ma
.ENDS inv
*
.END
```

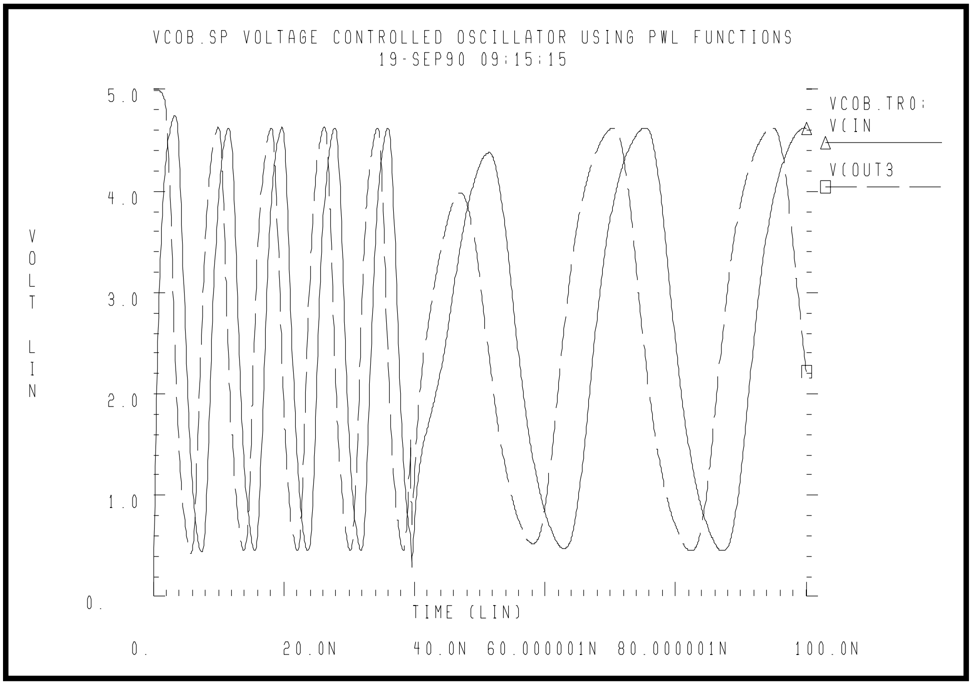


Figure 22-36: Voltage Controlled Oscillator Response

LC Oscillator

The capacitor is initially charged to 5 volts. The value of capacitance is the function of voltage at node 10. The value of capacitance becomes four times higher at time t2. The frequency of this LC circuit is given by:

$$freq = \frac{1}{6.28 \cdot \sqrt{L \cdot C}}$$

At time t2, the frequency must be halved. The amplitude of oscillation depends on the condition of the circuit when the capacitance value changes. The stored energy is:

$$E = (0.5 \cdot C \cdot V^2) + (0.5 \cdot L \cdot I^2)$$

$$E = 0.5 \cdot C \cdot Vm^2, I = 0$$

$$E = 0.5 \cdot L \cdot Im^2, V = 0$$

Assuming at time t2, when V=0, C changes to A · C, then:

$$0.5 \cdot L \cdot Im^2 = 0.5 \cdot Vm^2 = 0.5 \cdot (A \cdot C) \cdot Vm'^2$$

and from the above equation:

$$Vm' = \frac{Vm}{\sqrt{A}}$$

$$Qm' = \sqrt{A} \cdot Vm$$

The second condition to consider is when V=Vin, C changes to A· C. In this case:

$$Qm = Qm'$$

$$C \cdot Vm = A \cdot C \cdot Vm'$$

$$Vm' = \frac{Vm}{A}$$

Therefore the voltage amplitude is modified between Vm/sqrt(A) and Vm/A depending on the circuit condition at the switching time. This example tests the CTYPE 0 and 1 results. The result for CTYPE=1 must be correct because capacitance is a function of voltage at node 10, not a function of the voltage across the capacitor itself.

Example Voltage Variable Capacitance

```
calg2.sp voltage variable capacitance
*
.OPTION POST
.IC v(1)=5 v(2)=5
C1 1 0 C='1e-9*V(10)' CTYPE=1
L1 1 0 1m
*
C2 2 0 C='1e-9*V(10)' CTYPE=0
L2 2 0 1m
*
V10 10 0 PWL(0sec,1v t1,1v t2,4v)
R10 10 0 1
*
.TRAN .1u 60u UIC SWEEP DATA=par
.MEAS TRAN period1 TRIG V(1) VAL=0 RISE=1
+ TARG V(1) VAL=0 RISE=2
.MEAS TRAN period2 TRIG V(1) VAL=0 RISE=5
+ TARG V(1) VAL=0 RISE=6
.PROBE TRAN V(1) q1=LX0(C1)
*
.PROBE TRAN V(2) q2=LX0(C2)
.DATA par t1 t2
15.65us 15.80us
17.30us 17.45us
.ENDDATA
.END
```

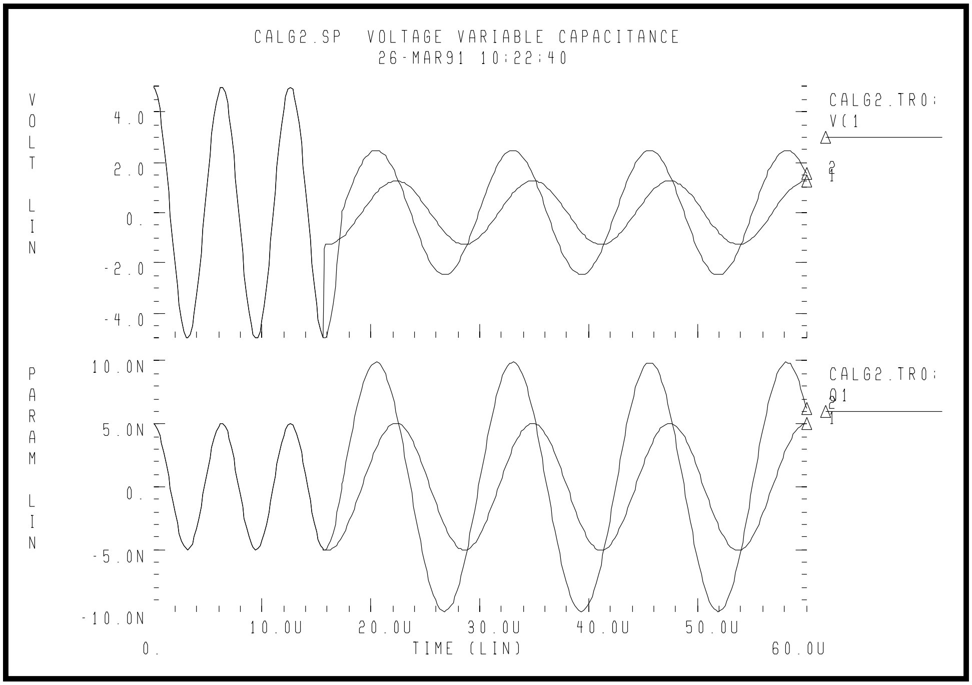


Figure 22-37: Correct Result Corresponding to CTYPE=1

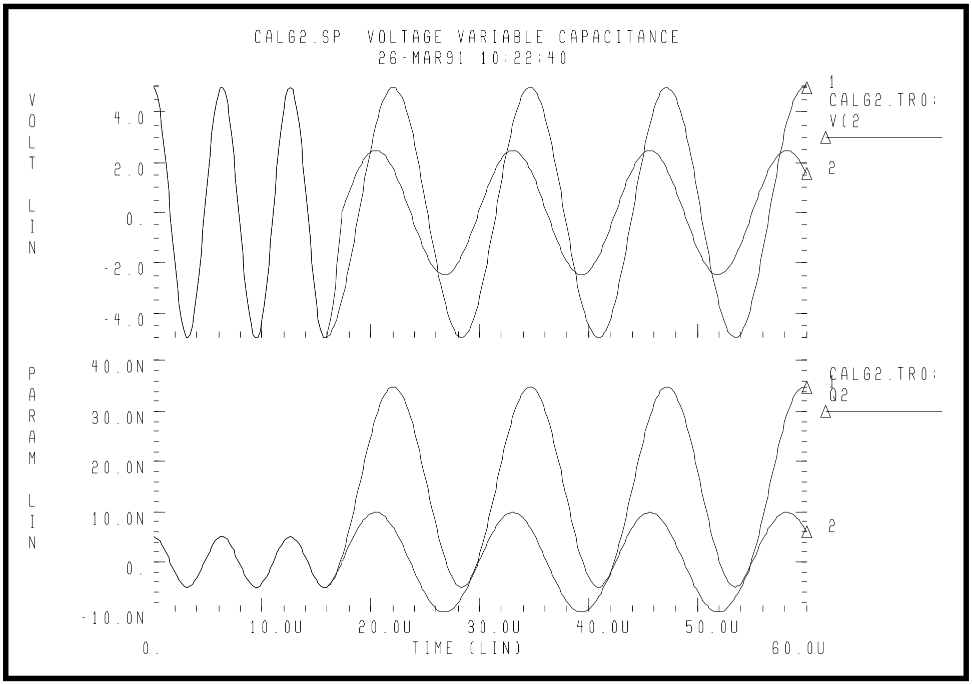


Figure 22-38: Incorrect Result Corresponding to CTYPE=0

Using a Phase Locked Loop Design

Phase Detector Using Multi-Input NAND Gates

This circuit uses the behavioral elements to implement inverters, 2, 3, and 4 input NAND gates.

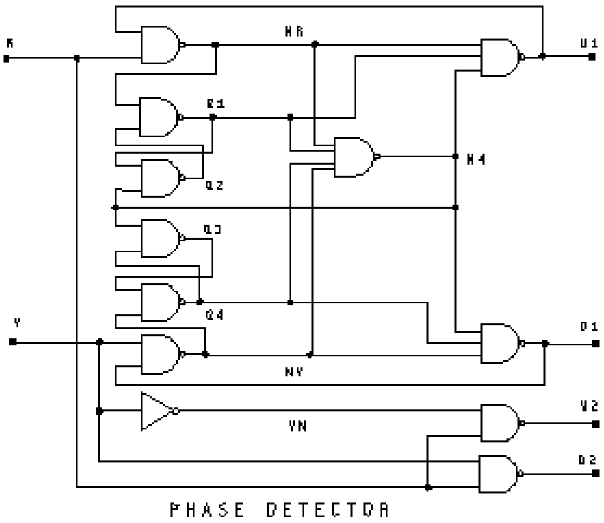


Figure 22-39: Circuit Schematic of Phase Detector

Example Phase Detector with Behavioral NAND Gates

```
pdb.sp phase detector using behavioral nand gates.
.option post=2
.tran .25n 50ns
*.graph tran v(r) v(v) v(u1)
*.graph tran v(r) v(v) v(u2) $ v(d2)
.probe tran v(r) v(v) v(u1)
.probe tran v(r) v(v) v(u2) $ v(d2)
xnr r u1 nr nand2 capout=.1p
```

```
xq1 nr q2 q1 nand2 capout=.1p
xq2 q1 n4 q2 nand2
xq3 q4 n4 q3 nand2
xq4 q3 nv q4 nand2
xnv v d1 nv nand2
xu1 nr q1 n4 u1 nand3
xd1 nv q4 n4 d1 nand3
xvn v vn inv
xu2 vn r u2 nand2
xd2 r v d2 nand2
xn4 nr q1 q4 nv n4 nand4
*
* waveform vv lags waveform vr
vr r 0 pulse(0,5,0n,1n,1n,15n,30n)
vv v 0 pulse(0,5,5n,1n,1n,15n,30n)
*
* waveform vr lags waveform vv
*vr r 0 pulse(0,5,5n,1n,1n,15n,30n)
*vv v 0 pulse(0,5,0n,1n,1n,15n,30n)
```

Subcircuit Definitions

```
.SUBCKT inv in out capout=.1p
cout out 0 capout
rout out 0 1.0k
gn 0 out nand(1) in 0 scale=1
+ 0. 4.90ma
+ 0.25 4.88ma
+ 0.5 4.85ma
+ 1.0 4.75ma
+ 1.5 4.42ma
+ 3.5 1.00ma
+ 4.000 0.50ma
+ 4.5 0.2ma
+ 5.0 0.1ma
.ENDS inv
*
.SUBCKT nand2 in1 in2 out capout=.15p
cout out 0 capout
rout out 0 1.0k
gn 0 out nand(2) in1 0 in2 0 scale=1
+ 0. 4.90ma
+ 0.25 4.88ma
+ 0.5 4.85ma
+ 1.0 4.75ma
+ 1.5 4.42ma
+ 3.5 1.00ma
+ 4.000 0.50ma
+ 4.5 0.2ma
+ 5.0 0.1ma
.ENDS nand2
*
.SUBCKT nand3 in1 in2 in3 out capout=.2p
cout out 0 capout
rout out 0 1.0k
gn 0 out nand(3) in1 0 in2 0 in3 0 scale=1
+ 0. 4.90ma
+ 0.25 4.88ma
+ 0.5 4.85ma
+ 1.0 4.75ma
+ 1.5 4.42ma
+ 3.5 1.00ma
+ 4.000 0.50ma
+ 4.5 0.2ma
+ 5.0 0.1ma
.ENDS nand3
*
.SUBCKT nand4 in1 in2 in3 in4 out capout=.5p
cout out 0 capout
rout out 0 1.0k
gn 0 out nand(4) in1 0 in2 0 in3 0 in4 0 scale=1
+ 0. 4.90ma
+ 0.25 4.88ma
+ 0.5 4.85ma
+ 1.0 4.75ma
+ 1.5 4.42ma
+ 3.5 1.00ma
+ 4.000 0.50ma
+ 4.5 0.2ma
+ 5.0 0.1ma
.ENDS nand4
.end
```

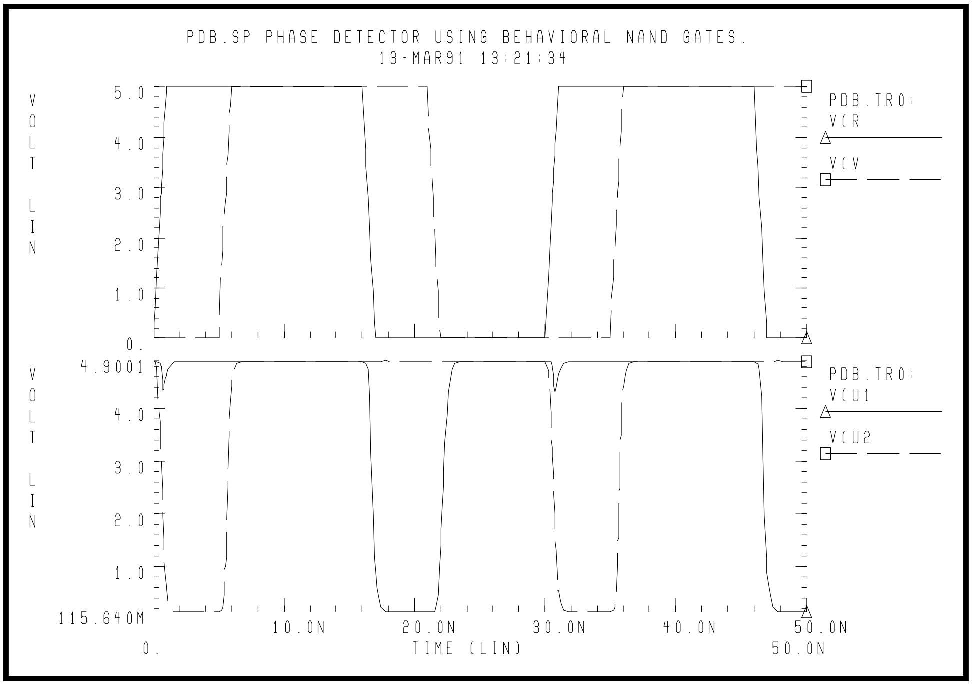


Figure 22-40: Phase Detector Response

PLL BJT Behavioral Modeling

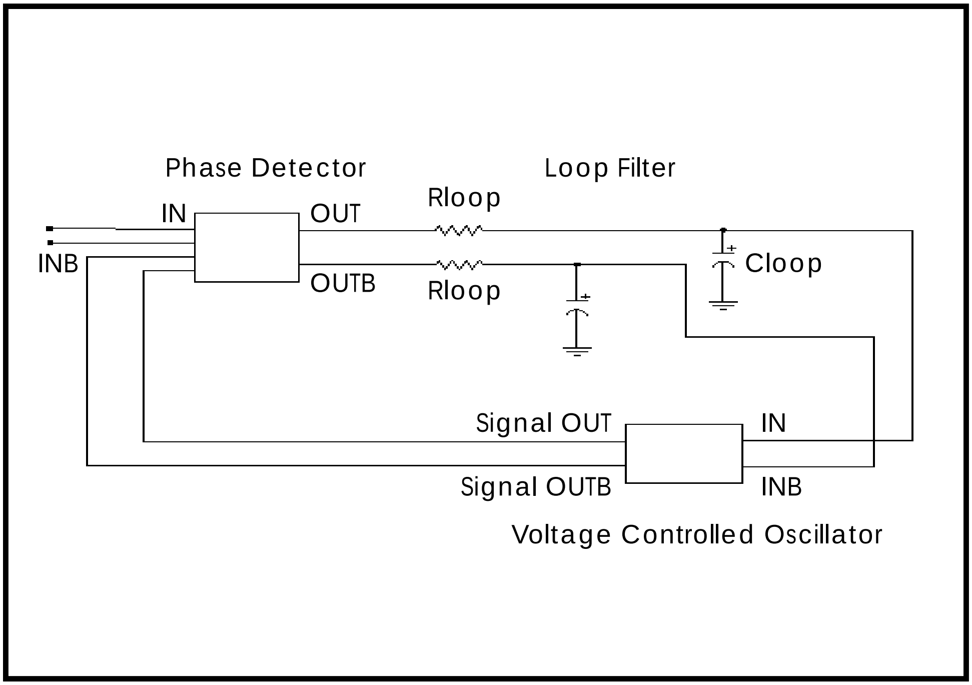


Figure 22-41: PLL Schematic

Example Phase Locked Loop

A Phase Locked Loop (PLL) circuit synchronizes to an input waveform within a selected frequency range, returning an output voltage proportional to variations in the input frequency. It has three basic components: a voltage controlled oscillator (VCO), which returns an output waveform proportional to its input voltage, a phase detector which compares the VCO output to the input waveform and returns an output voltage depending on their phase difference, and a loop filter, which filters the phase detector voltage, returning an output voltage which forms the VCO input and the external voltage output of the PLL.

The following example shows an Star-Hspice simulation of a full bipolar implementation of a PLL; its transfer function shows a linear region of voltage vs. (periodic) time which is defined as the “lock” range. The phase detector is modeled behaviorally, effectively implementing a logical XNOR function. This model was then substituted into the full PLL circuit and resimulated. The behavioral model for the VCO was then substituted into the PLL circuit, and this behavioral PLL was then simulated. The results of the transient simulations (Figure 22-42:) show minimal difference between implementations, but from the standpoint of run time statistics, the behavioral model shows a factor of five reduction in simulation time versus that of the full circuit.

Include the behavioral model if you use this PLL in a larger system simulation (for example, an AM tracking system) because it substantially reduces run time while still representing the subcircuit accurately.

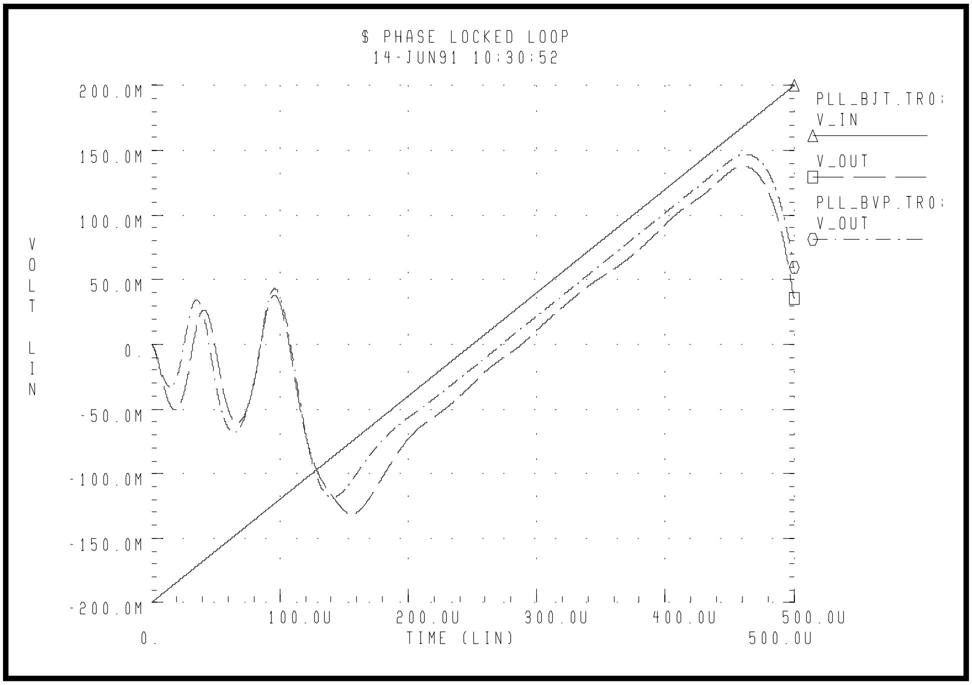


Figure 22-42: Behavioral (PLL_BVP Curve) vs. Bipolar (PLL_BJT Curve) Circuit Simulation

Example Phase Locked Loop

```
$ phase locked loop
.option post probe acct
.option relv=1e-5
$
$ wideband FM example, Grebene gives:
$ f0=1meg kf=250kHz/V
$ kd=0.1 V/rad
$ R=10K C=1000p
$ f_lock = kf*kd*pi/2 = 39kHz, v_lock = kd*pi/2 = 0.157
$ f_capture/f_lock ~= 1/sqrt(2*pi*R*C*f_lock)
$ = 0.63, v_capture ~= 0.100

*.ic v(out)=0 v(fin)=0
.tran .2u 500u
.option delmax=0.01u interp
.probe v_in=v(inc,0) v_out=v(out,outb)
.probe v(in) v(osc) v(mout) v(out)
```

Input

```
vin inc 0 pwl 0u,-0.2 500u,0.2
*vin inc 0 0
xin inc 0 in inb vco f0=1meg kf=125k phi=0 out_off=0
out_amp=0.3
$ vco
xvco e eb osc oscb vco f0=1meg kf=125k phi=0 out_off=-1
out_amp=0.3

$ phase detector
xpd in inb osc oscb mout moutb pd kd=0.1 out_off=-2.5

$ filter
rf mout e 10k
cf e 0 1000p
rfb moutb eb 10k
cfb eb 0 1000p

$ final output
rout out e 100k
cout out 0 100p
routb outb eb 100k
```

```
coutb outb 0 100p

.macro vco in inb out outb f0=100k kf=50k phi=0.0 out_off=0.0
out_amp=1.0
gs 0 s poly(2) c 0 in inb 0 ‘6.2832e-9*f0’ 0 0 ‘6.2832e-9*kf’
gc c 0 poly(2) s 0 in inb 0 ‘6.2832e-9*f0’ 0 0 ‘6.2832e-9*kf’
cs s 0 1e-9
cc c 0 1e-12
e1 s_clip 0 pwl(1) s 0 -0.1,-0.1 0.1,0.1
eout 0 s_clip 0 out_off vol=‘10*out_amp’
eboutb 0 s_clip 0 out_off vol=‘-10*out_amp’
.ic v(s)=’sin(phi)’ v(c)=’cos(phi)’
.eom

.macro pd in inb in2 in2b out outb kd=0.1 out_off=0
e1 clip1 0 pwl(1) in inb -0.1,-0.1 0.1,0.1
e2 clip2 0 pwl(1) in2 in2b -0.1,-0.1 0.1,0.1
e3 n1 0 poly(2) clip1 0 clip2 0 0 0 0 0 ‘78.6*kd’
e4 outb 0 n1 0 out_off 1
e5 out 0 n1 0 out_off -1
.eom

.end
```

Example BJT Level Voltage Controlled Oscillator (VCO)

```
$ phase locked loop
.option post probe acct
.option relv=1e-5
$
$ wideband FM example, Grebene gives:
$ f0=1meg kf=250kHz/V
$ kd=0.1 V/rad
$ R=10K C=1000p
$ f_lock = kf*kd*pi/2 = 39kHz, v_lock = kd*pi/2 = 0.157
$ f_capture/f_lock ~= 1/sqrt(2*pi*R*C*f_lock)
$ = 0.63, v_capture ~= 0.100

*.ic v(out)=0 v(fin)=0
.tran .2u 500u
.option delmax=0.01u interp
.probe v_in=v(inc,0) v_out=v(out,outb)
```

```
.probe v(in) v(osc) v(mout) v(out) v(e)

vcc vcc 0 6
vee vee 0 -6

$ input
vin inc 0 pwl 0u,-0.2 500u,0.2
xin inc 0 in inb vco f0=1meg kf=125k phi=0 out_off=0
out_amp=0.3

$ vco
xvco1 e eb osc oscb 0 vee vco1
.ic v(osc)=-1.4 v(oscb)=-0.7
```

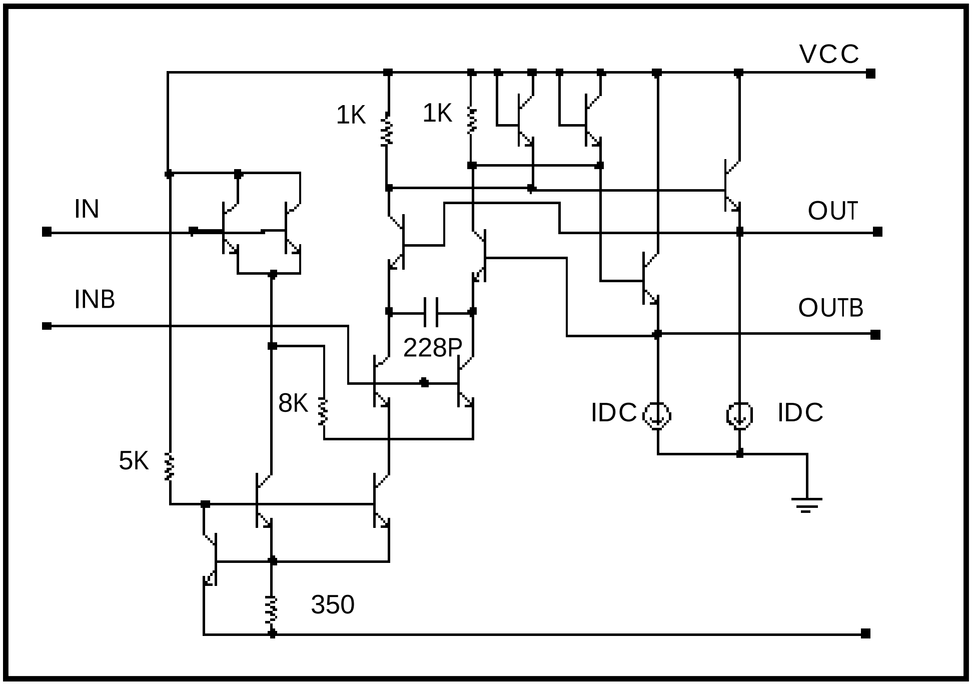


Figure 22-43: Voltage Controlled Oscillator Circuit

BJT Level Phase Detector

Example Phase Detector

```
$ phase detector
xpd1 in inb osc oscb mout moutb vcc vee pd1
```

Filter

```
rf mout e 10k
cf e 0 1000p
rfb moutb eb 10k
cfb eb 0 1000p
```

Final Output

```
rout out e 100k
cout out 0 100p
routb outb eb 100k
coutb outb 0 100p

.macro vco in inb out outb f0=100k kf=50k phi=0.0 out_off=0.0
out_amp=1.0
gs 0 s poly(2) c 0 in inb 0 ‘6.2832e-9*f0’ 0 0 ‘6.2832e-9*kf’
gc c 0 poly(2) s 0 in inb 0 ‘6.2832e-9*f0’ 0 0 ‘6.2832e-9*kf’
cs s 0 1e-9
cc c 0 1e-9
e1 s_clip 0 pwl(1) s 0 -0.1,-0.1 0.1,0.1
e out 0 s_clip 0 out_off ‘10*out_amp’
eb outb 0 s_clip 0 out_off ‘-10*out_amp’
.ic v(s)=’sin(phi)’ v(c)=’cos(phi)’
.eom

.macro pd in inb in2 in2b out outb kd=0.1 out_off=0
e1 clip1 0 pwl(1) in inb -0.1,-0.1 0.1,0.1
e2 clip2 0 pwl(1) in2 in2b -0.1,-0.1 0.1,0.1
e3 n1 0 poly(2) clip1 0 clip2 0 0 0 0 0 ‘78.6*kd’
e4 outb 0 n1 0 out_off 1
e5 out 0 n1 0 out_off -1
.eom

.macro vco1 in inb e7 e8 vcc vee vco_cap=228.5p
qout vcc vcc b7 npn1
```

```
qoutb vcc vcc b8 npn1
rb vcc c0 5k $ 1ma
q0 c0 b0 vee npn1
q7 vcc b7 e7 npn1
r4 vcc b7 1k
i7 e7 0 1m
q8 vcc b8 e8 npn1
r5 vcc b8 1k
i8 e8 0 1m
q9 b7 e8 e9 npn1
q10 b8 e7 e10 npn1
c0 e9 e10 vco_cap
q11 e9 in 2 npn1 $ ic=i0
q12 e10 in 2 npn1 $ ic=i0
q15 2 c0 b0 npn1 $ ic=2*i0
q16 3 c0 b0 npn1 $ ic=2*i0
rx 2 3 8k
q13 vcc inb 3 npn1
q14 vcc inb 3 npn1
rt b0 vee 350 $ i=4*i0=2m
.eom

.model npn1 npn
+ eg=1.1 af=1 xcjc=0.95 subs=1
+ cjs=0 tf=5p
+ tr=500p cje=0.2p cjc=0.2p fc=0.8
+ vje=0.8 vjc=0.8 mje=0.33 mjc=0.33
+ rb=0 rbm=0 irb=10u
+ is=5e-15 ise=1.5e-14 isc=0
+ vaf=150 bf=100 ikf=20m
+ var=30 br=5 ikr=15m
+ rc=0 re=0
+ nf=1 ne=1.5 nc=1.2
+ tbf1=8e-03

.macro pd1 in inb in2 in2b out outb vcc vee
rl vcc n1 1k
rlb vcc n1b 1k
q3 n1 in c1 npn1
q4 n1b inb c1 npn1
q5 n1 inb c2 npn1
q6 n1b in c2 npn1
```

```
q1 c1 in2 e npn1
q2 c2 in2b e npn1
ie e 0 0.5m
c1 n1 0 1p
c1b n1b 0 1p
q7 vcc n1 e7 npn1
q8 vcc n1b e8 npn1
r1 e7 out 625
r2 out vee 300
r1b e8 outb 625
r2b outb vee 300
.eom
.end
```

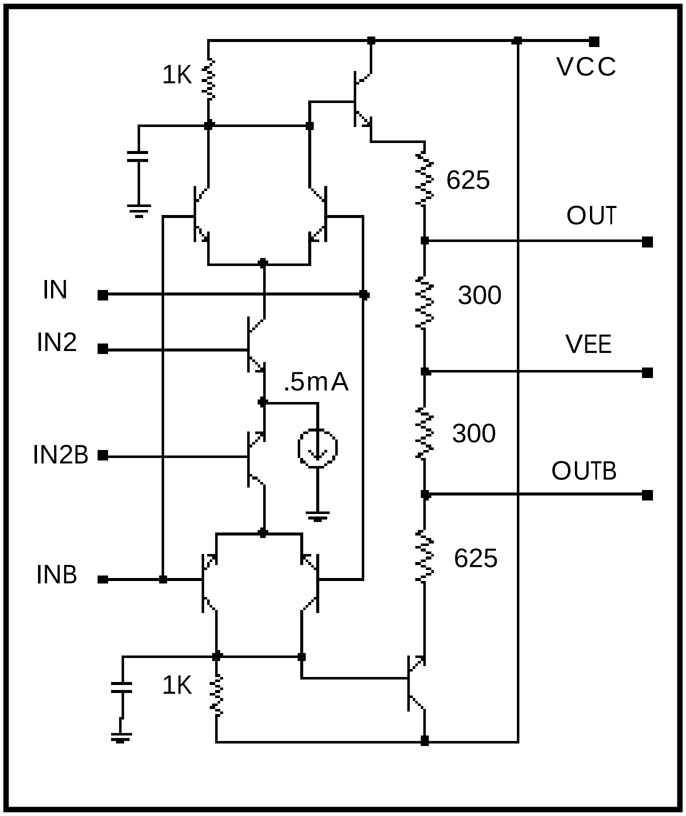


Figure 22-44: Phase Detector Circuit

References

1. Chua & Lin. *Computer Aided Analysis of Electronic Circuit*s. Englewood Cliffs: Prentice-Hall, 1975, page 117. See also “SPICE2 Application Notes for Dependent Sources,” by Bert Epler, *IEEE Circuits & Devices Magazine*, September 1987.