Avant!

Chapter 19

Performing Cell Characterization

Most ASIC vendors use Star-Hspice to characterize their standard cell libraries and prepare data sheets by using the basic capabilities of the .MEASURE statement. Input sweep parameters and the resulting measure output parameters are stored in the measure output data files *design.mt0*, *design.sw0*, and *design.ac0*. Multiple sweep data is stored in this file, and you can plot it by using AvanWaves. This lends itself to generating fanout plots of delay versus load. The slope and intercept of the loading curves can be used to calibrate VHDL, Verilog, Lsim, TimeMill, and Synopsys models.

This chapter covers:

- Determining Typical Data Sheet Parameters
  A series of typical data sheet examples show the flexibility of the MEASURE statement.
- Performing Data Driven Analysis
  Automates cell characterization, including timing simulator polynomial delay coefficient calculation. There is no limit on the number of parameters simultaneously varied or the number of analyses to be performed. Convenient ASCII file format for automated parameter input to Star-Hspice.
- Using Digital File Input Stimuli
  You can use logic state transition tables to produce the input stimuli for the characterization. The D2A model in Star-Hspice provides a 28-state logic simulator interface for rapid figuring of a cell characterization testbed.

Determining Typical Data Sheet Parameters

This section describes how to determine typical data sheet parameters.

Rise, Fall, and Delay Calculations

The following example first calculates *vmax*, using the MAX function over the time region of interest. Then it calculates *vmin* using the MIN function. Finally, the measured parameters can be used in subsequent calculations for accurate 10% and 90% points in the determination of the rise and fall time. Note that the RISE=1 is relative to the time window formed by the delay *TDval.* Finally, the delay *Tdelay* is calculated using a fixed value for the measure threshold.

Example

```
.MEAS TRAN vmax MAX V(out) FROM=TDval TO=Tstop
.MEAS TRAN vmin MIN V(out) FROM=TDval TO=Tstop
.MEAS TRAN Trise TRIG V(out) val='vmin+0.1*vmax'  TD=TDval
+  RISE=1 TARG  V(out) val='0.9*vmax' RISE=1
.MEAS TRAN  Tfall TRIG V(out) val='0.9*vmax' TD=TDval
+  FALL=2 TARG  V(out) val='vmin+0.1*vmax' FALL=2
.MEAS TRAN Tdelay TRIG V(in) val=2.5 TD=TDval FALL=1
+  TARG V(out) val=2.5 FALL=2
```

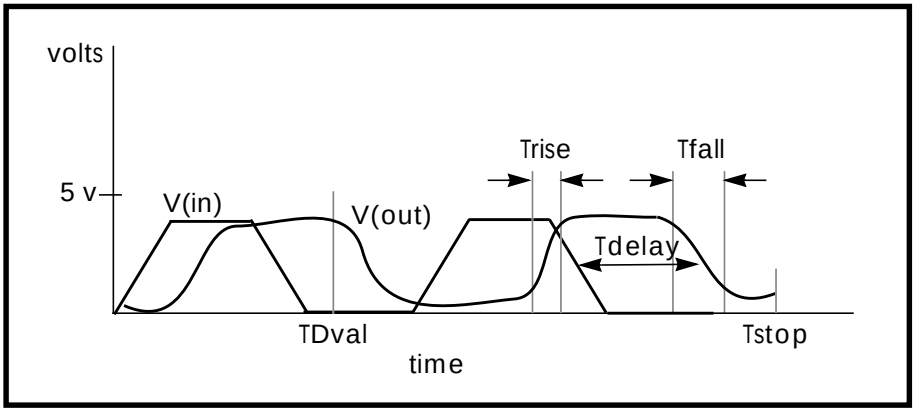

Figure 19-1: Rise, Fall, and Delay Time Demonstration

Ripple Calculation

This example performs the following:

- Delimits the wave at the 50% of VCC points
- Finds the midpoint *Tmid*
- Defines a bounded region by finding the pedestal voltage (*Vmid*) and then finding the first time that the signal crossed this value, *Tfrom*
- Measures the ripple in the defined region using the peak-to-peak (PP) measure function from *Tfrom* to *Tmid*

Example

```
.MEAS TRAN Th1 WHEN V(out)='0.5*vcc' CROSS=1
.MEAS TRAN Th2 WHEN V(out)='0.5*vcc' CROSS=2
.MEAS TRAN Tmid PARAM='(Th1+Th2)/2'
.MEAS TRAN Vmid FIND V(out) AT='Tmid'
.MEAS TRAN Tfrom WHEN V(out)='Vmid' RISE=1
.MEAS TRAN Ripple PP V(out) FROM='Tfrom' TO='Tmid'
```

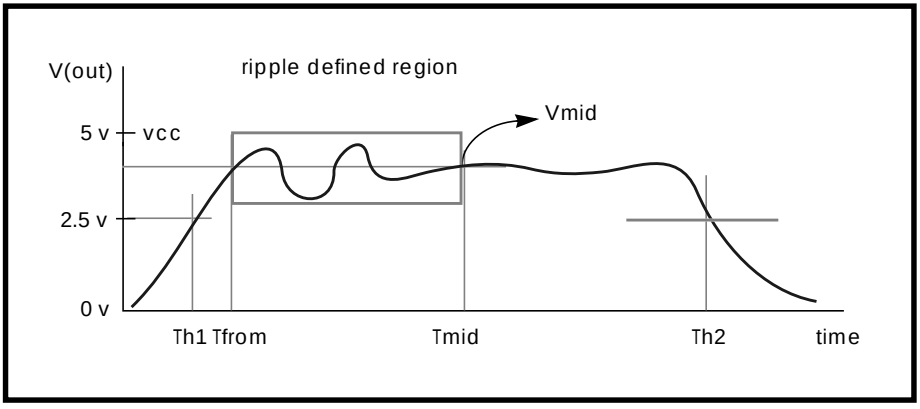

Figure 19-2: Waveform to Demonstrate Ripple Calculation

Sigma Sweep versus Delay

This file is set up to sweep sigma of the model parameter distribution while looking at the delay, giving the designer the delay derating curve for the model worst cases. This example is based on the demonstration file in *$installdir/demo/hspice/cchar/sigma.sp.* This technique of building a worst case sigma library is described in “Performing Worst Case Analysis” on page 10-33.

Example:

```
.tran 20p 1.0n sweep sigma -3 3 .5
.meas m_delay trig v(2) val=vref fall=1 targ v(4) val=vref
fall=1
.param xlnew =’polycd-sigma*0.06u’ toxnew=’tox-sigma*10’
.model nch nmos level=28 xl = xlnew tox=toxnew
```

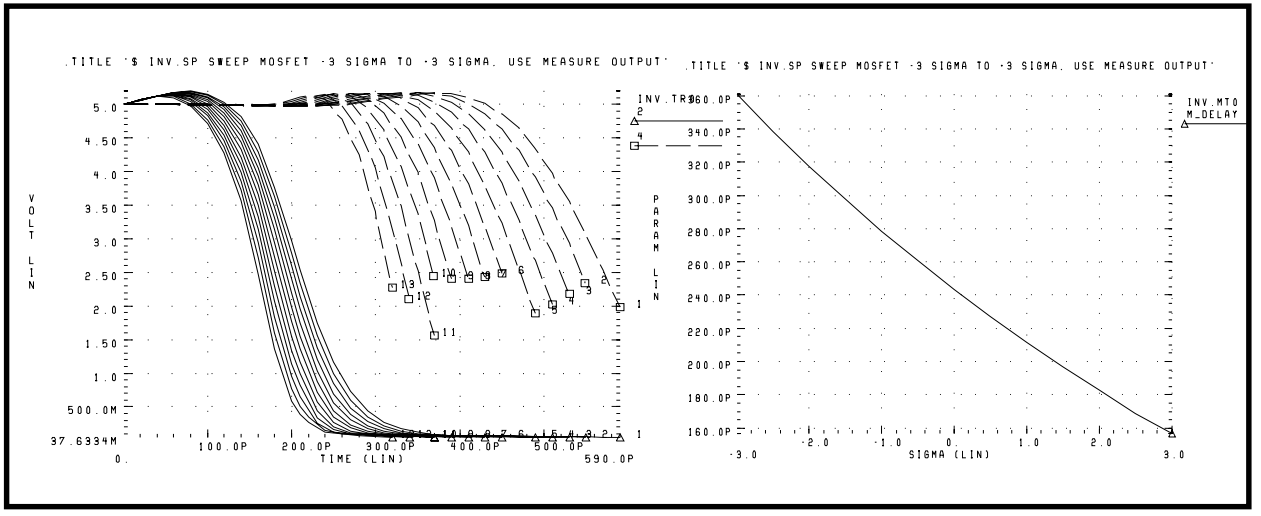

Figure 19-3: Inverter Pair Transfer Curves and Sigma Sweep vs. Delay

Delay versus Fanout

This example sweeps the subcircuit multiplier to quickly generate a family of five load curves. By buffering the input source with one stage, more accurate results are obtained. The example calculates the mean, variance, sigma, and average deviance for each of the second sweep variables (*m_delay* and *rms_power*). This example is based on the demonstration file *$installdir/demo/hspice/cchar/load1.sp*.

Input File Example

```
.tran 100p 2.0n sweep fanout 1 10 2
.param vref=2.5
.meas m_delay trig v(2) val=vref fall=1
+ targ v(3) val=vref rise=1
.meas rms_power rms power

x1 in 2 inv
x2 2 3 inv
x3 3 4 inv m=fanout
```

Output Statistical Results

```
meas_variable = m_delay
mean = 273.8560p varian = 1.968e-20
sigma = 140.2711p avgdev = 106.5685p

meas_variable = rms_power
mean = 5.2544m varian = 8.7044u
sigma = 2.9503m avgdev = 2.2945m
```

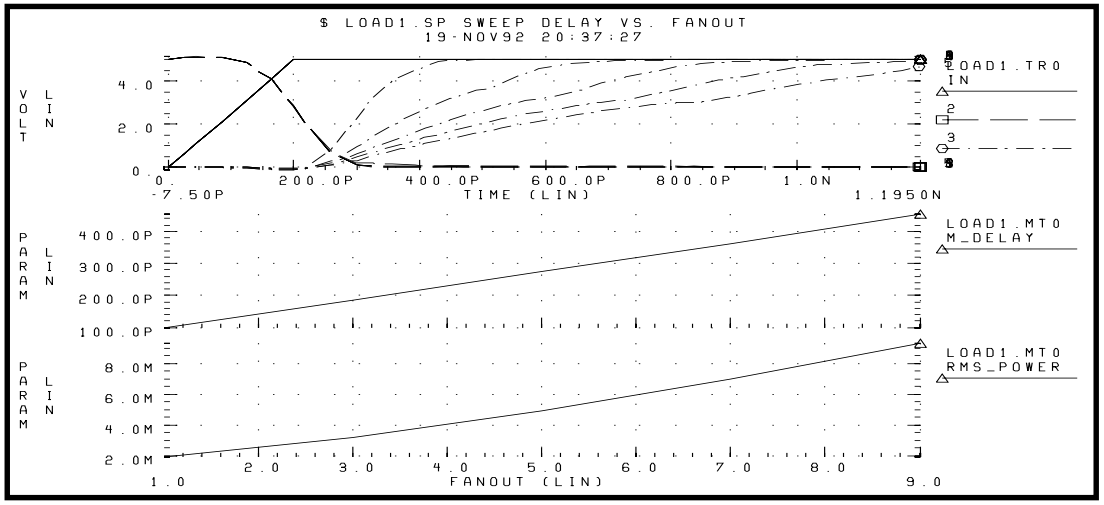

Figure 19-4: Inverter Delay and Power versus Fanout

Pin Capacitance Measurement

This example shows the effect of dynamic capacitance at the switch point. It sweeps the DC input voltage (*pdcin*) to the inverter and performs an AC analysis each 0.1 volt. The measure parameter *incap* is calculated from the imaginary current through the voltage source at the 10 kilohertz point in the AC curve (not shown). The peak capacitance at the switch point occurs when the voltage at the output side is changing in the opposite direction from the input side of the Miller capacitor, adding the Miller capacitance times the inverter gain to the total effective capacitance.

Example

```
mp out in 1 1 mp w=10u l=3u
mn out in 0 0 mn w=5u l=3u
vin in 0 DC= pdcin AC 1 0

.ac lin 2 10k 100k sweep pdcin 0 5 .1
.measure ac incap find par( '-1 * ii(vin)/(hertz*twopi)' )
AT=10000hertz
```

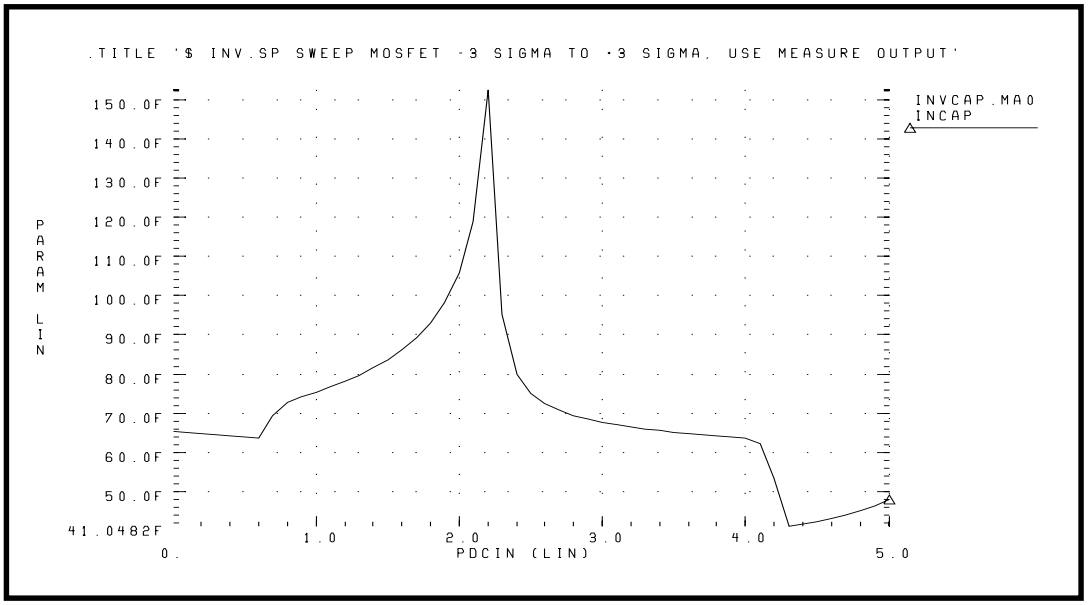

Figure 19-5: Graph of Pin Capacitance versus Inverter Input Voltage

Op-amp Characterization of ALM124

This example analyzes op-amps with .MEASURE statements to present a very complete data sheet. It references op-amp circuit output node out0 in the four .MEASURE statements using output variable operators for decibels vdb(out0), voltage magnitude vm(out0), and phase vp(out0). The example is taken from the demonstration file *demo/apps/alm124.sp.*

Input File Example

```
.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)
```

Measure Results

```
unitfreq = 9.0786E+05 targ= 9.0786E+05 trig= 1.0000E+00
phasemargin = 6.6403E+01
gain(db) = 9.9663E+01 at= 1.0000E+00 from= 1.0000E+00 to=
1.0000E+07
gain(mag)= 9.6192E+04 at= 1.0000E+00 from= 1.0000E+00 to=
1.0000E+07
```

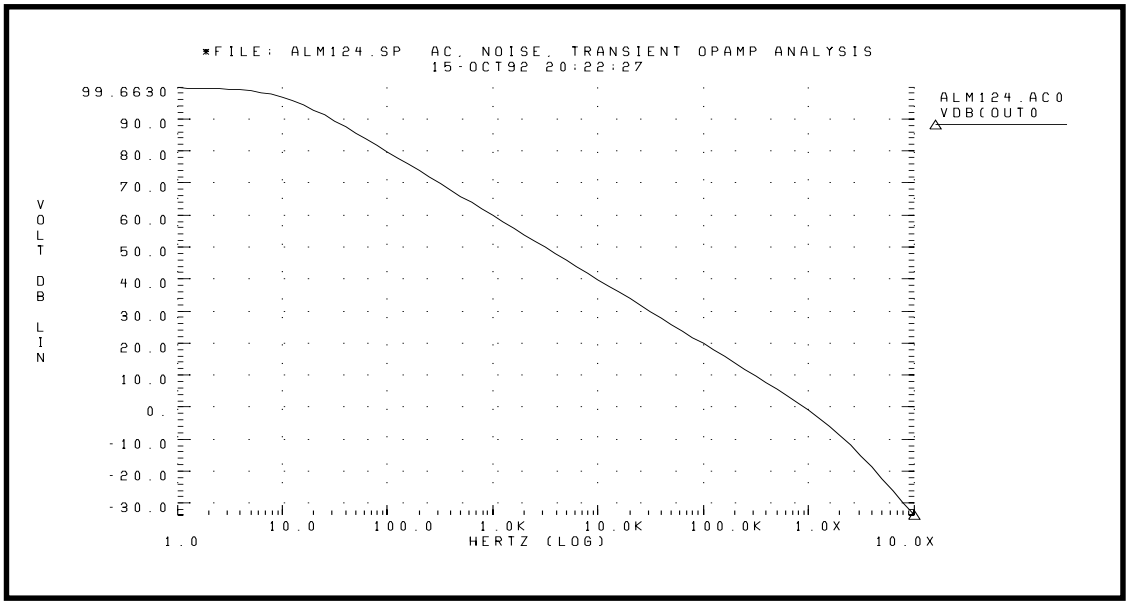

Figure 19-6: Magnitude Plot of Op-Amp Gain

Performing Data Driven Analysis

Data driven analysis allows you to simultaneously modify any number of parameters, then perform an operating point, DC, AC, or transient analysis.The parameter value array is either included in the simulation input file directly or is stored as an external ASCII file. The .DATA statement associates the parameters with the value array, and it replaces the .PARAM statement.

Data driven analysis requires a .DATA statement and an analysis statement that contains a DATA=dataname keyword.

Note: The .DATA statement format is almost the same as the measure output format. It does not require the "+" sign to continue a line.

The syntax is:

```
.DATA dataname pname1 pname2 ... pnamen
val11 val12 ... val1n
   ...
valm1 valm2 ... valmn
.ENDDATA
```

or

```
.DATA dataname Merge file=filename pname1=column1
pname2=column2 ...
.ENDDATA
```

The first general form specifies *n* parameters with *m* iterations. The number of parameters is not limited, and this number determines the number of values taken per iteration. There is no requirement that all parameter values be on the same physical line. The .DATA statement syntax is covered in greater detail in *Chapter 4, Specifying Simulation Output*.

The syntax is:

Operating point:

```
.DC DATA=dataname
```

DC sweep:

```
.DC vin 1 5 .25 SWEEP DATA=dataname
```

AC sweep:

```
.AC dec 10 100 10meg SWEEP DATA=dataname
```

TRAN sweep:

```
.TRAN 1n 10n SWEEP DATA=dataname
```

The .MEASURE statement computes rise, fall, and propagation delay times. For cell characterization, many measurements are necessary because of changing specifications for load capacitance, fanout, temperature, and so on. This process can be very time consuming. Use the AUTOSTOP option to stop the transient analysis when all the rise, fall, and delays specified in the .MEASURE statements are calculated. This option saves a great deal of CPU time.

Cell Characterization Example

This section provides example input files that perform cell characterization of an inverter based on 3-micron MOSFET technology. The program finds the propagation delay and rise and fall times for the inverter for best, worst and typical cases for different fanouts. This data then can be used as library data for digital-based simulators such as those found in the simulation of gate arrays and standard cells.

The example, taken from the demonstration file *$installdir/demo/hspice/apps/cellchar.sp*, demonstrates the use of the .MEASURE statement, the .DATA statement, and the AUTOSTOP option in the characterization of a CMOS inverter.Figure 19-7: and Figure 19-8: are identical except that their input signals are complementary. The circuit in Figure 19-7: calculates the rise time and the low-to-high propagation delay time. The circuit in Figure 19-8: calculates the fall time and the high-to-low propagation delay time. When only one circuit is used, CPU time increases because the analysis time increases to calculate both rise and fall times.

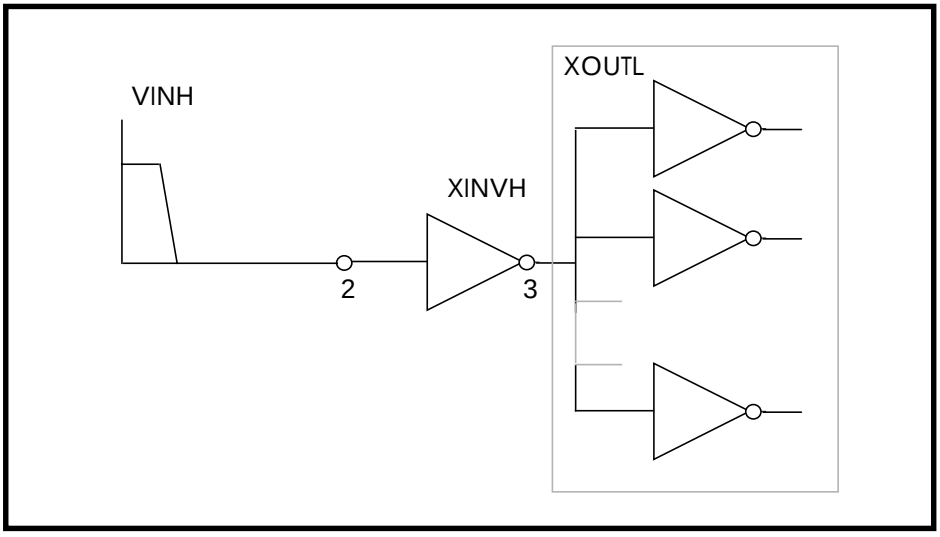

Figure 19-7: Cell Characterization Circuit 1

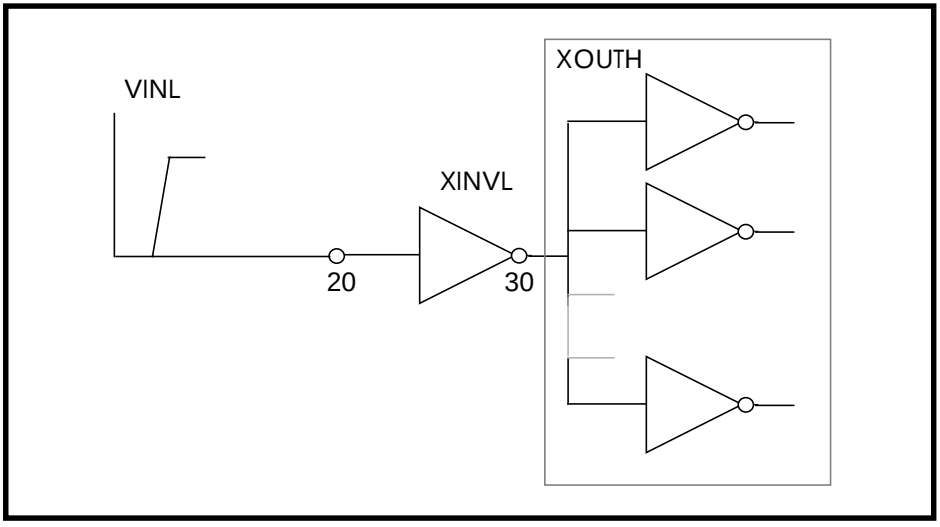

Figure 19-8: Cell Characterization Circuit 2

The subcircuit XOUTL or XOUTH represents the fanout of the cell (inverter). Star-Hspice modifies fanout by specifying different multipliers (m) in the subcircuit calls.

Star-Hspice also provides local and global temperature specifications. This example characterizes the cell at global temperature 27, while devices M1 and M2 are at temperature (27+DTEMP). The .DATA statement specifies the DTEMP value.

The example uses a transient parameterized sweep with the .DATA and .MEASURE statements to determine the timing of the inverter for best, typical and worst cases. The parameters varied include power supply, input rise and fall time, fanout, MOSFET temperature, n-channel and p-channel threshold, and both the drawn width and length of the MOSFET. Use the AUTOSTOP option to speed simulation time and work with the .MEASURE statement. Once the .MEASURE statement determines the parameter to be measured, the AUTOSTOP option terminates the transient sweep, even though it has not completely swept the transient sweep range specified.

The .MEASURE statement uses quoted string parameter variables to measure the rise and fall times, as well as the propagation delays. Rise time starts when the voltage at node 3 (the output of the inverter) is equal to 0.1 · VDD (that is, V(3) = 0.1VDD) and ends when the voltage at node 3 is equal to 0.9 · VDD (that is, V(3) = 0.9VDD).

For more accurate results, start the .MEASURE calculation after a time delay, a simulation cycle specifying delay time in the .MEASURE statement, or in the input pulse statement.

The following example features:

- AUTOSTOP and .MEASURE statements
- Mean, variance, sigma and avgdev calculations
- Circuit and element temperature
- Algebraic equation handling
- PAR( ) as output variable in the .MEASURE statement
- Subcircuit parameter passing and subcircuit multiplier
- .DATA statement

Example Input Files

```
FILE: CELLCHAR.SP
*
.OPTIONS SPICE NOMOD AUTOSTOP
.PARAM TD=10N PW=50N TRR=5N TRF=5N VDD=5 LDEL=0 WDEL=0
+ NVT=0.8 PVT=-0.8 DTEMP=0 FANOUT=1
.GLOBAL VDD
* — global supply name
.TEMP 27
```

SUBCKT Definition

```
.SUBCKT INV IN OUT
M1 OUT IN VDD VDD P L=3U W=15U DTEMP=DTEMP
M2 OUT IN 0 0 N L=3U W=8U DTEMP=DTEMP
CL OUT 0 200E-15 .001
CI IN 0 50E-15 .001
.ENDS
```

SUBCKT Calls

```
XINVH 2 3 INV      $—- INPUT START HIGH
XOUTL 3 4 INV M=FANOUT
XINVL 2030 INV      $—- INPUT START LOW
XOUTH 30 40INV M=FANOUT
* — INPUT VOLTAGE SOURCES
VDD VDD 0 VDD
VINH 2  0 PULSE(VDD,0,TD,TRR,TRF,PW,200NS)
VINL 20 0 PULSE(0,VDD,TD,TRR,TRF,PW,200NS)
* — MEASURE STATEMENTS FOR RISE, FALL, AND PROPAGATION DELAYS
.MEAS RISETIME  TRIG PAR('V(3) -0.1*VDD')  VAL=0  RISE=1
+               TARG PAR('V(3) -0.9*VDD')  VAL=0  RISE=1
.MEAS FALLTIME  TRIG PAR('V(30)-0.9*VDD')  VAL=0  FALL=1
+               TARG PAR('V(30)-0.1*VDD')  VAL=0  FALL=1
.MEAS TPLH      TRIG PAR('V(2) -0.5*VDD')  VAL=0  FALL=1
+               TARG PAR('V(3) -0.5*VDD')  VAL=0  RISE=1
.MEAS TPHL      TRIG PAR('V(20)-0.5*VDD')  VAL=0  RISE=1
+               TARG PAR('V(30)-0.5*VDD')  VAL=0  FALL=1
* — ANALYSIS SPECIFICATION
.TRAN 1N 500N SWEEP DATA=DATNM
* — DATA STATEMENT SPECIFICATION
```

```
.DATA DATNM
VDD TRR TRF FANOUT DTEMP  NVT   PVT   LDEL  WDEL
5.0  2N 2N    2      0    0.8  -0.8   0     0     $ TYPICAL
5.5  1N 1N    1    -80    0.6  -0.6  -0.2U  0.2U  $ BEST
4.5  3N 3N   10    100    1.0  -1.0  +0.2U -0.2U  $ WORST
5.0  2N 2N    2      0    1.0  -0.6   0     0     $ STRONG P,
WEAK N
5.0  2N 2N    2      0    0.6  -1.0   0     0     $ WEAK P,
STRONG N
5.0  2N 2N    4      0    0.8  -0.8   0     0     $ FANOUT=4
5.0  2N 2N    8      0    0.8  -0.8   0     0     $ FANOUT=8
.ENDDATA
```

Models

```
.MODEL N NMOS LEVEL=2 LDEL=LDEL WDEL=WDEL
+ VTO=NVT    TOX =300     NSUB=1.34E16  UO=600
+ LD=0.4U    WD  =0.6U    UCRIT=4.876E4 UEXP=.15
+ VMAX=10E4  NEFF=15      PHI=.71       PB=.7
+ RS=10      RD  =10      GAMMA=0.897   LAMBDA=0.004
+ DELTA=2.31 NFS =6.1E11  CAPOP=4
+ CJ=3.77E-4 CJSW=1.9E-10 MJ=.42 MJSW=.128
*
.MODEL P PMOS  LEVEL=2 LDEL=LDEL WDEL=WDEL
+ VTO=PVT     TOX=300      NSUB=0.965E15 UO=250
+ LD=0.5U     WD=0.65U     UCRIT=4.65E4  UEXP=.25
+ VMAX=1E5    NEFF=10      PHI=.574      PB=.7
+ RS=15       RD=15        GAMMA=0.2     LAMBDA=.01
+ DELTA=2.486 NFS=5.2E11   CAPOP=4
+ CJ=1.75E-4  CJSW=2.3E-10 MJ=.42        MJSW=.128
.END
```

A sample of measure statements is printed:

```
***  MEASURE STATEMENT RESULTS FROM THE FIRST ITERATION ($
TYPICAL)
RISETIME  =  3.3551E-09  TARG=  1.5027E-08  TRIG=  1.1672E-08
FALLTIME  =  2.8802E-09  TARG=  1.4583E-08  TRIG=  1.1702E-08
TPLH      =  1.8537E-09  TARG=  1.2854E-08  TRIG=  1.1000E-08
TPHL      =  1.8137E-09  TARG=  1.2814E-08  TRIG=  1.1000E-08
```

```
***  MEASURE STATEMENT RESULTS FROM THE LAST ITERATION ($
FANOUT=8)
RISETIME  =  8.7909E-09  TARG=  2.0947E-08  TRIG=  1.2156E-08
FALLTIME  =  7.6526E-09  TARG=  1.9810E-08  TRIG=  1.2157E-08
TPLH      =  3.9922E-09  TARG=  1.4992E-08  TRIG=  1.1000E-08
TPHL      =  3.7995E-09  TARG=  1.4800E-08  TRIG=  1.1000E-08

MEAS_VARIABLE = RISETIME
MEAN  =  6.5425E-09  VARIAN =  4.3017E-17
SIGMA =  6.5588E-09  AVGDEV =  4.6096E-09

MEAS_VARIABLE = FALLTIME
MEAN  =  5.7100E-09  VARIAN =  3.4152E-17
SIGMA =  5.8440E-09  AVGDEV =  4.0983E-09

MEAS_VARIABLE = TPLH
MEAN  =  3.1559E-09  VARIAN =  8.2933E-18
SIGMA =  2.8798E-09  AVGDEV =  1.9913E-09

MEAS_VARIABLE = TPHL
MEAN  =  3.0382E-09  VARIAN =  7.3110E-18
SIGMA =  2.7039E-09  AVGDEV =  1.8651E-0
```

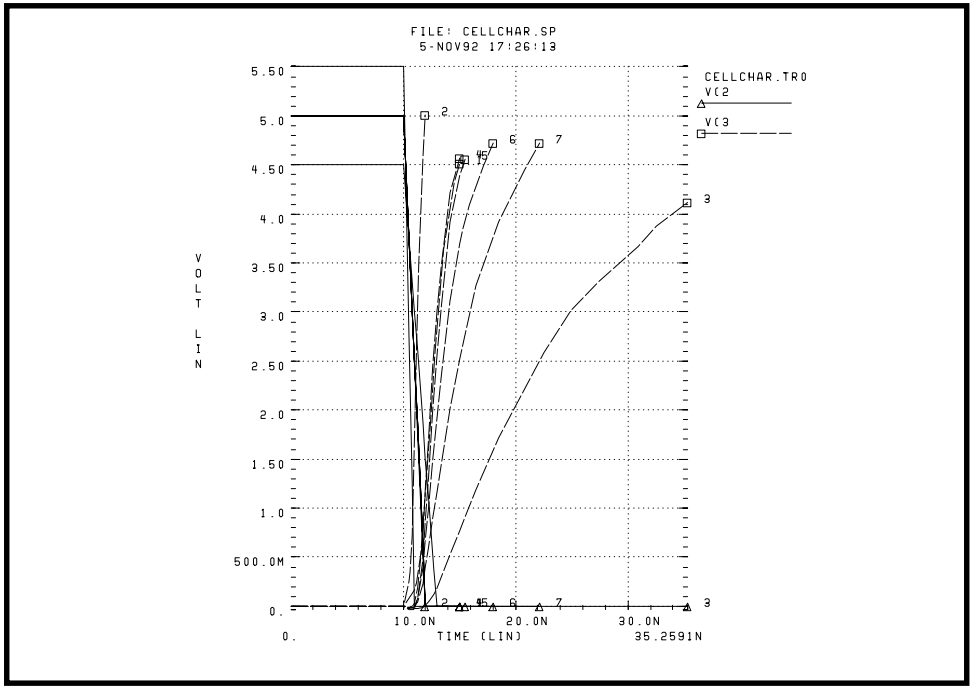

Figure 19-9: Plotting the Simulation Outputs

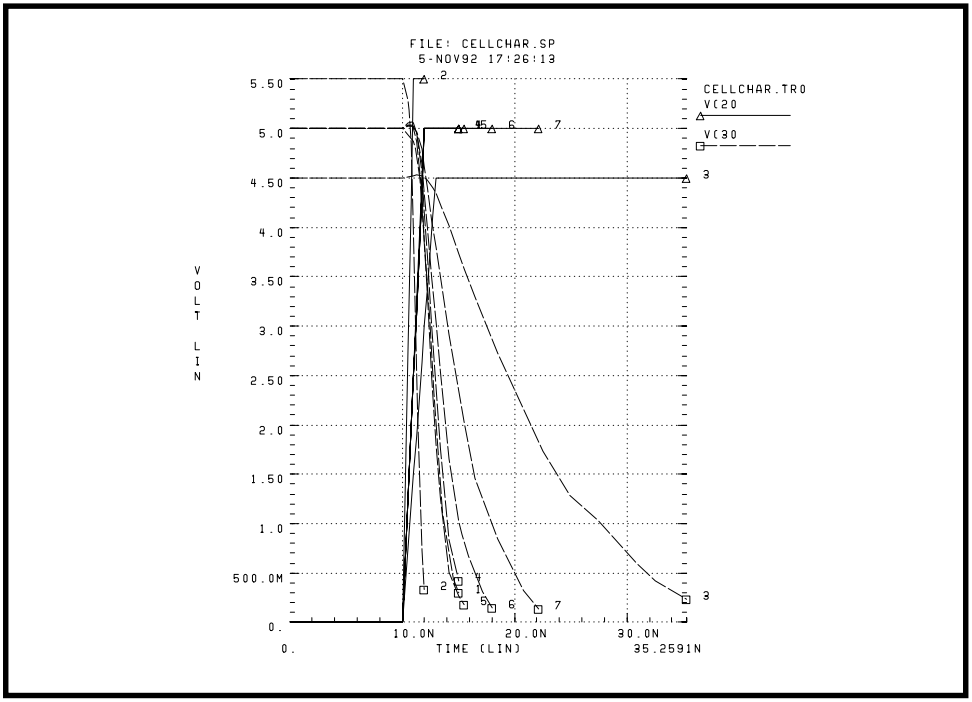

Figure 19-10: Verifying the Measure Statement Results by the Plots

Using Digital File Input Stimuli

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. The example outputs a digital file.

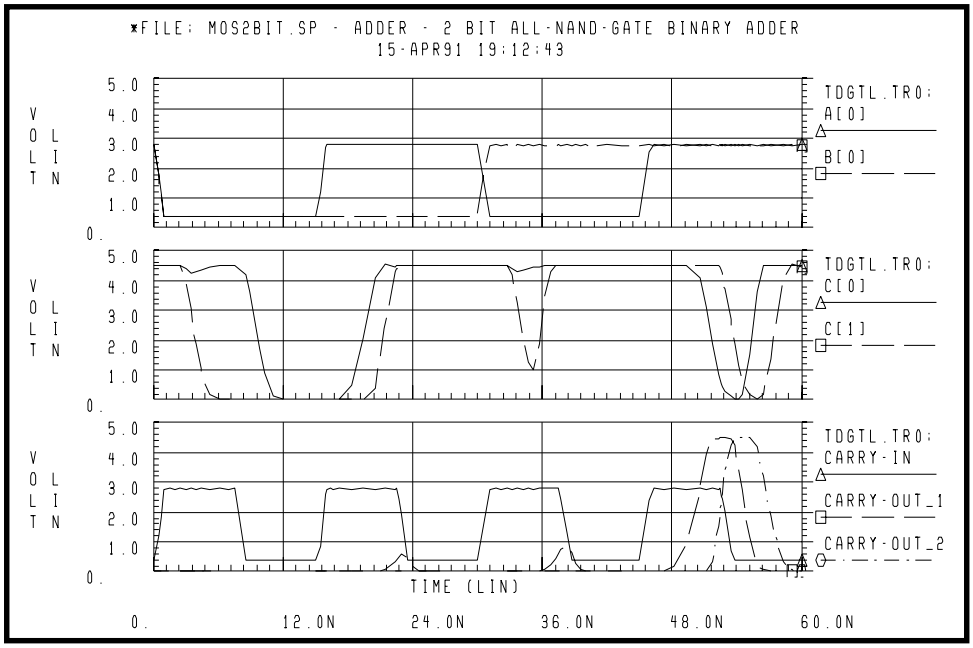

Figure 19-11: Digital Stimuli File Input

The simulation above uses the digital-to-analog interface model. The example, found in the demo directory *$installdir/demo/hspice/cchar/tdgtl.sp*, shows a way of generating stimuli using an external stimuli file produced by a logic simulator (in this case, the Viewsim simulator).

Top-down design generally starts with a system-level hardware description language (HDL) description of the circuit. This is decomposed to the logic cell level, and cells are then synthesized into transistor level circuits. Since simulation has been done at the logic level, it is possible to capture all of the basic input stimuli to the cell. With the integrated 28-state logic interface, Star-Hspice enables you to reuse the logic output as circuit simulator stimuli input.

Replacing Sources With Digital Inputs

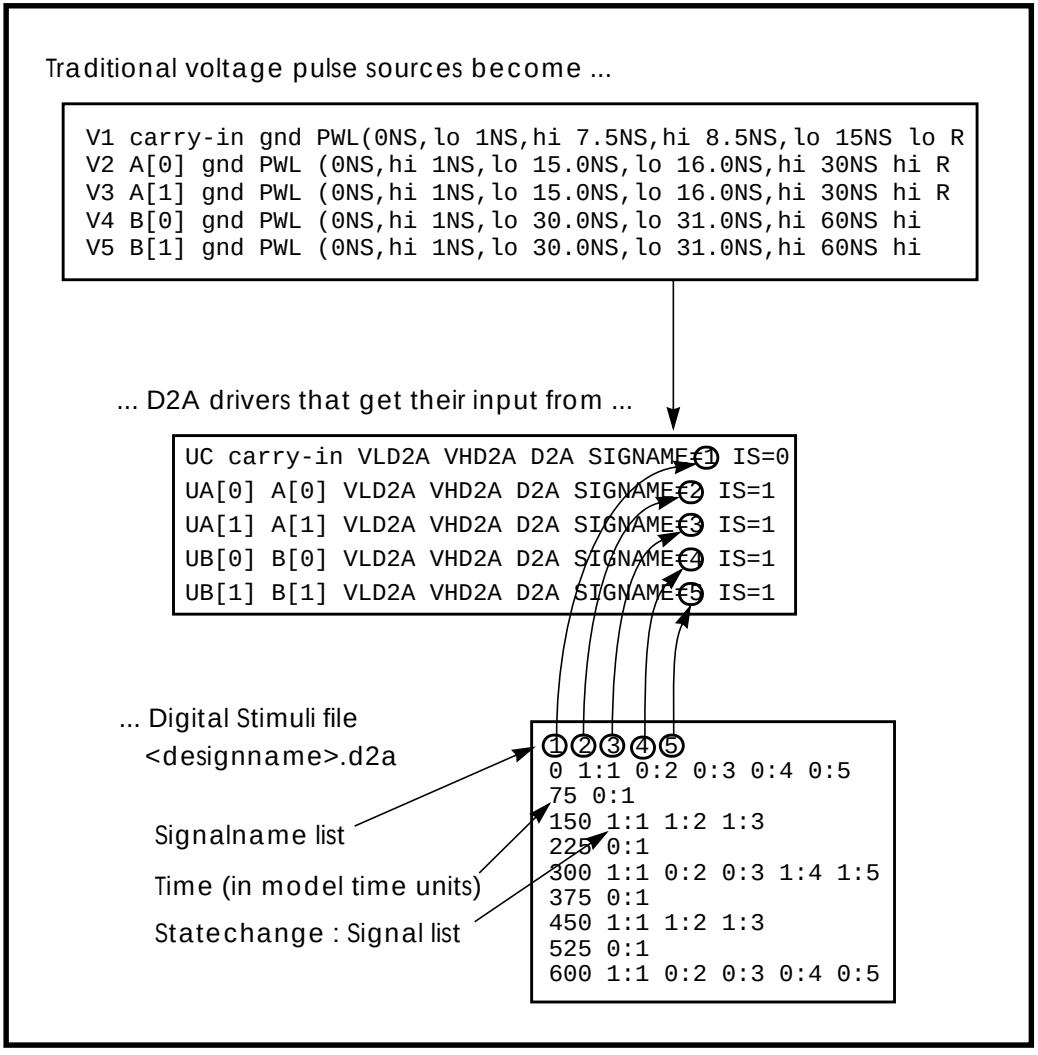

Figure 19-12: Digital File Signal Correspondence

Example

```
.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
```