HSPICE - Highlights and Introductions

Techniques for SI

09-17-03

Features Use for SI

- Parameters
- Alters
- Libraries
- Syntax based – NO GUI
- Self documenting ASCII node names
- Voltage Controlled Resistor
- Monte Carlo
- Node equation based source
- PWL linear based source
- Nodal measurement produce measurement files
- Accurate Transmission lines with W elements
- Frequency dependant transmission lines Transient
- IBIS buffers

Good Practices

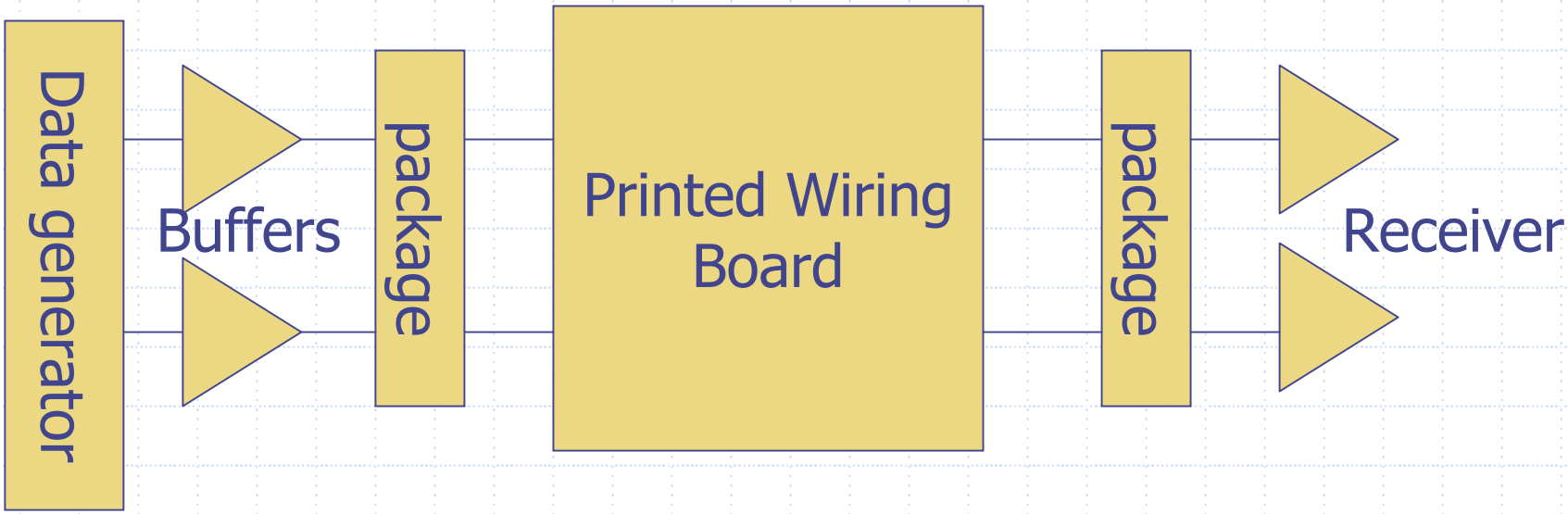
- Modularize with sub-circuits and/or libraries!
- Circuit text should flow line a drawing.
  - Don't put all caps, resistors, and transmission lines respective separate sections.
- Most SI circuits are composed of data generator, buffers, transmission lines, package models, and connectors.

Global, Local, and Position

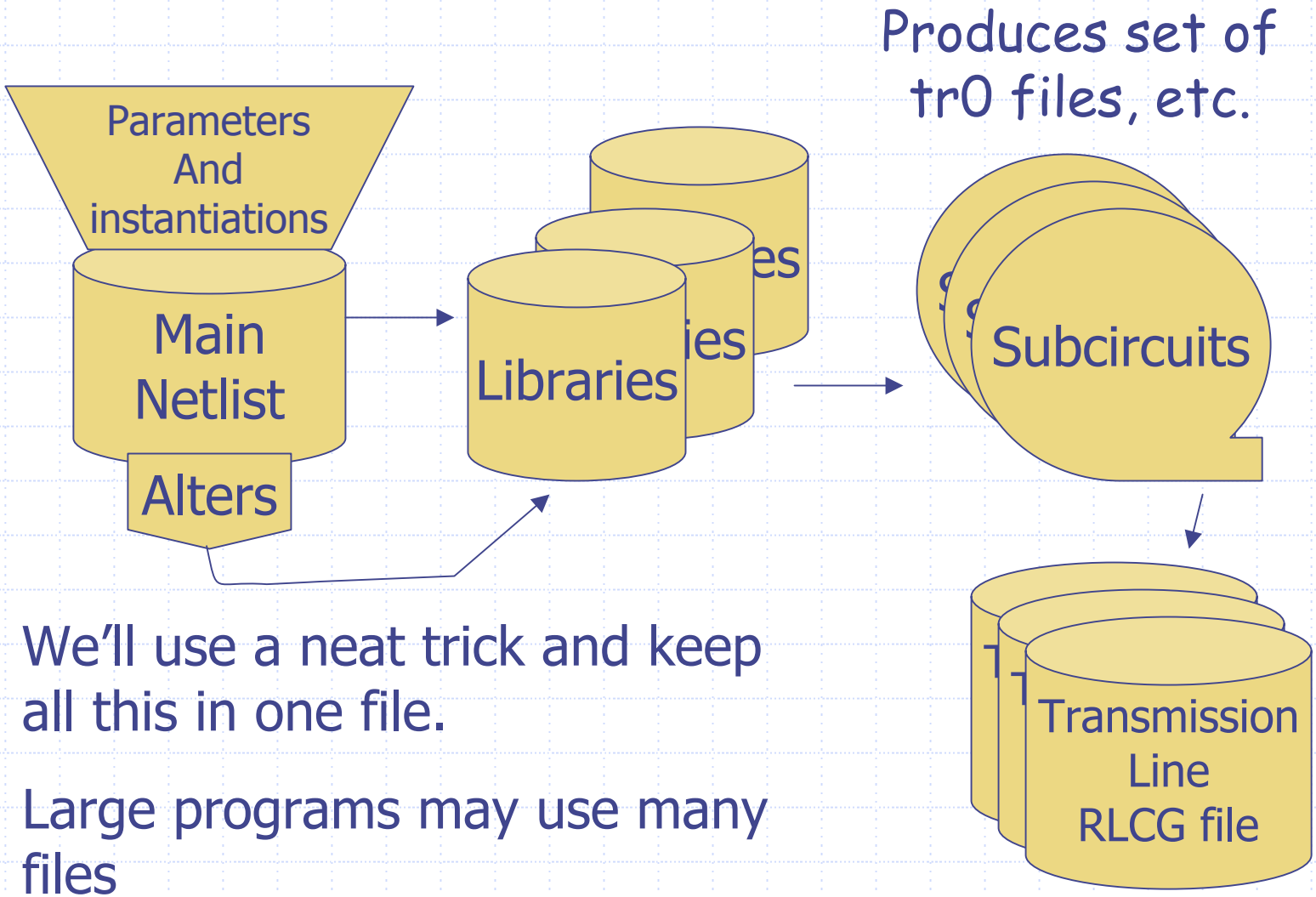
- Circuit elements within a sub-circuit or the main net list are position insensitive.
  - Good-news/bad news
  - It is easier to follow elements whose code traces out the circuit.
- In general parameters are global unless passed into a sub-circuit.
- Parameters are not positions insensitive
  - Treat definition of parameters as last reference wins the definition.
  - This can be trick to determine for complex decks.
  - Deck is old terminology that comes form "punch card decks"
- Make the first 8 characters of library names unique
- Most HSPICE is case insensitive.
  - The exception is libraries and file names that are enclosed in single quotes

Top Level Program

- First step is to draw as simulation block diagram.
- The following slides are a learn by example method
- We will review some common HSPICE elements used for signal integrity



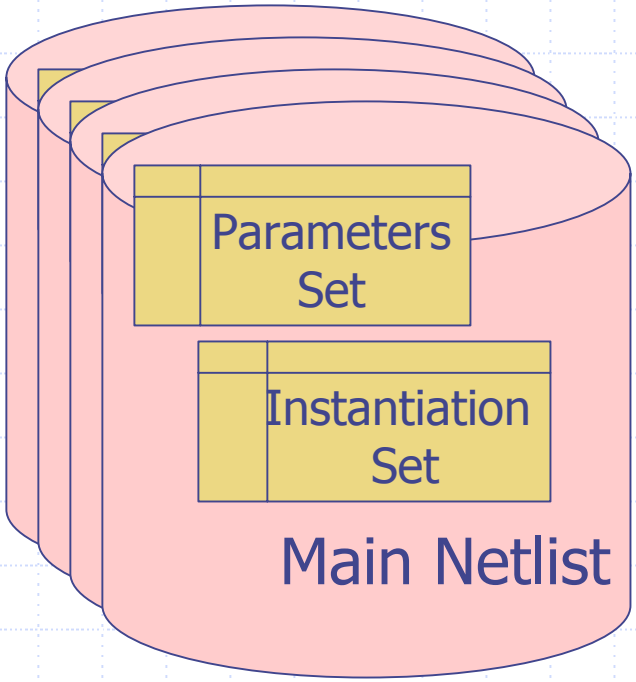
Structure I - We will use this one for now



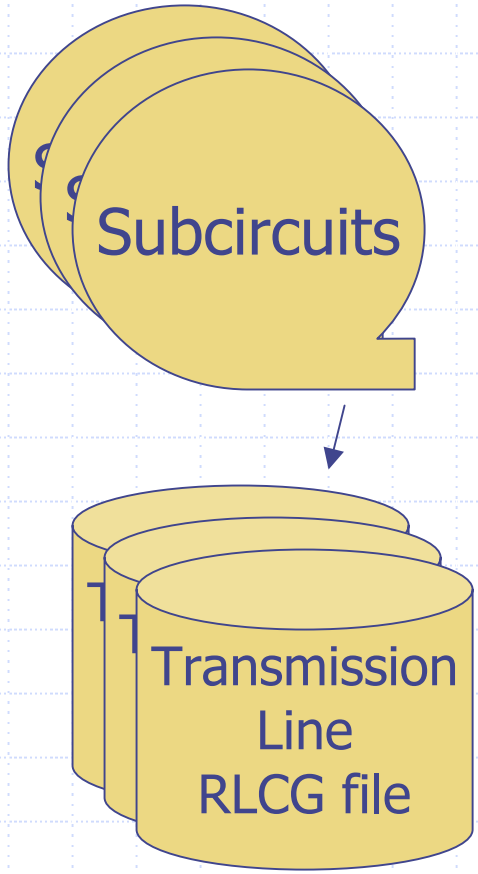
We'll use a neat trick and keep all this in one file.

Large programs may use many files

Structure II

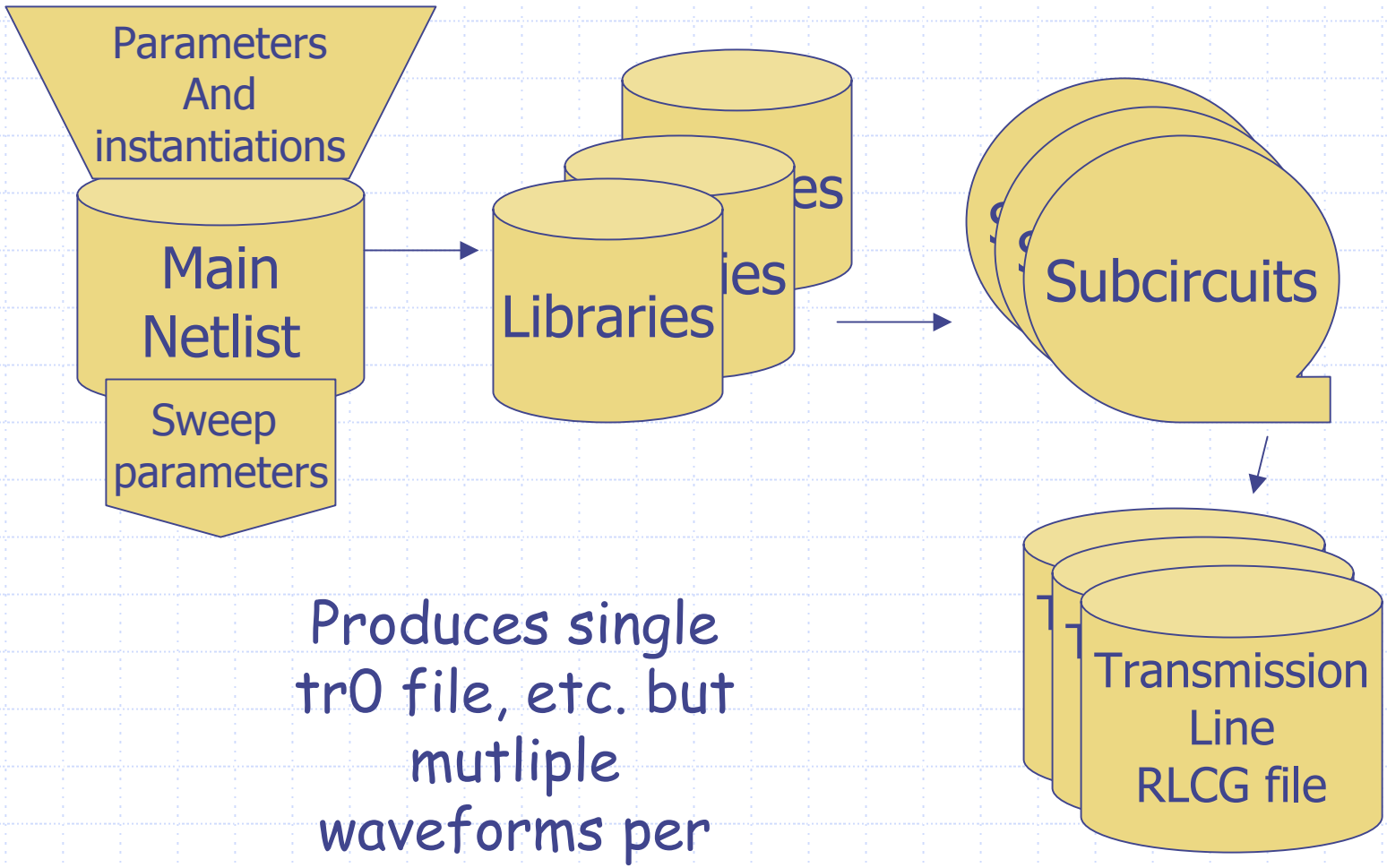


Produces tr0, tr1, tr2, … files etc.



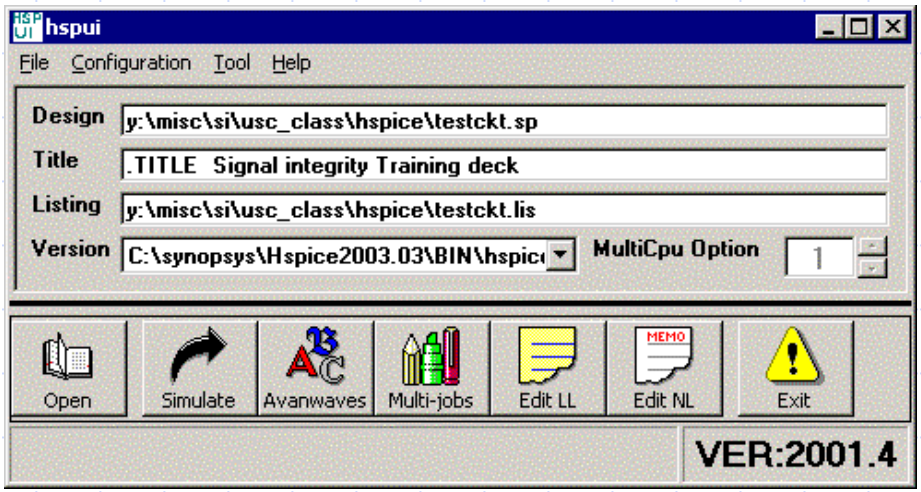
Each Simulation case is a different cataloged file

Structure III



Produces single tr0 file, etc. but mutliple waveforms per file

Running the netlist



- Clicking on simulate will create the following files
  - Transient analysis nodal file – testckt.tr0
    - This is where the waveform data is
  - Measurement results file – testckt.mt0
  - List file – testckt.lis
    - Edit this to debug errors
  - Initial conditions file – testckt.in0
  - Sub-circuit cross reference list – testckt.pa0
  - Output status file – testckt.st0

Data Generator

```
.LIB                    'pulse'
.SUBCKT         DATAS   bit1      bit2      datarate=-1
V1 bit1 0 PULSE 0v 1v 0n 0.5n 0.5n 'datarate- 0.5n' '2*datarate'
V2 bit2 0 PULSE 0v 1v 0n 0.5n 0.5n 'datarate- 0.5n' '2*datarate'
.ENDS
.ENDL
```

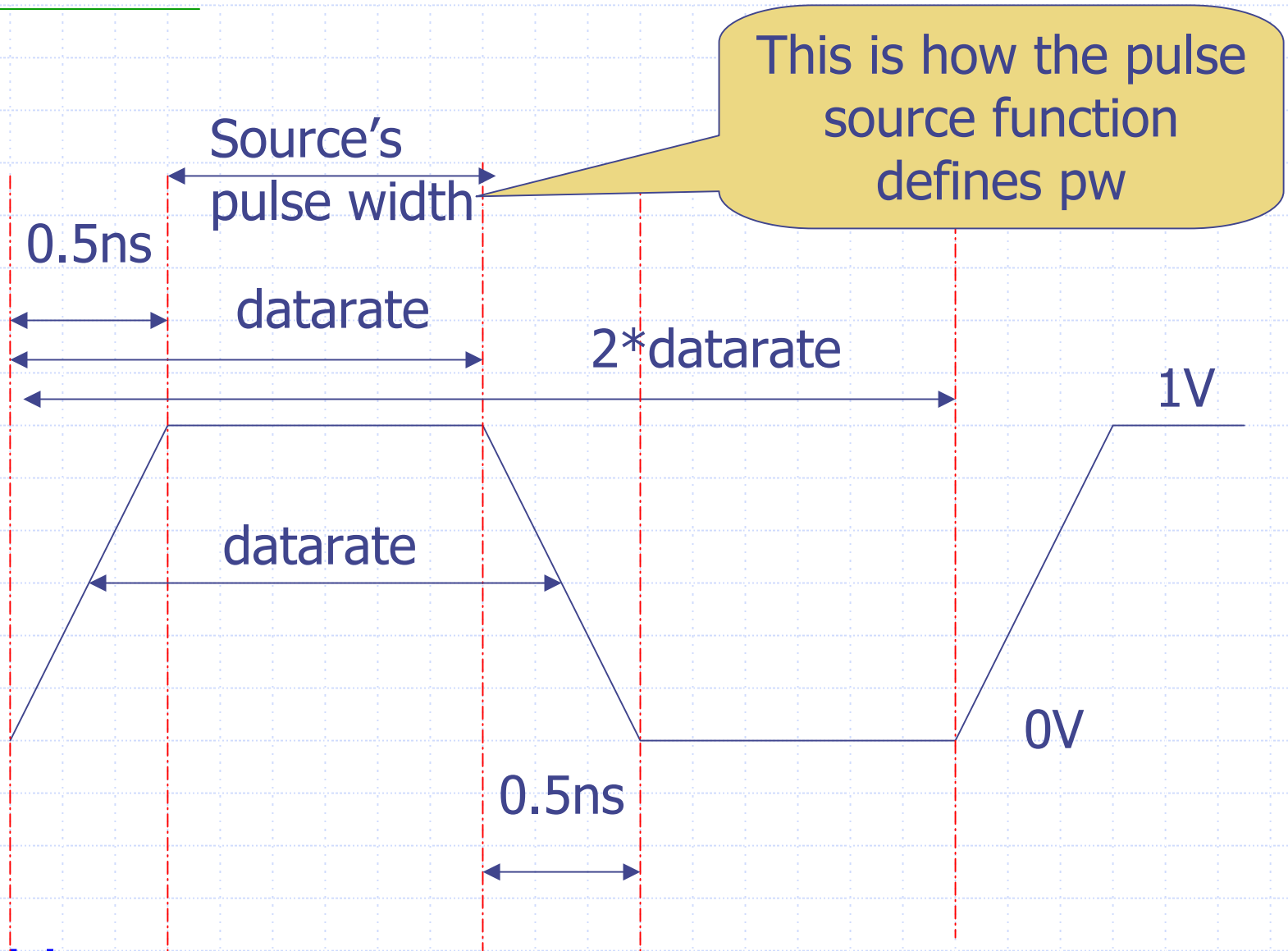
- Bit1 and Bit2 are data stream outputs for this sub-circuit
- "datarate" is passed from the call site
  - Note that a subcircuit is analogous to a software subroutine
  - "datarate" is set to "-1" to force an error if the parameter was not passed.
- This pulse generator example produces a 1V Aggressor and victim w/ 500 ps rise/fall time.
  - The pulse width is "datarate" adjusted by the risetime.
  - The period is 2*datarate
  - This special case uses 0v and 1v as a bit stream which has advantages that we will learn later in behavioral modeling.

Parameterized Generator

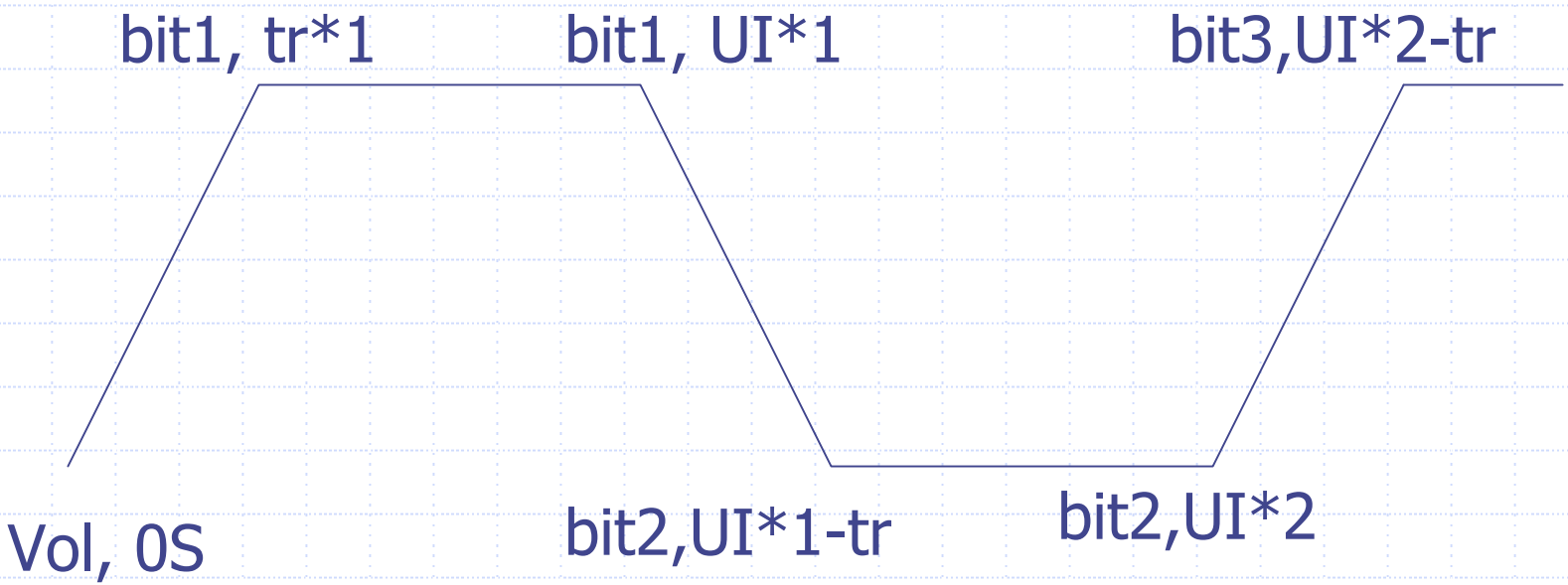
```
.LIB                    'pulse'
.SUBCKT         DATAS   bit1      bit2      datarate=-1
V1 bit1 0 PULSE 0v 1v 0n tr tr 'datarate-tr' '2*datarate'
V2 bit2 0 PULSE 0v 1v 0n tr tr 'datarate-tr' '2*datarate'
.ENDS
.ENDL
```

- We can replace the 0.5n entries with a parameter called tr. (equal rise/fall)
- We can set this parameter in the main net list as follows: .PARAM tr=0.5n
- Notice the difference between the two parameters tr and datarate

Square Wave in Previous



Piece Wise Linear Source



*rise/fall time = tr

Assignment

- Create same driver with a PWL source and with data pattern "101100110".
- Assume all parameter except datarate are global
- Parameterize bits as bit0, bit1, bit2…
- Parameter for rise and fall time with a signal parameter Tr
- Write separate code for parameter statements in the main netlist.

Driver Sub-circuit

```
.LIB      'driver'
.SUBCKT   MYBUF      in       out      Vss
Edrive    out1       Vss      in       0 VOL='(Voh-Vol)*V(in)+Vol'
Rout out1 out 50
Cout out Vss 1p
.ENDS
.ENDL
```

- This example just uses the bits on node "in" and creates an output voltage with Vol and Vol one node "out".
- Vol and Voh are global in this case because they were not passed
- This example uses a equation controlled voltage source. This a very powerful feature.
  - The equation is enclosed in quotes much the same why a parameter equation is.
- This entire subcircuit can be replaced at a later time with a transistor based buffer model or an IBIS model.
- The source impedance in this case is 50 ohms with a pF across the output terminal "out"

Parameterize Driver

```
.LIB                'driver`
.SUBCKT     MYBUF   in      out         Vss
Edrive      out1    Vss     VOL='(Voh-Vol)*V(in)+Vol`
Rout        out1    out      Tx_rterm
Cout        out1    Vss      Tx_cterm
Rin in vss 1G
.ENDS
.ENDL
```

- Change the source terminations into parameters: Tx_rterm and Rx_cterm
- As a good practice place a hi impedance DC path across input.
  - This can avoid transient errors.
- We can set this parameter in the main net list as follows:

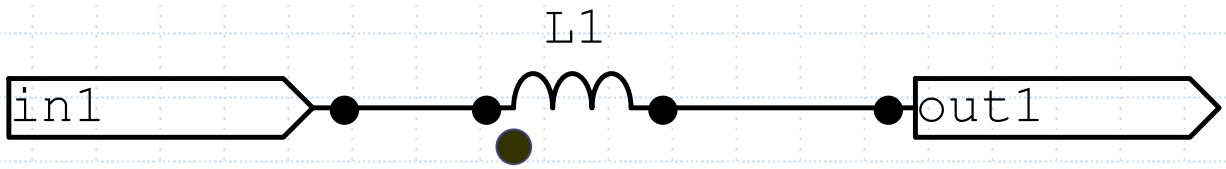
  .PARAM Tx_rterm=50 Tx_Cterm=1pF

Package Sub-circuit

```
.LIB          'LC_pack'
.SUBCKT             PKG   in1    in2    out1   out2   Vss
L1      in1    out1   1n
L2      in2    out2   1n
K1      L1     L2     0.2
.ENDS
.ENDL
```

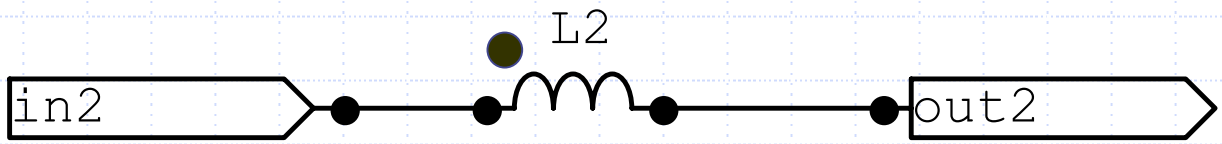
- This is a simple package that uses a coupled inductor circuit.
- Often this subcircuit is more complex and derive from tools like Ansoft

Coupled Inductors



$$K = \frac{L12}{\sqrt{L1 \cdot L2}}$$

Where L12 is the mutual inductance between inductor L1 and L2



Printed wiring board modeling

```
.LIB            'easy_lines'
.SUBCKT                 BRD     in1     in2     out1    out2    Vss
Wline1      in1         in2     Vss     out1    out2    Vss
+             RLGCFILE=     's5_z068.9_z0d108.8'    N=2     L=0.1
.ENDS
.ENDL
```

- Board etches can be accurately modeled with W-elements.
- For that case we use coupled transmission lines for the board traces
- The file 's5_z068.9_z0d108.8.rlc' contains the transmission line characteristics. This data may be created with internal HSPICE 2-D field solver or any other 2-D field solver such as Ansoft.
- The symbol "+" is a continuation line.
- Often this subcircuit can become quite substantial containing many transmission lines and board features modeled as passive elements.

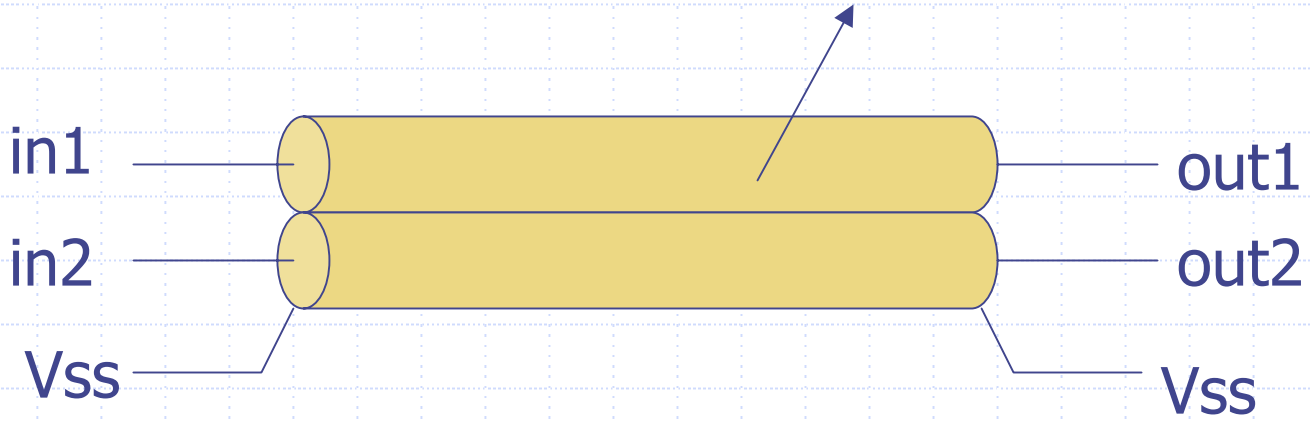
W-element: Model Reference

```
.LIB                'easy_lines'
.SUBCKT                 BRD    in1     in2     out1    out2    Vss
Wline1      in1         in2    Vss     out1    out2    Vss
+       RLGCMODEL=  ' s5_z068.9_z0d108.8 '          N=2     L=.1
.ENDS
.ENDL
```

- Additionally a model statement can be used to specify RLGC data.

Transmission Line W-Element

```
Wline1          in1   in2   Vss   out1   out2   Vss
+ RLGCMODEL='s5_z068.9_z0d108.8' N=2  L=0.1
```

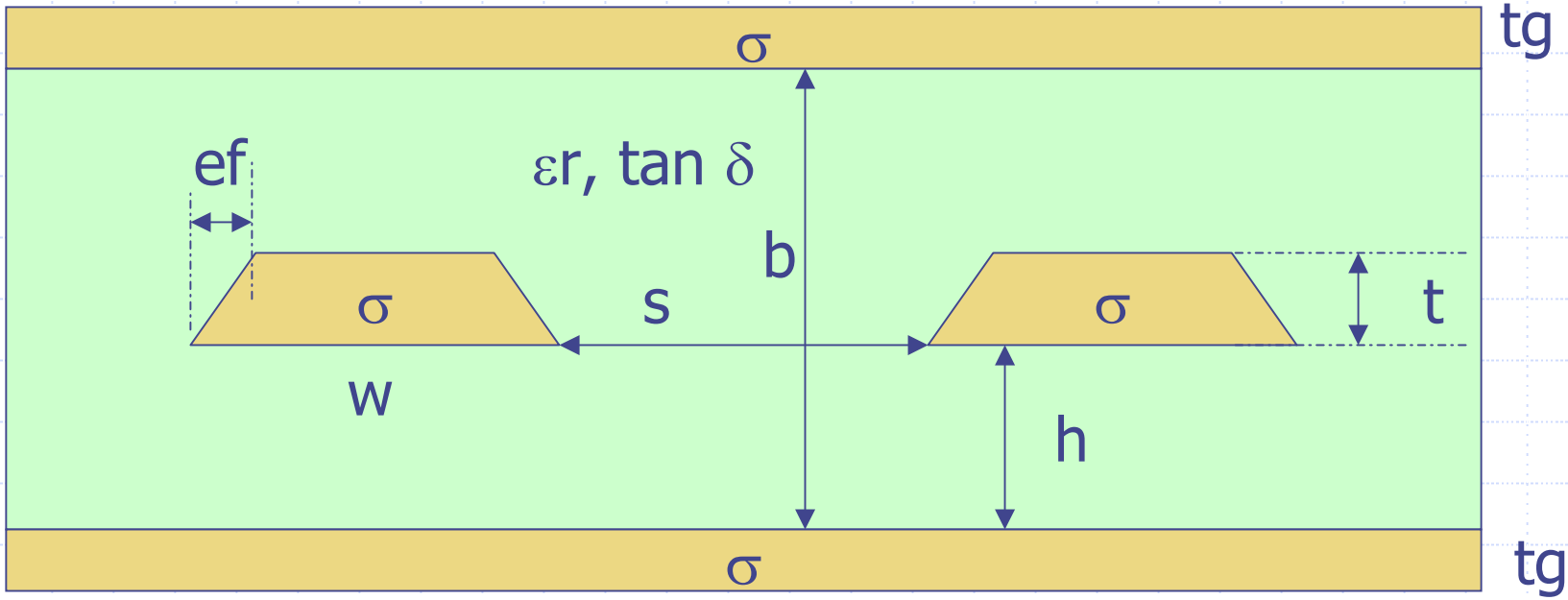


- The general syntax support any number of input and equal number of output port.
- This length in this example is 0.1
- The units are the often assumed to be meter but actually are the per length units of the RLCG model.
  - The internal field solver produces units in meters

Creating a field solution

- Create a file that invokes the target transmission lines.
- In this file also specify:
  - Field solver options
  - Materials
  - Stackup
    - The dielectric and power/ground conductor plane sandwich of a PWB
  - Trace geometries shapes
  - The a models that include the above

Couple Strip Line Example (twolines.sp)




```
.Title Field Solver
W2
+ 1 2  0  a b 0
+ Fsmodel=s5_z068.9_z0d108.8   N=2 L=1 DELAYOPT=1
*                          s          w          ef         t          Tg
* mils                     5          5          0.5        0.5        1
* mils converted to meters 1.270E-04  1.27E-04   1.27E-05   1.27E-05   2.54E-03
*                          b          h          er   tand  u          conduct.
* mils                     0.5        10         3.90  .02  1          4.2E+07
* mils converted to meter  5.207E-04  2.54E-04
```

Using the Solver

First line should be comment or title else it gets ignored

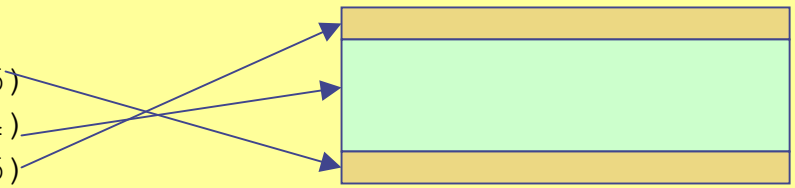
Invoking the W element will cause the field solver to run, if the FSMODEL parameter is specified

```
.Title Field Solver
W2
+ 1 2  0  a b 0
+ Fsmodel=s5_z068.9_z0d108.8   N=2 L=1 DELAYOPT=1
*                              s          w          ef         t          Tg
* mils                          5          5          0.5        0.5        1
* mils co      d to meters     1.270E-04  1.27E-04   1.27E-05   1.27E-05   2.54E-03
*                              b          h          er   tand  u          conduct.
* mils                         0.5        10         3.90  .02  1          4.2E+07
* mils converte                5.207E-04  2.54E-04

(cont'd on next pa
```

Often this is done outside the main net list to insure solution quality. Then a RLGCMODEL or RLGCFILE statement would be used here instead

Field Solver Option Statement

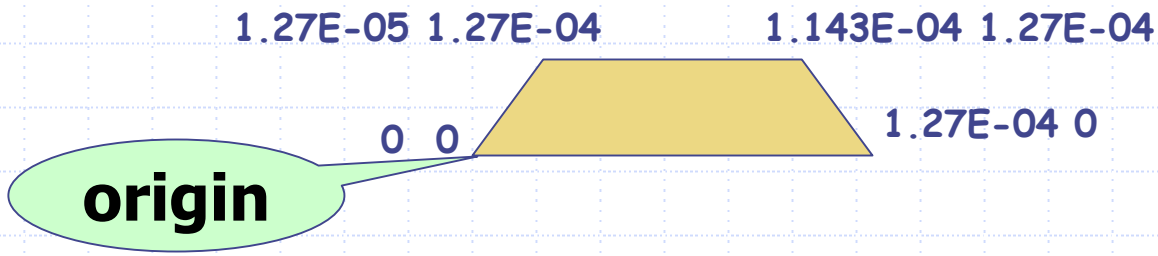
```
.FSoptions  brd_opt2      ACCURACY = HIGH GRIDFACTOR = 1
+ ComputeRo=yes ComputeRs=yes ComputeGo=yes ComputeGd=yes PRINTDATA=yes
*
.MATERIAL   brd_dielct2   DIELECTRIC   ER=3.9, LOSSTANGENT=0.019
.MATERIAL   brd_cu2       METAL        CONDUCTIVITY=42000000
*
.LAYERSTACK   brd_ssl_stk2
+LAYER=(  brd_cu2       ,1.2700E-05)
+LAYER=(  brd_dielct2   ,5.2070E-04)
+LAYER=(  brd_cu2       ,1.2700E-05)
```



- This statement is good start to set up options.
- In this case the options are called brd_opt2
- The board is material "brd_dielect2"
- The conductor material is "brd_cu2"
- Notice that the traces are not specified here.
- The stackup actually starts at the bottom and works up. Each layer thickness is specified.

Material and Shapes

```
.MATERIAL   brd_dielct2   DIELECTRIC   ER=3.9, LOSSTANGENT=0.019
.MATERIAL   brd_cu2       METAL        CONDUCTIVITY=42000000

.SHAPE  brd_trap1 POLYGON   VERTEX =
+( 0   0   1.2700E-05 1.2700E-04 1.1430E-04 1.2700E-04 1.2700E-04 0   )
```



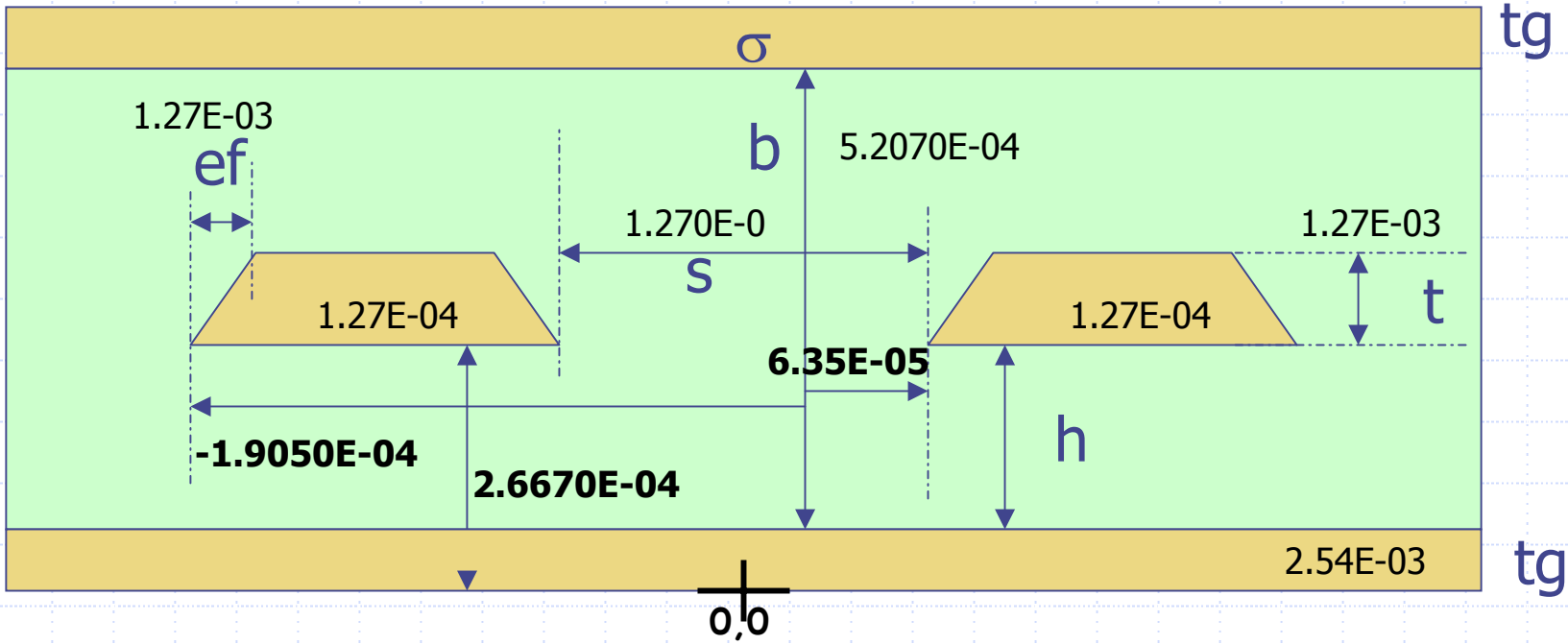
- Next specify the properties of the dielectrics and metal
- Then specify the shape.
- A first pass guess often uses rectangle for trace.
- In this example we use a trapezoid

The Model statement

```
.MODEL  s5_z068.9_z0d108.8
+W  MODELTYPE=FieldSolver, LAYERSTACK=brd_ssl_stk2  FSoptions=brd_opt2
+CONDUCTOR=(SHAPE=brd_trap2 MATERIAL=brd_cu2
+   ORIGIN=(  6.3500E-05, 2.6670E-04)
+CONDUCTOR=(SHAPE=brd_trap2 MATERIAL=brd_cu2
+    ORIGIN=( -1.9050E-04, 2.6670E-04)
+RLGCfile=s5_z068.9_z0d108.8.rlc
.END
```

- Here the field solver calls out what was specified.
  - brd_ssl_stk2
  - brd_opt2
  - brd_cu2
  - brd_trap2

Placing the Shapes



- The origin for the solution is at the bottom of the stackup
- The positioning of the trapezoids with the stackup are in relation to the shape origin

The RLCG Model

```
.MODEL s5_z068.9_z0d108.8 W MODELTYPE=RLGC, N=2
+ Lo = 4.460644e-007
+      9.544025e-008 4.460644e-007
+ Co = 1.019475e-010
+      -2.181277e-011 1.019475e-010
+ Ro = 1.637366e+001
+      0.000000e+000 1.637366e+001
+ Go = 0.000000e+000
+      0.000000e+000 0.000000e+000
+ Rs = 2.056598e-003
+      9.268906e-005 2.056651e-003
+ Gd = 1.217055e-011
+      -2.604020e-012 1.217055e-011
.ENDS
```

$$\begin{pmatrix} 1.019475\times 10^{-10} & \\ -2.181277\times 10^{-11} & 1.019475\times 10^{-10} \end{pmatrix}$$

- Only half of the diagonal and the lower half of the matrix is specified
- Default units are H/m, F/m, Ω/m, S/m, Ω/(m*srqt(Hz), S/(m*Hz) respectively
- Alternatively H/in, F/in, Ω/in, S/in, Ω/(in*srqt(Hz), S/(in*Hz) can be used if L units are to be specified in inches.
- A more detailed description can be found in the HSPICE transmission line chapter

Tline issues for SI engineers

- Validation of transmission line models
- Comparison to equations.
  - Most equation are only accurate to a few ohms and have are limited to only certain ratios of trace geometry
- Differential impedance equations are not readily available.
- Tools to compare to measurement
  - Vector Network Analyzer
  - Time Domain Refectometry

Receiver

```
.LIB        'receiver'
.SUBCKT     RCV   in     Vss
Rin    in   Vss   45
Cin    in   Vss   0.5pf
.ENDS
.ENDL
```

- This too could be more complicated transistor or IBIS circuit.
- In this case we start with 45 ohms to ground with a 0.5 pF shunt across the load.

The main net list – Top Half

testckt.sp *

```
.TITLE  Signal integrity Training deck
* top level net list
.LIB 'testckt.sp'       'pulse'
.LIB 'testckt.sp'       'driver'
.LIB 'testckt.sp'       'LC_pack'
.LIB 'testckt.sp'       'easy_lines'
.LIB 'testckt.sp'       'receiver'
.PARAM          UI=10ns          VOL=0.25v VOH=1.2v    $  parameters for data stream
.TRAN           10p 50n                                  $ set time step and end time
```

- The libraries will go at the end for this example
  - In fact all of the above statements are position independent although parameter usage is position sensitive. Be careful if parameter are set in libraries. This can effect the order of parameter processing.
- The libraries are normally in the another file. This example is not standard practice but it is convenient for collaborating on issues.
- The global parameter for the bit interval UI is set to 10 nanoseconds.
- Two more global parameters are used for buffer voltage control, Vol and Voh.
- The transient statement tells Hspice to start a transient analysis when the ".end" statement is processed. In this case the time step interval is 10ps and will stop at 20 ns.

Helpful hints to resolve time step errors

- Voltage transitions that are too fast
  - Consider slower transition time
- Un-initialized reactive components can case instantaneous spikes that create very fast transitions before setting.
  - Consider setting "IC" (initial condition.)
- Capacitors and inductors that are too small
  - Consider eliminating or combining
  - Consider putting shunt resistor across device
- Floating references or nodes can cause time step errors.
  - DC path can't be determined if switches or controlled sources are used and may be considered floating at time t=0
  - Provide high resistance shunts to node 0
- Transmission lines that are too short.
  - Consider replacing with LC
- Switches can cause spikes.
  - Use voltage controlled resistor to soften open and close resistance as function of time. (more on this later)
- Small mutual "k" elements.
  - Consider elimiating same k elements.

Main Net List Flows Like a Circuit

testckt.sp *

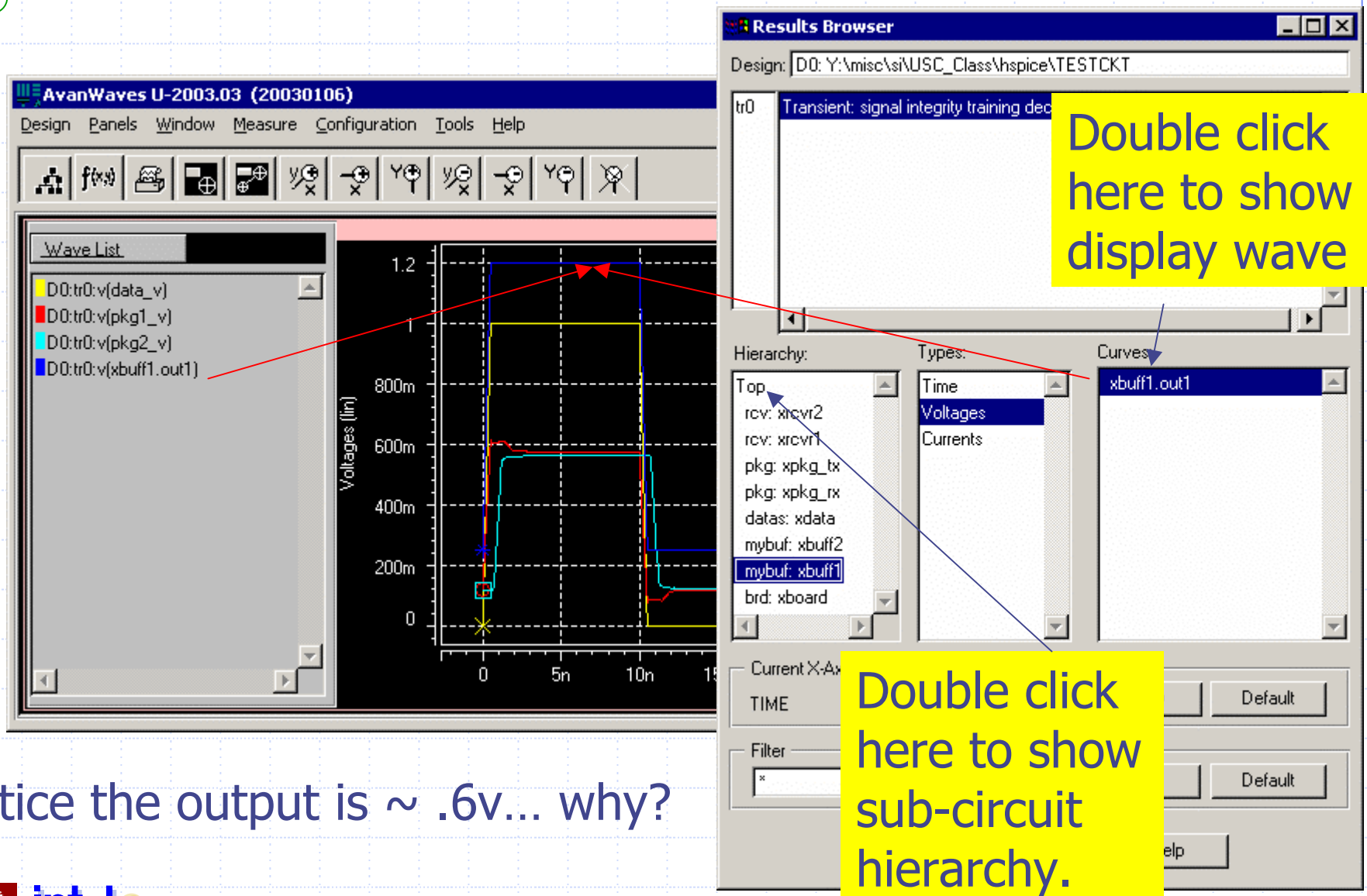
```
Xdata           data_v  data_a       DATAS              $ Data Stream for victim and aggressor
+ Datarate=UI                                           $ pass data rate parameter

XBUFF1          data_v  pkg1_v          0    MYBUF      $ victim buffer
XBUFF2          data_a  pkg1_a          0    MYBUF      $ aggressor buffer

XPKG_Tx         pkg1_v  pkg1_a
+               pkg1_vo pkg1_ao         0      PKG      $ Tx package

Xboard          pkg1_vo pkg1_ao
+               pkg2_vo pkg2_ao         0      BRD      $ printed wiring board subcircuit

XPKG_Rx         pkg2_vo pkg2_ao
+               pkg2_v  pkg2_a          0      PKG      $ Rx package

XRCVR1          pkg2_v          0               RCV      $ victim receiver
XRCVR2          pkg2_a          0               RCV      $ aggressor receiver

.MEAS TRAN flight_time_v TRIG V(data_v) VAL=.5V RISE=1  TD=0ns TARG V(pkg2_v) VAL=.5 RISE=1 TD=0ns
.MEAS TRAN flight_time_a TRIG V(data_a) VAL=.5V RISE=1  TD=0ns TARG V(pkg2_a) VAL=.5 RISE=1 TD=0ns
.END
```

- "$" is a comment after column 1
- "*" in column 1 comments that line
- If an .option probe control statement is used on the nodes data_v and pkg2_v will be stored in the tr0 file.

Assignment

- Take the previous HSPICE example and draw a circuit schematic.
- Produce the last picture in AvanWaves (if available)
- Look up and read all chapters in the HSPICE manual on:
  - Subcircuits
  - Libraries
  - E source
  - Coupled inductor
  - W-elements
  - Notice this part to the assignment is looser that most academic reading assignments. In business data-mining is a required skill. Also look up any element we cover that you do not understand.

A look at results in AvanWaves



Notice the output is ~ .6v… why?

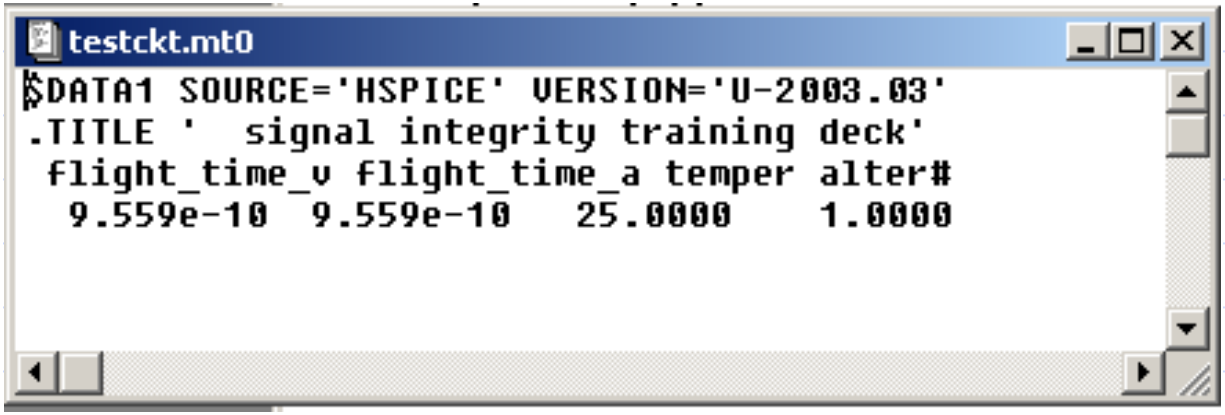
Measurement

testckt.sp *

```
.MEAS TRAN flight_time_v TRIG V(data_v) VAL=.5V RISE=1  TD=0ns TARG V(pkg2_v) VAL=.5 RISE=1 TD=0ns
.MEAS TRAN flight_time_a TRIG V(data_a) VAL=.5V RISE=1  TD=0ns TARG V(pkg2_a) VAL=.5 RISE=1 TD=0ns
```

- There is a manual contain an extensive list of measurements that can be made.
- In this case we are making a measurement called "flight_time_v" and "flight_time_a"
- The trigger for the beginning of the measurement is at 0.5 V on the first rising on node data_v (and data_a.)
- The completion of the measurement is when the first rising edge on node pkg2_v (and pkg2_a) reaches 0.5 V.
- TD parameter means time delay before the measurement starts and is 0s in this example.

MT0 file



- This resultant MT0 file
- The second line is the title
- The third line and all the lines that follow up to the "alter#" parameters are the parameters names.
- The following lines are the corresponding measurement values
  - For this case the measurement for the parameters "flight_time_v" and "flight_time_a" are the 955.9 ps.

Monte Carlo Analysis

```
.TITLE  Signal integrity Training deck
* parameter variations
.param rx_rterm1=AGAUSS(50, 10, 3)
rx_cterm1=AGAUSS(1pf,.8pf, 3)
.param rx_rterm=rx_rterm1 rx_cterm=rx_cterm1
.param tx_rterm1=GAUSS(45, 0.1, 3)
tx_cterm1=GAUSS(.5pf,0.1, 3)
.param tx_rterm=tx_rterm1 tx_cterm=tx_cterm1
.param pkg_coulping1=GAUSS(.2,.5, 3)
.param pkg_coulping=pkg_coulping1
.param tr1=AGAUSS(.5ns,.45ns, 3)
.param tr=tr1
```

- Notice we use a dummy variable (suffixed with 1).
- This is because every time a for example Rx_term1 is used it will get a new value. By assigning it a dummy variable at the beginning of a sweep the value will be set for that entire sweep. Else each time the variable is used a new value will be assigned.

Invoking a Monte Carlo Sweep

testckt_monte.sp *

```
.TITLE  Signal integrity Training deck
* parameter variations
.param rx_rterm1=AGAUSS(50, 15, 3) rx_cterm1=AGAUSS(1pf,.8pf, 3)
.param rx_rterm=rx_rterm1 rx_cterm=rx_cterm1
.param tx_rterm1=GAUSS(45, 0.25, 3) tx_cterm1=GAUSS(.5pf,0.40, 3)
.param tx_rterm=tx_rterm1 tx_cterm=tx_cterm1
.param pkg_coulping1=GAUSS(.2,.5, 3)
.param pkg_coulping=pkg_coulping1
.param tr1=AGAUSS(.5ns,.45ns, 3)
.param tr=tr1

* top level net list
.LIB 'testckt_monte.sp'      'pulse'
.LIB 'testckt_monte.sp'      'driver'
.LIB 'testckt_monte.sp'      'LC_pack'
.LIB 'testckt_monte.sp'      'easy_lines'
.LIB 'testckt_monte.sp'      'receiver'

.PARAM         UI=10ns          VOL=0.25v VOH=1.2v    $  parameters for data stream
.TRAN          10p 50n SWEEP MONTE=5000               $ set time step and end time
.OPTION PROBE
```

- The .TRAN statement is new syntax added to it "SWEEP MONTE=5000"
  - This will cause 5000 sweeps to be created in the tr0 file.
- Option probe statement was added so that only node annotated with the .probe statement will be stored since 5000 sweeps will create a very large tr0 file.

A few changes added to the end

testckt_monte.sp *

```
+                pkg2_v   pkg2_a              0      PKG     $ Rx package

XRCVR1           pkg2_v              0               RCV     $ victim receiver
XRCVR2           pkg2_a              0               RCV     $ aggressor receiver


.PROBE v(data_v) V(pkg2_v)
.MEAS TRAN flight_time_v TRIG V(data_v) VAL=.25V RISE=1  TD=0ns
+                        TARG V(pkg2_v) VAL=.25  RISE=1  TD=0ns

.END
```

- The ".PROBE" statement is used in conjunction with the .OPTION PROBE statement so only node data_v and pkg2_v are reported.
- Only one measurement is used and the threshold was lowered to 250 mv

The "Sweep" MTO file

testckt_monte.mt0

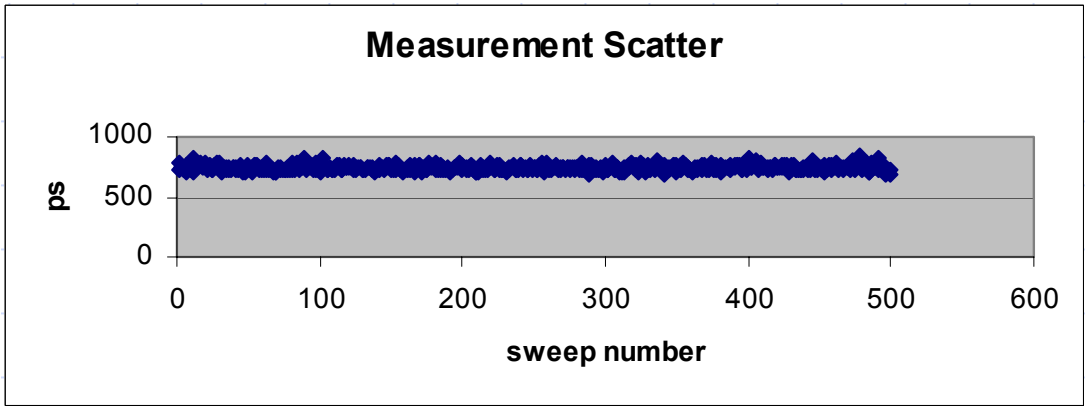
```
$DATA1 SOURCE='HSPICE' VERSION='U-2003.03'
.TITLE '  signal integrity training deck'
 index rx_rterm@rx_rter rx_cterm@rx_cter tx_rterm@tx_rter tx_cterm@tx_cter
 pkg_coulping@pkg tr@tr1 flight_time_v temper alter#
    1.0000   52.2931  7.175e-13   44.4769  5.104e-13    0.1647  6.183e-10
  7.249e-10   25.0000    1.0000
    2.0000   43.9272  1.297e-12   49.1339  4.297e-13    0.2426  5.371e-10
  7.845e-10   25.0000    1.0000
    3.0000   53.0351  1.277e-12   45.6194  5.103e-13    0.1689  5.959e-10
  7.439e-10   25.0000    1.0000
    4.0000   51.4748  1.126e-12   43.7288  3.418e-13    0.1991  6.902e-10
  7.396e-10   25.0000    1.0000
    5.0000   52.7243  6.996e-13   43.6163  4.501e-13    0.2459  5.369e-10
  7.206e-10   25.0000    1.0000
    6.0000   40.2705  7.143e-13   45.1774  5.328e-13    0.1918  5.413e-10
  7.730e-10   25.0000    1.0000
    7.0000   54.0315  9.344e-13   36.7611  4.847e-13    0.2682  4.747e-10
  7.086e-10   25.0000    1.0000
    8.0000   48.6630  1.286e-12   43.7602  4.424e-13    0.1639  3.082e-10
  7.423e-10   25.0000    1.0000
    9.0000   48.8762  1.402e-12   43.4610  4.772e-13    0.1713  5.757e-10
```

- Note each sweep entry contains the values that were assigned to the Monte Carlo parameters
- A VBA or perl script is normally used (and required) to convert into a spreadsheet format

Viewing Monte Carlo in a Spreadsheet

- Step 1: create spreadsheet with result column
- Step 2: create cells with the min, max, mean (average), and standard deviation of the results
- Step 4: On a new sheet create a column that contains a number of equally spaced bins which at least bound the maximum and minimum readings.
- Step 5: Select the cells adjacent to the bins.
- Step 6: Got to the main menu and insert function and select "FREQUENY" from the statistics section. A window will pop up.
- Step 7: Enter the result data cell range point and the bin cell range points respectively but "DO NOT HIT RETURN or ENTER"!
  - FREQUENCY(B2:B6000,F8:F36)
- Step 7: Press CTL-SHIFT-ENTER. This is the range entry terminator. The frequency of each bin will appear next to each bin cell.
- Step 8: Create a column next to the frequency column that is each frequency column entry divided by the sum of all bins. This is the probability that a result will be in that bin.
- Step 9: Create a column next to the bin probability that used the 'NORMDIST' function.
  - NORMDIST(F8,MEAN,SIGMA,FALSE)
- Step 10: Create a column next to the normdist column that is normalized.
  - K8/SUM(K:K)
- Step 11: Select the normalized distribution and bin probability column and choose chart from the insert menu. Select the "custom types" tap and the "line-column" type. Use the bin name as x labels.

Check Scatter Plot First

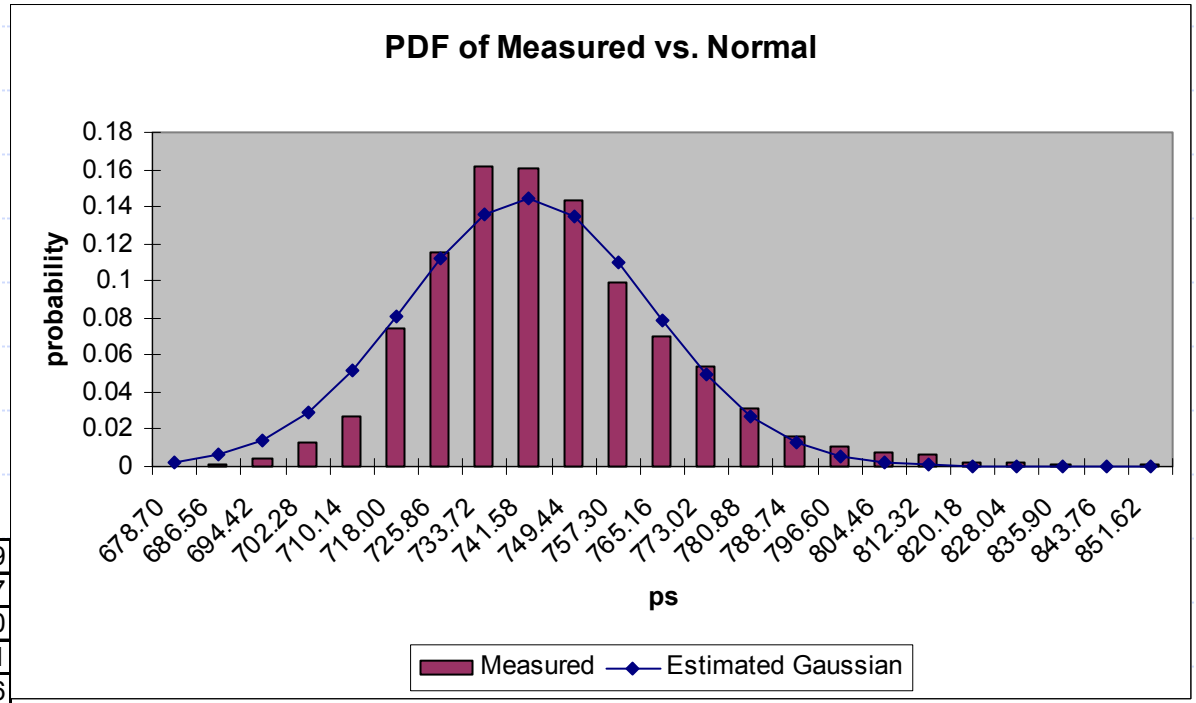


Threshold = 0.35 V

- The above scatter plot suggests that the measurements are reasonable well distributed.

Results of Monte Carlo Analysis

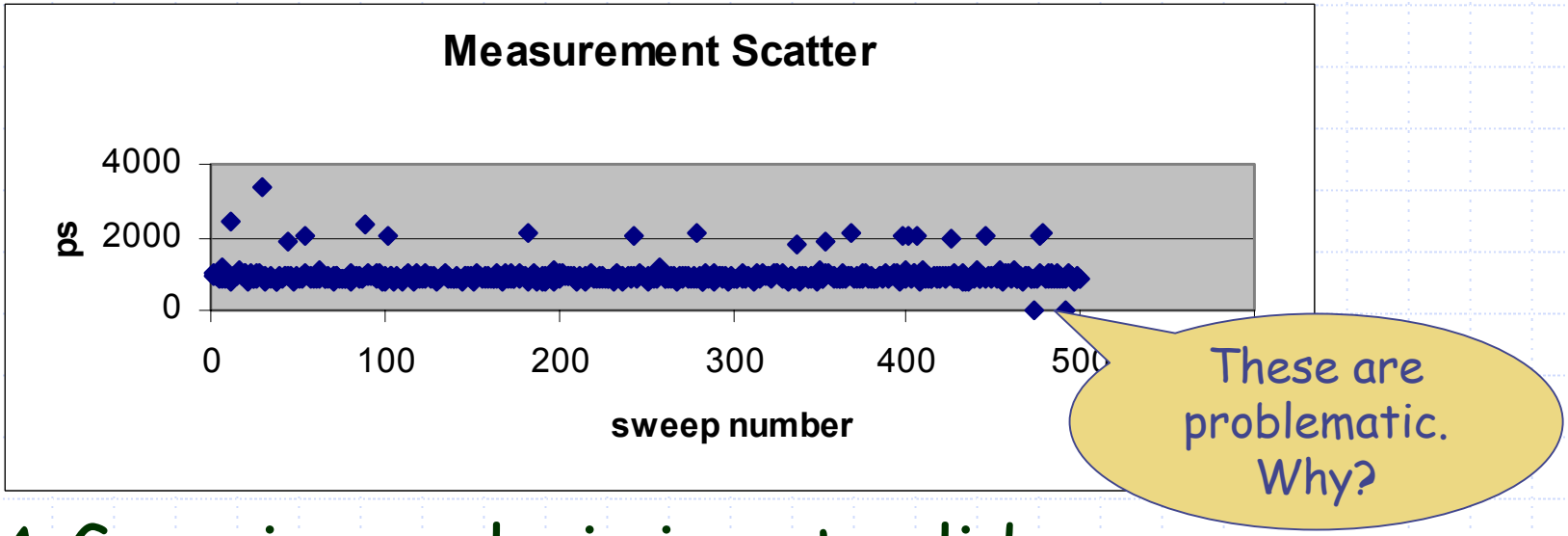
- Break For spreadsheet walk through
- Results below



Measured Data		Top	835.9
	724.9	bottom	678.7
	784.5	bins	20
	743.9	SIGMA	21.71
	739.6	MEAN	741.31546
	720.6		
	773		
	708.6		
	742.3		
	755.7		

Bin Number	Bin Values	Frequency	copy of bin values	Measured PDF	Normalized Gaussian	Gaussian Curve for bin value
1	678.70	1	678.70	0.000205	0.002	0.00028687
2	686.56	4	686.56	0.000821	0.006	0.00076344
3	694.42	22	694.42	0.004516	0.014	0.001782097

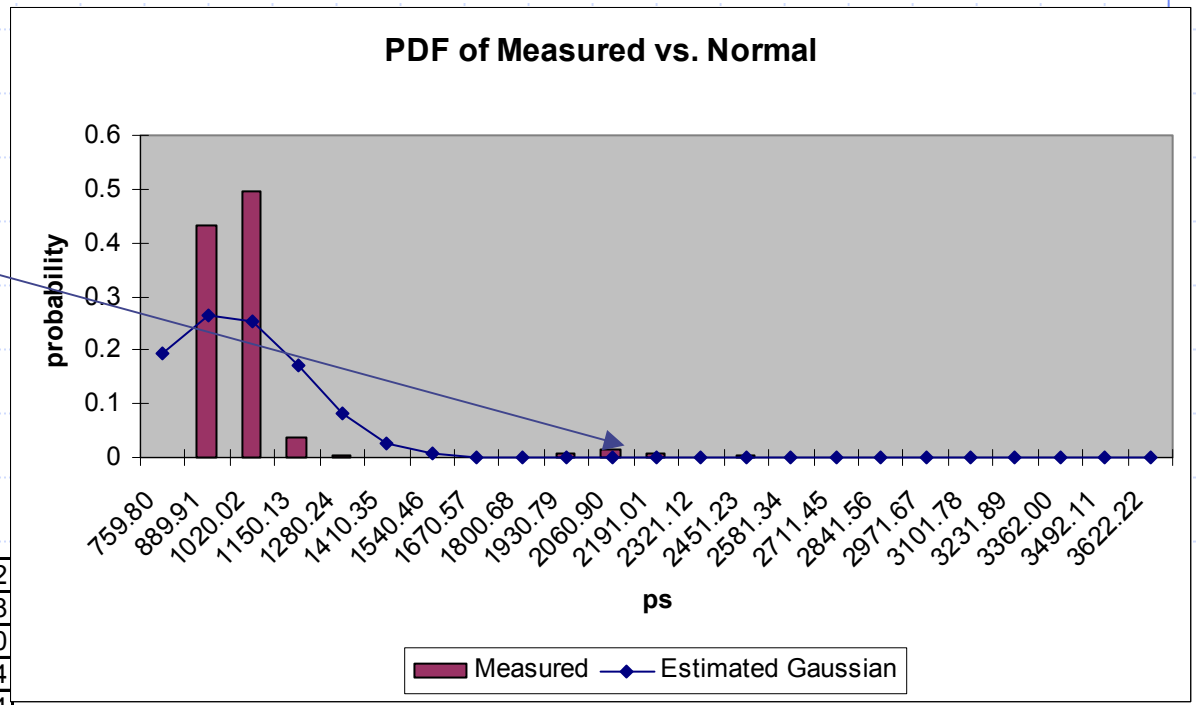
What happens if the scatter plot has outliers



- A Gaussian analysis is not valid
- Outliers suggest that there exists physical anomalies that must be determined.
- The next step is to look a the waveforms.
- The following page will illustrate the issues with using a Gaussian fit for the above data.

Distribution results for pervious bad scatter

Looks like a "mode" or likely occurrence here



Measured Data		Top	3362
	910.6	bottom	759.8
	1021	bins	20
	923.1	SIGMA	218.64
	947.5	MEAN	941.1475904
	878.4	4.5 π high	1925.005679
	1209	top π	11.07256826
	831.9	3 π high	1597.052983
	848.2	3 π low	285.2421982
	940.7	bottom π	0.829453116
	882.1	4.5 π low	-42.71049789
	799.2		

Bin Number	Bin Values	Frequency	copy of bin values	Measured PDF	Normalized Gaussian	Gaussian Curve for bin value
1	759.80	1	759.80	0.001004	0.193	0.001293583
2	889.91	429	889.91	0.430723	0.264	0.001775269
3	1020.02	493	1020.02	0.49498	0.255	0.001709742
4	1150.13	38	1150.13	0.038153	0.172	0.001155563
5	1280.24	2	1280.24	0.002008	0.082	0.000548092

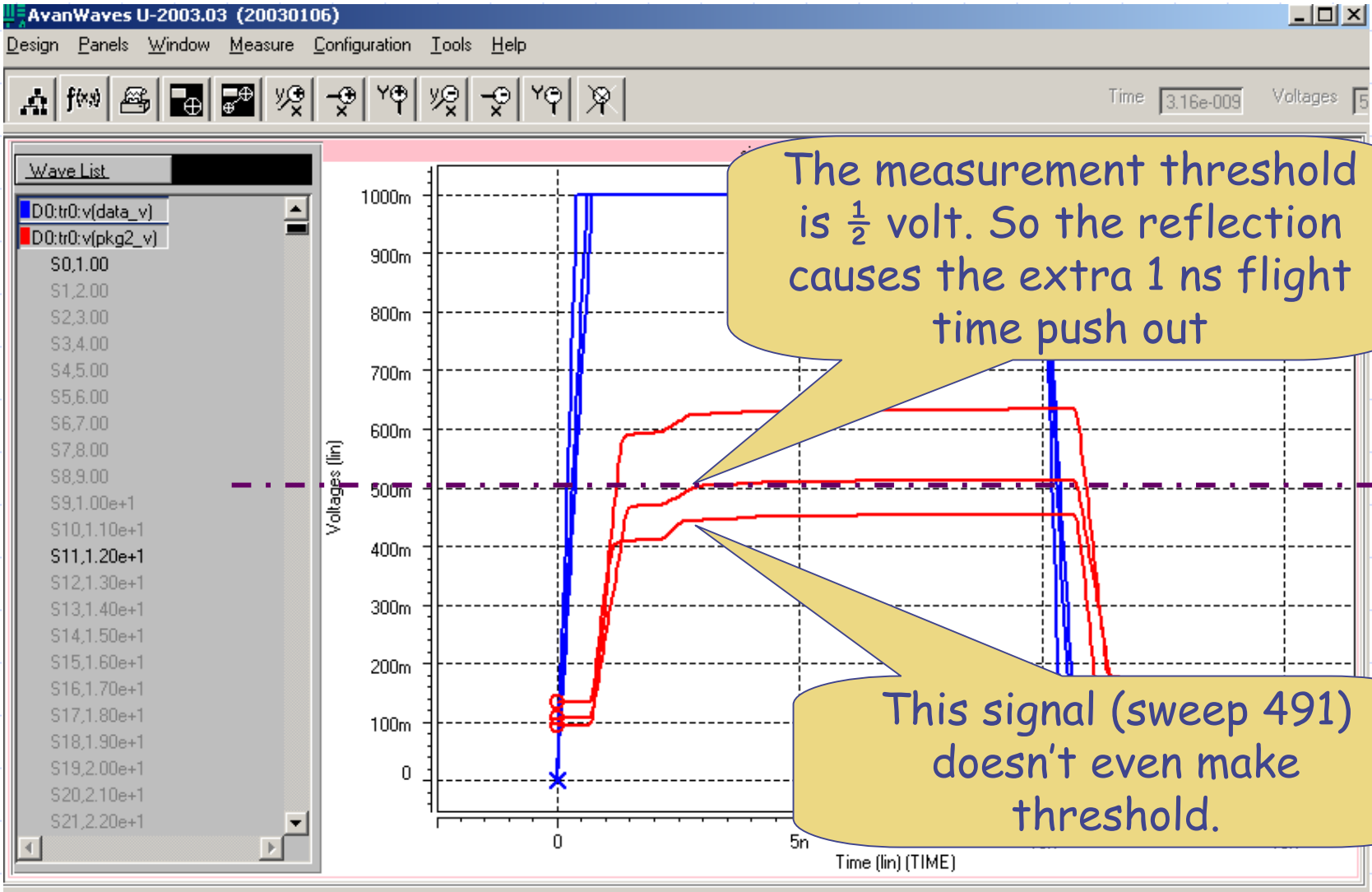
Identify a sweep with the anomaly

index	rx_rterm@	rx_cterm@	tx_rterm@1	tx_cterm@	pkg_coulpi	tr@tr1	flight_time_	temper	alter#
1	52.2931	7.18E-13	44.4769	5.10E-13	0.1647	6.18E-10	9.11E-10	25	1
2	43.9272	1.30E-12	49.1339	4.30E-13	0.2426	5.37E-10	1.02E-09	25	1
3	53.0351	1.28E-12	45.6194	5.10E-13	0.1689	5.96E-10	9.23E-10	25	1
4	51.4748	1.13E-12	43.7288	3.42E-13	0.1991	6.90E-10	9.48E-10	25	1
5	52.7243	7.00E-13	43.6163	4.50E-13	0.2459	5.37E-10	8.78E-10	25	1
6	40.2705	7.14E-13	45.1774	5.33E-13	0.1918	5.41E-10	1.21E-09	25	1
7	54.0315	9.34E-13	36.7611	4.85E-13	0.2682	4.75E-10	8.32E-10	25	1
8	48.663	1.29E-12	43.7602	4.42E-13	0.1639	3.08E-10	8.48E-10	25	1
9	48.8762	1.49E-12	43.4619	4.48E-13	0.1713	5.76E-10	9.41E-10	25	1
10	48.9853	8.33E-13	42.243	5.27E-13	0.2528	4.82E-10	8.82E-10	25	1
11	58.1424	9.49E-13	41.5538	5.02E-13	0.2519	3.18E-10	7.99E-10	25	1
12	**38.3262**	**6.58E-13**	**49.1414**	**3.78E-13**	**0.163**	**7.29E-10**	**2.45E-09**	**25**	**1**
13	53.1818	8.41E-13	52.6199	5.03E-13	0.2237	5.20E-10	9.17E-10	25	1
14	47.8564	9.43E-13	41.956	5.72E-13	0.197	6.80E-10	9.58E-10	25	1
15	48.0179	1.20E-12	48.6048	5.71E-13	0.2064	4.72E-10	9.29E-10	25	1
16	48.1116	5.70E-13	47.2849	5.20E-13	0.1799	8.24E-10	1.04E-09	25	1

- Look at the 11th sweep
  - Sweeps start with 0

Break for using statistics in Excel demo

Sweep, sweep 11, and sweep 491



Resolving problems

- There are actually 3 mode for the previous case.
  - Normal case
  - Measurement on reflection part of signal
  - Signal is below the threshold.
- There first two can be delt with by increasing margin.
- The third suggest a design change.
- Assignment: What Rx range will guarantee the only the normal case assuming the ½ volt threshold.
  - You need to get the basic data in the from the Monte netlist.

Entering Flight Time Into Budget

- If the distribution looks Gaussian then most designs will use the 3 sigma numbers.
- A more conservative approach would be to use 4 sigma number.
- If the result are realistically bounded, but not Gaussian, the extreme limits can be used but there is a risk that the worst combination was not simulated.

Backup - Hspice Listings

testckt.sp main program

testckt.sp

```
.TITLE  Signal integrity Training deck
* top level net list
.LIB 'testckt.sp'      'pulse'
.LIB 'testckt.sp'      'driver'
.LIB 'testckt.sp'      'LC_pack'
.LIB 'testckt.sp'      'easy_lines'
.LIB 'testckt.sp'      'receiver'
.PARAM         UI=10ns          VOL=0.25v VOH=1.2v   $  parameters for data stream
.TRAN          10p 50n                                 $ set time step and end time

Xdata          data_v  data_a       DATAS              $ Data Stream for victim and aggressor
+ Datarate=UI                                          $ pass data rate parameter

XBUFF1         data_v  pkg1_v             0   MYBUF    $ victim buffer
XBUFF2         data_a  pkg1_a             0   MYBUF    $ aggressor buffer

XPKG_Tx        pkg1_v  pkg1_a
+              pkg1_vo pkg1_ao            0     PKG    $ Tx package

Xboard         pkg1_vo pkg1_ao
+              pkg2_vo pkg2_ao            0     BRD    $ printed wiring board subcircuit

XPKG_Rx        pkg2_vo pkg2_ao
+              pkg2_v  pkg2_a             0     PKG    $ Rx package

XRCVR1         pkg2_v            0               RCV    $ victim receiver
XRCVR2         pkg2_a            0               RCV    $ aggressor receiver

.MEAS TRAN flight_time_v TRIG V(data_v) VAL=.5V RISE=1  TD=0ns TARG V(pkg2_v) VAL=.5 RISE=1 TD=0ns
.MEAS TRAN flight_time_a TRIG V(data_a) VAL=.5V RISE=1  TD=0ns TARG V(pkg2_a) VAL=.5 RISE=1 TD=0ns


.END
```

* Review in printed form

testckt.sp – libraries (cont'd)

testckt.sp

```
.LIB    'easy_lines'
.SUBCKT                 BRD       in1           in2 out1   out2     Vss
Vline1                  in1       in2           Vss out1  out2     Vss
+                       RLGCMODEL= 's5_z068.9_z0d108.8'    N=2 L=.1
*SYSTEM_NAME : s5_z068.9_z0d108.8
*
*  ------------------------------------ Z = 5.461000e-004
*  //// Top Ground Plane ///////////////
*  ------------------------------------ Z = 5.334000e-004
*       brd_dielct2   H = 5.207000e-004
*  ------------------------------------ Z = 1.270000e-005
*  //// Bottom Ground Plane ////////////
*  ------------------------------------ Z = 0

* L(H/m), C(F/m), Ro(Ohm/m), Go(S/m), Rs(Ohm/(m*sqrt(Hz)), Gd(S/(m*Hz))

.MODEL s5_z068.9_z0d108.8 W MODELTYPE=RLGC, N=2
+ Lo = 4.460644e-007
+      9.544025e-008 4.460644e-007
+ Co = 1.019475e-010
+      -2.181277e-011 1.019475e-010
+ Ro = 1.637366e+001
+      0.000000e+000 1.637366e+001
+ Go = 0.000000e+000
+      0.000000e+000 0.000000e+000
+ Rs = 2.056598e-003
+      9.268906e-005 2.056651e-003
+ Gd = 1.217055e-011
+      -2.604020e-012 1.217055e-011
.ENDS
.ENDL

.LIB    'receiver'
.SUBCKT         RCV     in        Vss
Rin             in      Vss               45
Cin             in      Vss       0.5pf
.ENDS
.ENDL
```

testckt.sp - libraries

testckt.sp

```
.LIB                    'pulse'
.SUBCKT         DATAS   bit1      bit2     datarate=-1
V1              bit1              0           PULSE    0v      1v      0n      .5n  .5n  'datarate-.5n' '2*datarate'
V2              bit2              0           PULSE    0v      1v      0n      .5n  .5n  'datarate-.5n' '2*datarate'
.ENDS
.ENDL

.LIB                    'driver'
.SUBCKT         MYBUF   in        out           Vss
Edrive          out1    Vss     VOL='(Voh-Vol)*V(in)+Vol'
Rout            out1    out               50
Cout            out1    Vss       1pF
Rin in vss 1G
.ENDS
.ENDL

.LIB      'LC_pack'
.SUBCKT                 PKG       in1           in2 out1   out2     Vss
Lin1            in1     out1      1n
Lin2            in2     out2      1n
K1              Lin1    Lin2            0.2
.ENDS
.ENDL
```

testckt_monte.sp main program

testckt_monte.sp *

```
.param tr=tr1

* top level net list
.LIB 'testckt_monte.sp'      'pulse'
.LIB 'testckt_monte.sp'      'driver'
.LIB 'testckt_monte.sp'      'LC_pack'
.LIB 'testckt_monte.sp'      'easy_lines'
.LIB 'testckt_monte.sp'      'receiver'

.PARAM         UI=10ns         VOL=0.25v VOH=1.2v   $  parameters for data stream


.TRAN          10p 50n SWEEP MONTE=1000           $ set time step and end time
.OPTION PROBE

Xdata          data_v  data_a       DATAS             $ Data Stream for victim and aggressor
+ Datarate=UI                                         $ pass data rate parameter

XBUFF1         data_v  pkg1_v            0   MYBUF    $ victim buffer
XBUFF2         data_a  pkg1_a            0   MYBUF    $ aggressor buffer

XPKG_Tx        pkg1_v  pkg1_a
+              pkg1_vo pkg1_ao           0     PKG    $ Tx package

Xboard         pkg1_vo pkg1_ao
+              pkg2_vo pkg2_ao           0     BRD    $ printed wiring board subcircuit

XPKG_Rx        pkg2_vo pkg2_ao
+              pkg2_v  pkg2_a            0     PKG    $ Rx package

XRCVR1         pkg2_v           0              RCV    $ victim receiver
XRCVR2         pkg2_a           0              RCV    $ aggressor receiver


.PROBE v(data_v) V(pkg2_v)
.MEAS TRAN flight_time_v TRIG V(data_v) VAL=.5V RISE=1
+TD=0ns TARG V(pkg2_v) VAL=.5 RISE=1 TD=0ns

.END
```

testckt_monte.sp - libraries

testckt_monte.sp *

```
.LIB                    'pulse'
.SUBCKT         DATAS   bit1      bit2      datarate=-1
V1              bit1              0            PULSE     0v       1v       0n       tr   tr   'datarate-tr' '2*datarate'
V2              bit2              0            PULSE     0v       1v       0n       tr   tr   'datarate-tr' '2*datarate'
.ENDS
.ENDL

.LIB                    'driver'
.SUBCKT         MYBUF   in        out             Vss
Edrive          out1    Vss     VOL='(Voh-Vol)*V(in)+Vol'
Rout            out1    out       Tx_rterm
Cout            out1    Vss       Tx_cterm
Rin in vss 1G
.ENDS
.ENDL

.LIB       'LC_pack'
.SUBCKT                 PKG       in1           in2 out1   out2     Vss
Lin1            in1     out1      1n
Lin2            in2     out2      1n
K1              Lin1    Lin2           .2
.ENDS
.ENDL
```

testckt_monte.sp - libraries (cont'd)

testckt_monte.sp *

```
.LIB    'easy_lines'
.SUBCKT                 BRD       in1           in2 out1   out2     Vss
Wline1                  in1       in2           Vss out1  out2     Vss
+                       RLGCMODEL= 's5_z068.9_z0d108.8'    N=2 L=.1
*SYSTEM_NAME : s5_z068.9_z0d108.8
*
*  ------------------------------------ Z = 5.461000e-004
*  //// Top Ground Plane ///////////////
*  ------------------------------------ Z = 5.334000e-004
*       brd_dielct2   H = 5.207000e-004
*  ------------------------------------ Z = 1.270000e-005
*  //// Bottom Ground Plane ////////////
*  ------------------------------------ Z = 0

* L(H/m), C(F/m), Ro(Ohm/m), Go(S/m), Rs(Ohm/(m*sqrt(Hz)), Gd(S/(m*Hz))

.MODEL s5_z068.9_z0d108.8 W MODELTYPE=RLGC, N=2
+ Lo = 4.460644e-007
+      9.544025e-008 4.460644e-007
+ Co = 1.019475e-010
+      -2.181277e-011 1.019475e-010
+ Ro = 1.637366e+001
+      0.000000e+000 1.637366e+001
+ Go = 0.000000e+000
+      0.000000e+000 0.000000e+000
+ Rs = 2.056598e-003
+      9.268906e-005 2.056651e-003
+ Gd = 1.217055e-011
+      -2.604020e-012 1.217055e-011
.ENDS
.ENDL

.LIB    'receiver'
.SUBCKT         RCV     in        Vss
Rin             in      Vss       rx_rterm
Cin             in      Vss       rx_cterm
.ENDS
.ENDL
```

射 频 和 天 线 设 计 培 训 课 程 推 荐

易迪拓培训(www.edatop.com)由数名来自于研发第一线的资深工程师发起成立，致力并专注于微波、射频、天线设计研发人才的培养；我们于 2006 年整合合并微波 EDA 网(www.mweda.com)，现已发展成为国内最大的微波射频和天线设计人才培养基地，成功推出多套微波射频以及天线设计经典培训课程和 ADS、HFSS 等专业软件使用培训课程，广受客户好评；并先后与人民邮电出版社、电子工业出版社合作出版了多本专业图书，帮助数万名工程师提升了专业技术能力。客户遍布中兴通讯、研通高频、埃威航电、国人通信等多家国内知名公司，以及台湾工业技术研究院、永业科技、全一电子等多家台湾地区企业。

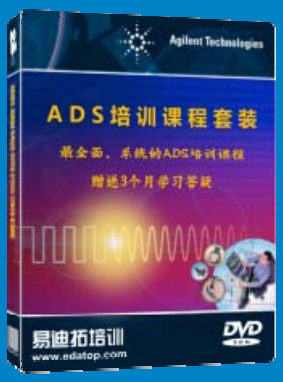
易迪拓培训课程列表：http://www.edatop.com/peixun/rfe/129.html

射频工程师养成培训课程套装

该套装精选了射频专业基础培训课程、射频仿真设计培训课程和射频电路测量培训课程三个类别共 30 门视频培训课程和 3 本图书教材；旨在引领学员全面学习一个射频工程师需要熟悉、理解和掌握的专业知识和研发设计能力。通过套装的学习，能够让学员完全达到和胜任一个合格的射频工程师的要求…

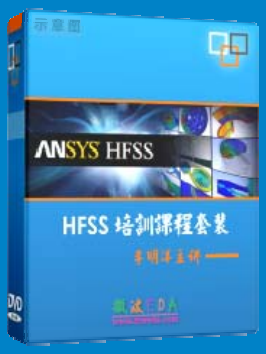
课程网址：http://www.edatop.com/peixun/rfe/110.html

ADS 学习培训课程套装



该套装是迄今国内最全面、最权威的 ADS 培训教程，共包含 10 门 ADS 学习培训课程。课程是由具有多年 ADS 使用经验的微波射频与通信系统设计领域资深专家讲解，并多结合设计实例，由浅入深、详细而又全面地讲解了 ADS 在微波射频电路设计、通信系统设计和电磁仿真设计方面的内容。能让您在最短的时间内学会使用 ADS，迅速提升个人技术能力，把 ADS 真正应用到实际研发工作中去，成为 ADS 设计专家...

课程网址： http://www.edatop.com/peixun/ads/13.html



HFSS 学习培训课程套装

该套课程套装包含了本站全部 HFSS 培训课程，是迄今国内最全面、最专业的 HFSS 培训教程套装，可以帮助您从零开始，全面深入学习 HFSS 的各项功能和在多个方面的工程应用。购买套装，更可超值赠送 3 个月免费学习答疑，随时解答您学习过程中遇到的棘手问题，让您的 HFSS 学习更加轻松顺畅…

课程网址：http://www.edatop.com/peixun/hfss/11.html

CST 学习培训课程套装

该培训套装由易迪拓培训联合微波 EDA 网共同推出，是最全面、系统、专业的 CST 微波工作室培训课程套装，所有课程都由经验丰富的专家授课，视频教学，可以帮助您从零开始，全面系统地学习 CST 微波工作的各项功能及其在微波射频、天线设计等领域的设计应用。且购买该套装，还可超值赠送 3 个月免费学习答疑…

课程网址：http://www.edatop.com/peixun/cst/24.html

HFSS 天线设计培训课程套装

套装包含 6 门视频课程和 1 本图书，课程从基础讲起，内容由浅入深，理论介绍和实际操作讲解相结合，全面系统的讲解了 HFSS 天线设计的全过程。是国内最全面、最专业的 HFSS 天线设计课程，可以帮助您快速学习掌握如何使用 HFSS 设计天线，让天线设计不再难…

课程网址：http://www.edatop.com/peixun/hfss/122.html

13.56MHz NFC/RFID 线圈天线设计培训课程套装

套装包含 4 门视频培训课程，培训将 13.56MHz 线圈天线设计原理和仿真设计实践相结合，全面系统地讲解了 13.56MHz 线圈天线的工作原理、设计方法、设计考量以及使用 HFSS 和 CST 仿真分析线圈天线的具体操作，同时还介绍了 13.56MHz 线圈天线匹配电路的设计和调试。通过该套课程的学习，可以帮助您快速学习掌握 13.56MHz 线圈天线及其匹配电路的原理、设计和调试…

详情浏览：http://www.edatop.com/peixun/antenna/116.html

我们的课程优势：

※ 成立于 2004 年，10 多年丰富的行业经验，

※ 一直致力并专注于微波射频和天线设计工程师的培养，更了解该行业对人才的要求

※ 经验丰富的一线资深工程师讲授，结合实际工程案例，直观、实用、易学

联系我们：

※ 易迪拓培训官网：http://www.edatop.com

※ 微波 EDA 网：http://www.mweda.com

※ 官方淘宝店：http://shop36920890.taobao.com