EE323000 積體電路設計導論

Introduction to HSPICE

Tutor ： Fang-Yi Chen, Benjamin You

2022/09/27

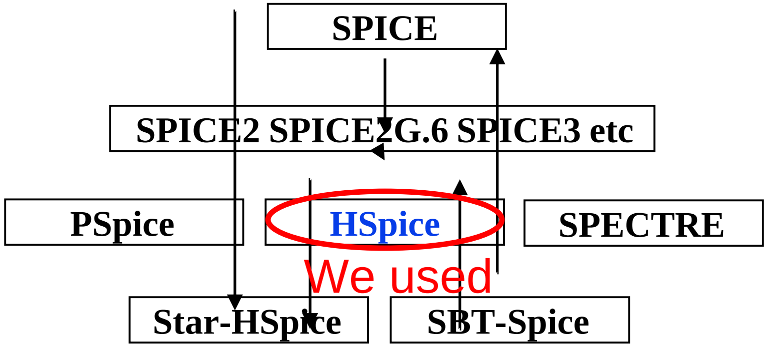
Outline

- What is SPICE, HSPICE
- Prepared Files
- Basic Syntax
- spi file
- sp file
- How to run HSPICE
- Waveform
- Monte Carlo

WHAT IS SPICE, HSPICE

What is SPICE, HSPICE

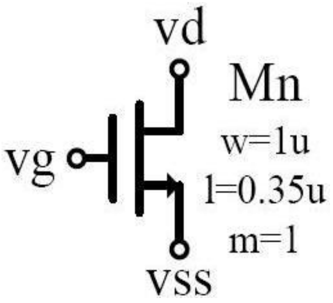
- **<u>S</u>imulation <u>P</u>rogram with <u>I</u>ntegrated <u>C</u>ircuit <u>E</u>mphasis**



- PSPICE,SPECTRE use **graphical** interface
- HSPICE use **text** interface like

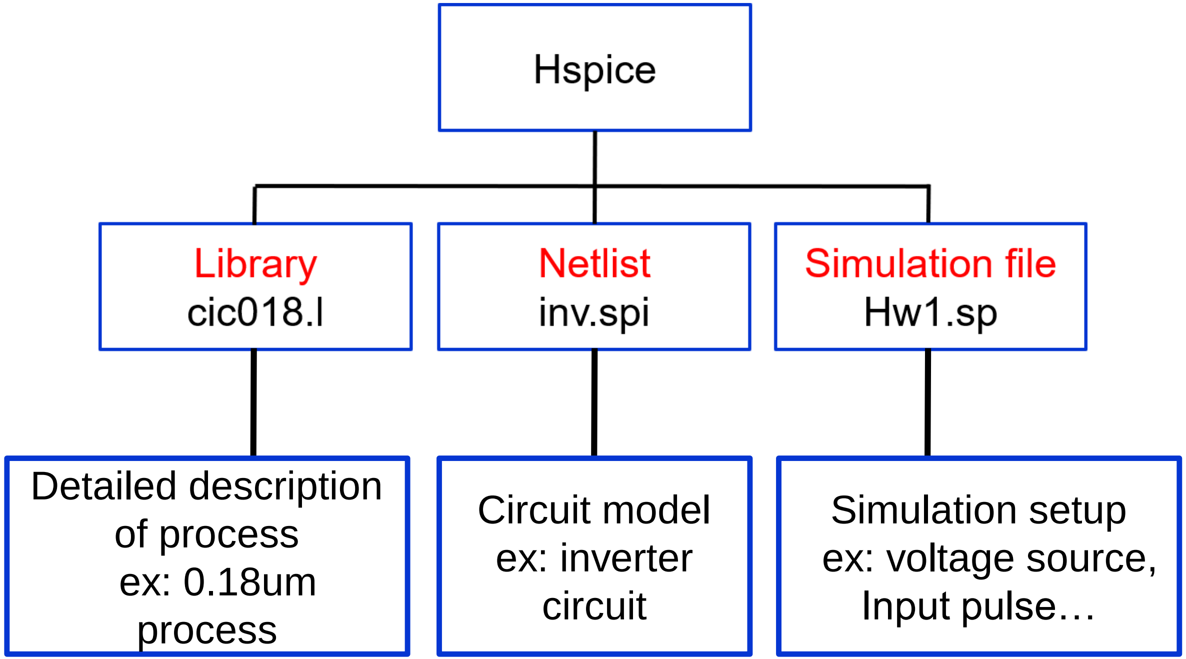
Mn vd vg vss vss n_18 w=1u l=0.35u m=1

(device name net1 net2……. type parameter value)



PREPARED FILES

Prepared Files



BASIC SYNTAX

Basic Syntax

- HSPICE won't read the 1st line of sp file.
  (usually for the title of sp file)
- Comment: add * for the whole line; $ for partial part.
- Capital and lower case letter are the same in SPICE code.
- Name net as 0, gnd meaning that it is ground (0V)
- .END(.end) must be added at the end of sp file
  .ENDS(.ends) must be added at the end of spi file
- If you cannot finish your code in one line, use + for continue you code at the next line

SPI FILE

Introduction - spi

- A script for expressing circuit model

Script in .spi file

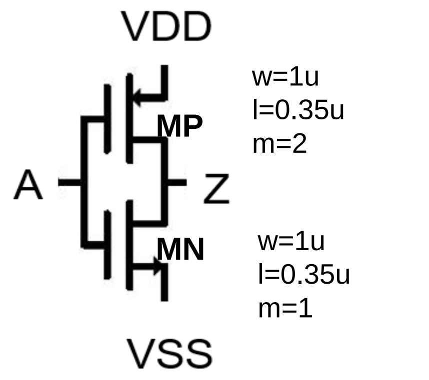
```
.subckt inv A Z VDD VSS

MP Z A VDD VDD p_18 w=1u l=0.35u m=2

MN Z A VSS VSS n_18 w=1u l=0.35u m=1

.ends
```

Circuit model (in composer)



Device – MOS (M)

What Net MOS Connect to? MOS size

- Mxxx drain gate source body channel-type width length m

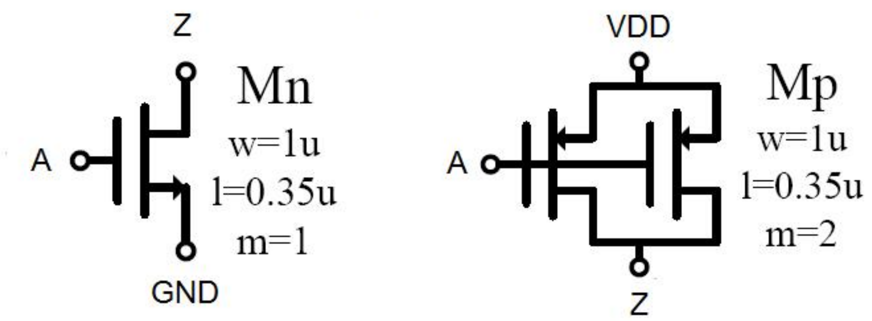
MOS Name p_18/n_18 for cic018

  - NMOS body connect to negative bias (GND)
  - PMOS body connect to positive bias (VDD)

- Ex :

MN Z A GND GND n_18 w=1u l=0.35u m=1

MP Z A VDD VDD p_18 w=1u l=0.35u m=2

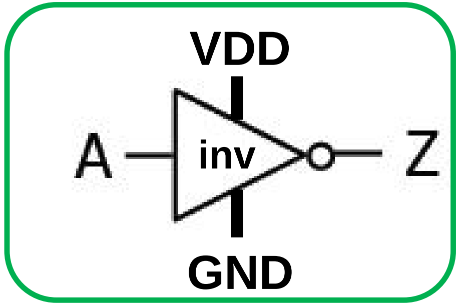


Sub-circuit

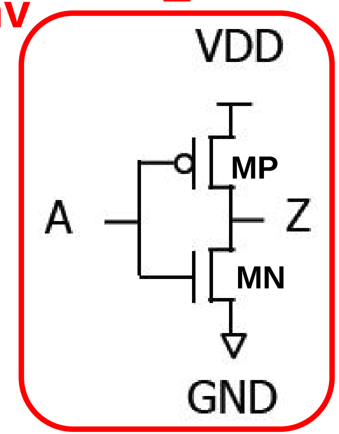
- .subckt sub-circuit name port name

  Your sub-circuit schematic

  .ends



Build a subckt which name is inv



- Ex: Inverter (inv.spi)

  *title

  .subckt inv A Z VDD GND

  MP Z A VDD VDD p_18 w=1u l=0.35u m=2

  MN Z A GND GND n_18 w=1u l=0.35u m=1

  .ends

Call sub-circuit (X)

What this subckt net connect to?

- Xxxx net1 net2 net3 net4 net5 / Subckt Origin Name

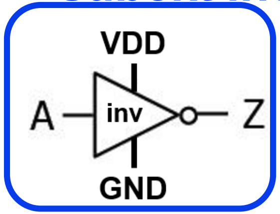
Subckt Name　　　Subckt Origin Name

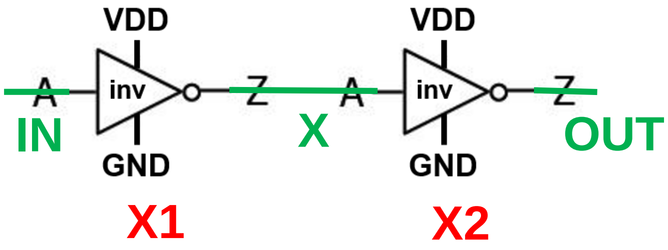
- Ex: (Inverter chain)

```
*title
.subckt inv A Z VDD GND
MP Z A VDD VDD p_18 w=1u l=0.35u m=2
MN Z A GND GND n_18 w=1u l=0.35u m=1
.ends

.subckt inv_chain IN OUT VDD GND
X1 IN X VDD GND / inv
X2 X OUT VDD GND / inv
.ends
```

subckt inv





Example (2 to 4 decoder)

2to4_decoder.spi

```
.SUBCKT NAND2 Va Vb Vout vdd vss
MP1 Vout Vb vdd vdd p_18 W=1u L=180.00n m=1
MP0 Vout Va vdd vdd p_18 W=1u L=180.00n m=1
MN1 N1_d Vb vss vss n_18 W=1u L=180.00n m=1
MN0 Vout Va N1_d vss n_18 W=1u L=180.00n m=1
.ENDS
```

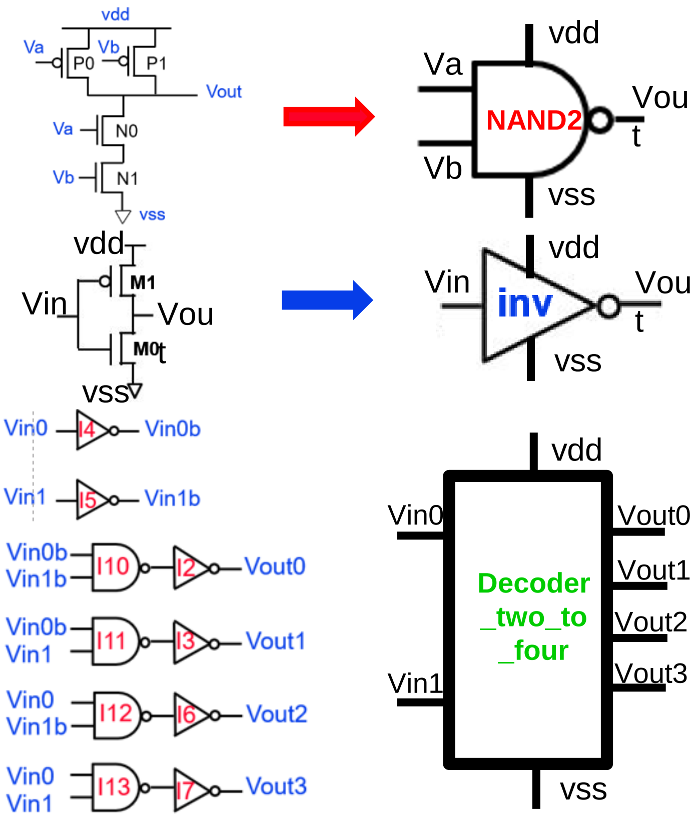
NAND2

```
.SUBCKT inv Vin Vout vdd vss
MM1 Vout Vin vdd vdd p_18 W=1u L=180.00n m=1
MM0 Vout Vin vss vss n_18 W=1u L=180.00n m=1
.ENDS
```

inv

```
.SUBCKT decoder_two_to_four Vin0 Vin1 Vout0 Vout1 Vout2 Vout3 vdd vss
XI13 Vin0 Vin1 n_out3 vdd vss / NAND2
XI10 Vin0b Vin1b n_out0 vdd vss / NAND2
XI11 Vin0b Vin1 n_out1 vdd vss / NAND2
XI12 Vin0 Vin1b n_out2 vdd vss / NAND2
XI7 n_out3 Vout3 vdd vss / inv
XI6 n_out2 Vout2 vdd vss / inv
XI5 Vin1 Vin1b vdd vss / inv
XI4 Vin0 Vin0b vdd vss / inv
XI3 n_out1 Vout1 vdd vss / inv
XI2 n_out0 Vout0 vdd vss / inv
.ENDS
```

Decoder_two_to_four



SP FILE

.sp Example (inverter analysis)

```
*title
.prot
.lib "cic018.l" TT
.temp 25
.unprot
.inc "inv.spi"
.option post
x1 IN OUT VDD GND inv
C1 OUT 0 0.05p
v1 VDD 0 1.8
v2 GND 0 0
v3 IN 0 0
.dc v3 0 1.8 0.01
.probe V(OUT)
.end
```

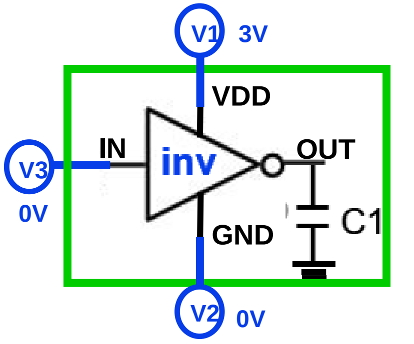
Simulation Condition Setup

Circuit Setup

Voltage Apply

Simulation Setup

- title
- The content between .prot and .unprot will not display in the .lis file. It was used when you have some secret circuits or some useless information so you don't want to print in the .lis file.
- include netlist (.spi)
- create post analysis file
- call inv sub-circuit
- voltage source
- simulation analysis
- output node voltage
- file end



Simulation Condition: TT 25°C

Condition Setup - Corner Condition

- TT: Typical NMOS and PMOS
- FF: Fast NMOS and Fast PMOS
- SS: Slow NMOS and Slow PMOS
- SF: Slow NMOS and Fast PMOS
- FS: Fast NMOS and Slow PMOS

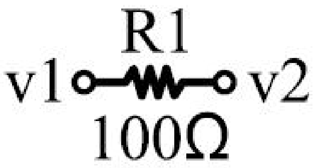
```
*title
.prot
.lib "cic018.l" TT      Corner Setting
.temp 25                Temperature Setting
.unprot
.inc "inv.spi"
.option post
x1 IN OUT VDD GND inv
C1 OUT 0 0.05p
v1 VDD 0 1.8
v2 GND 0 0
v3 IN 0 0
.dc v3 0 1.8 0.01
.probe V(OUT)
.end
```

Circuit Setup - Device

- **Resistor**

  Rxxx net1 net2 R_value(Ω)

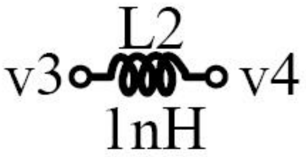
  Ex ： R1 v1 v2 100



- **Inductor**

  Lxxx net1 net2 L_value(H)

  Ex ： L2 v3 v4 1n



- **Capacitor**

  Cxxx net1 net2 C_value(F)

  Ex ： C3 v5 v6 2p

Device table

- Device table

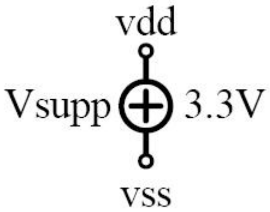
Voltage Source	V
Current Source	I
Resistor	R
Inductor	L
Capacitor	C
MOS	M
Sub Circuit	X

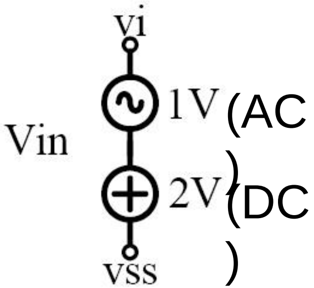
- Unit table

Unit	Meaning	Unit	Meaning
t	1e12	m	1e-3
g	1e9	u	1e-6
Meg or x	1e6	n	1e-9
K	1e3	p	1e-12
dB	20log	f	1e-15

Voltage Apply - Source

❑ Voltage source





Vxxx net1 net2 DC=? V AC=? V

Ex : Vsupp vdd vss DC=3.3

Ex : Vin vi vss DC=2 AC=1

Vxxx net1 net2 **SIN vdc vamp freq td df phase**

Ex : VSIN in 0 SIN 1.65 0.1 1Meg

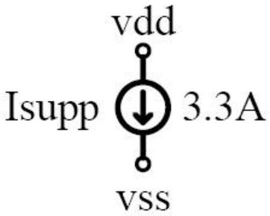
Vxxx net1 net2 **PULSE v1 v2 td tr tf pw period**

Ex : VPULSE in 0 PULSE 0 3.3 1n 0.1n 0.1n 0.9n 2n

Vxxx net1 net2 **PWL t1 v1 t2 v2 t3 v3……..**

Ex : VPWL in 0 PWL 0 0 1n 0 1.1n 3.3 2n 3.3 2.1n 0 3n 0

❑ Current source

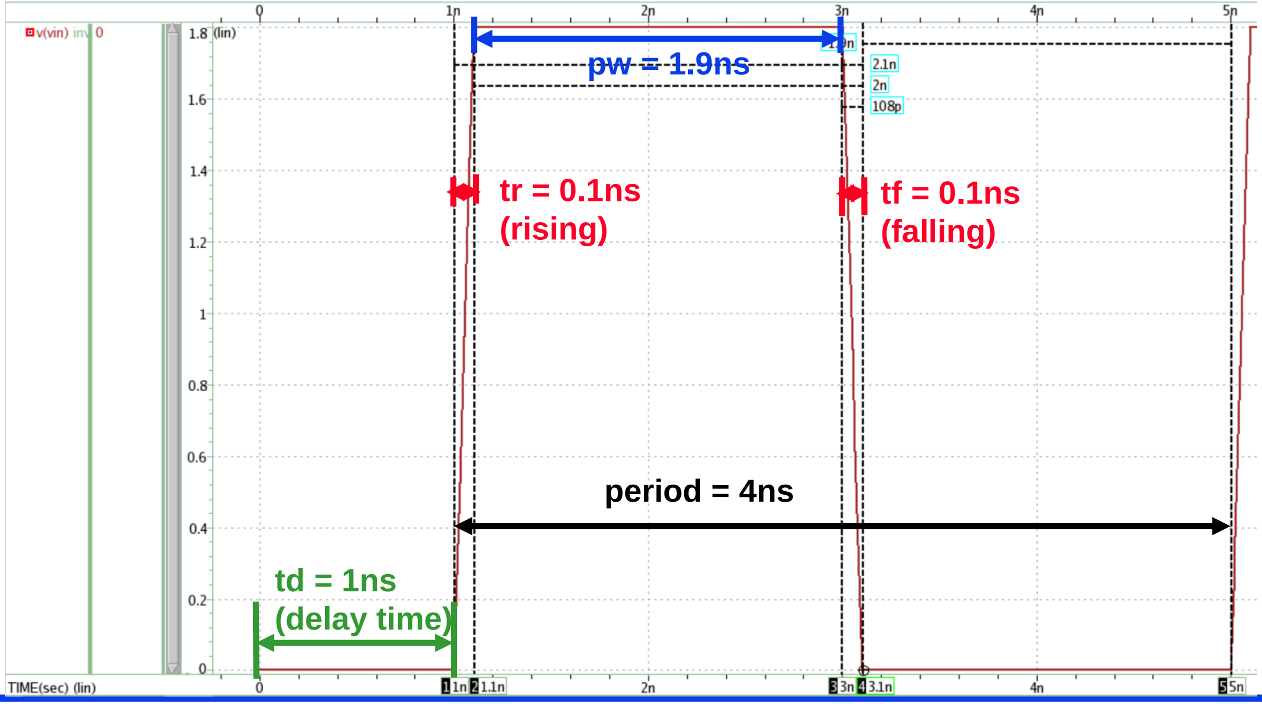


Ixxx net1 net2 DC=? A AC=? A Ex : Isupp vdd vss DC=3.3

Voltage Source: PULSE

- Vxxx net1 net2 **PULSE v1 v2 td tr tf pw period**

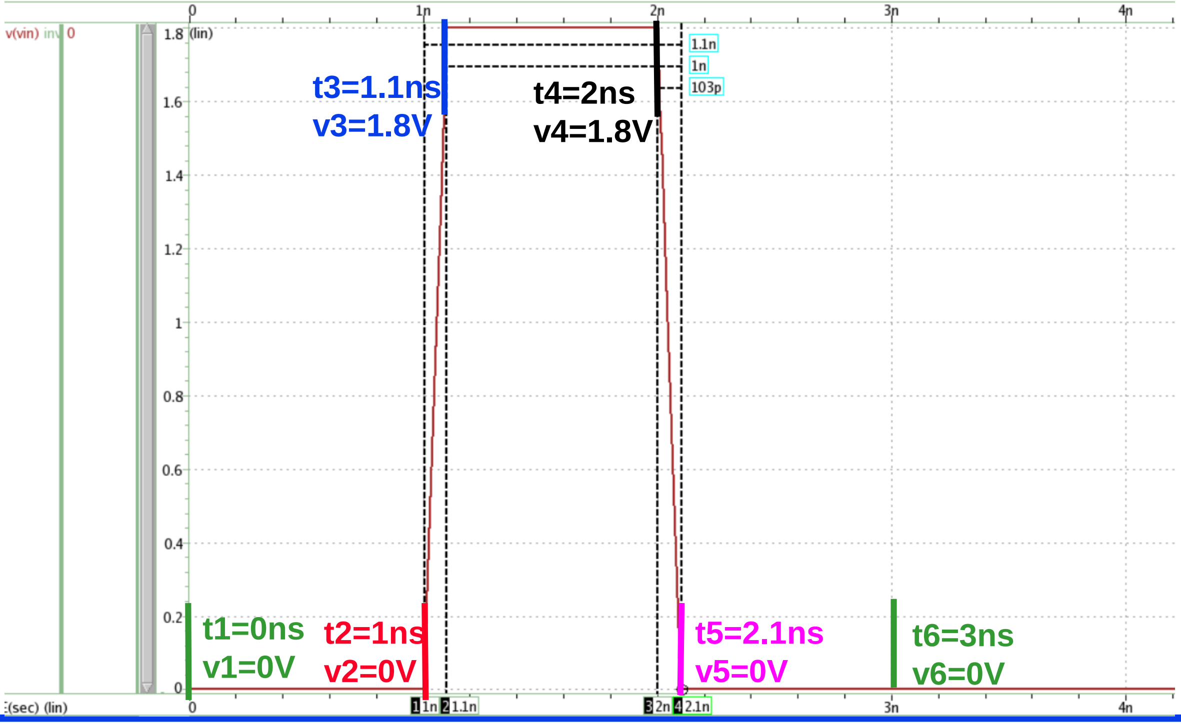
Ex : VPULSE VIN 0 PULSE 0 1.8 1n 0.1n 0.1n 1.9n 4n



Voltage Source: PWL

- V xxx net1 net2 **PWL t1 v1 t2 v2 t3 v3……..**

  Ex : VPWL VIN 0 PWL 0 0 1n 0 1.1n 1.8 2n 1.8 2.1n 0 3n 0



Simulation Setup – Initial

- **.ic: Initializes conditions for .TRAN**

  Syntax. .ic V(node1) = val1 V(node2) = val2 …
  Example. .ic V(in) = 3 V(x1.x2.in1) = 1

  **(In .tran analysis, the initial condition of node "in" is 3V.
  In x2 subckt of x1 subckt, there is a node "in1" = 1V)**

- **.nodeset: Initializes specified node voltages for .DC operating point analysis**

  Syntax. .nodeset V(node1) = val1 <V(node2) = val2>
  Example. .nodeset V(in) = 3 V(x1.x2.in1) = 1

  (In .dc analysis, the initial condition of node "in" is 3V.

  In x2 subckt of x1 subckt, there is a node "in1" = 1V)

Simulation Setup - Analysis

- **.OP: Analyze operation point of nodes in circuit.**

Syntax.	.OP
Example.	.OP

- **.DC: DC analysis to sweep parameter, source and temperature values.**

Syntax.	.DC	<var1>	<start>	<stop>	<step>
Example.	.DC	Vgs	0	5	0.1

- **.AC: AC analysis to sweep frequency.**

Syntax.	.AC	<DEC/LIN>	<np>	<start>	<stop>
Example.	.AC	DEC	10	1KHz	10MHz

- **.Tran: Transient analysis to sweep time.**

Syntax.	.Tran	<step>	<stop>
Example.	.Tran	1ns	1us

Simulation Setup – OP Analysis

- **.OP: Analyze operation point of nodes in circuit.**

Syntax. .OP

Example. 1) .OP 15ns

(It will print out the voltages and currents from 0ns to 15 ns of all the devices in .lis file)

2) .OP .5ns CUR 10ns VOL 17.5ns 20ns 25ns

(It calculates operating point voltages and currents for the DC solution, as well as currents at 10 ns, and voltages at 17.5 ns, 20 ns and 25 ns for the transient analysis.)

Simulation Setup – DC Analysis

- **.DC: DC analysis to sweep parameter, source and temperature values.**

Syntax.	.DC	<var1>	<start>	<stop>	<step>
Example.	1) .DC	Vgs	0	5	0.1

(Voltage source named Vgs sweeps from 0 to 5V and stops by every 0.1V)

2) .param wp = 0.5u

.DC v3 0.5 1.8 0.1 sweep wp 0.5u 10u 0.1u

(wp=0.5u, sweep v3 from 0.5V to 1.8V with step 0.1V
wp=0.6u, sweep v3 from 0.5V to 1.8V with step 0.1V
wp=0.7u, sweep v3 from 0.5V to 1.8V with step 0.1V
….
wp=10u, sweep v3 from 0.5V to 1.8V with step 0.1V)

Sweep v3 from 0.5V to 1.8V with step 0.1V

Simulation Setup – AC Analysis

- **.AC : AC analysis to sweep frequency.**

Syntax. .AC <DEC/LIN> <np> <start> <stop>

Example. 1) .AC DEC 10 1K 100MEG

(A frequency sweeps by 10 points per decade from 1kHz to 100 MHz)

2) .AC DEC 10 1 10K SWEEP cload LIN 20 1pf 10pf

(An AC analysis for each value of cload, which results from a linear sweep of cload between 1 pF and 10 pF (20 points), sweeping frequency by 10 points per decade from 1 Hz to 10 kHz)

3) .AC DEC 10 1 10K SWEEP MONTE=30

(A frequency sweep along with a Monte Carlo analysis with 30 trials)

Simulation Setup – Tran Analysis

- **.Tran: Transient analysis to sweep time.**

Syntax. .Tran <step> <stop>

Example. 1) .Tran 1ns 1us

(Perform and print the transient analysis every 1 ns for 1 us)

2) .TRAN 0.1 25NS 1NS 40NS START=10NS

(Perform the calculation every 0.1 ns for the first 25 ns, and then every 1 ns until 40 ns; the printing and plotting begin at 10 ns)

Analysis

- **.Probe ： Probe the observation will not show in the result file but can be seen at Waveviews**

  .probe V(net) I(device)

  .probe V(*) I(*) : print all voltages and currents of nodes

- **.Print ： Print the observation in the result file**

  .print V(net) I(device)

- **.Plot ： Plot the observation in the result file (Funny Plot)**

  .plot V(net) I(device)

Optional Analysis - SWEEP

- **SWEEP: Additional nested sweep analysis. Sweep parameter, source or temperature values but not model parameters.**

Syntax. <Analysis> **SWEEP** <var> <start> <stop> <step>
or <Analysis> **SWEEP** <var> <DEC/LIN> <np> <start> <stop>

Example:

1) .DC Vds 0 5 1 **SWEEP** Vgs 2 5 1

(Vgs=2V, sweep Vds from 0 to 5V with unit 0.1V
Vgs=3V, sweep Vds from 0 to 5V with unit 0.1V
Vgs=4V, sweep Vds from 0 to 5V with unit 0.1V
Vgs=5V, sweep Vds from 0 to 5V with unit 0.1V)

2) .AC DEC 10 1K 10M **SWEEP** RL 10K 30K 10K

3) .Tran 1ns 1us **SWEEP** TEMP 0 100 10

Optional Analysis – ALTER example

- **.ALTER: Alter condition and repeat analysis.**

```
*title
.prot
.lib "cic018.l" TT
.temp 25
.unprot
.inc "inv.spi"
.option post
```

Simulation Condition Setup

```
x1 IN OUT VDD GND inv
C1 OUT 0 0.05p
```

Circuit Setup

```
v1 VDD 0 1.8
v2 GND 0 0
v3 IN 0 0
```

Voltage Apply

```
.dc v3 0 1.8 0.01
.probe V(OUT)
.end
```

Simulation Setup

Add it between purple and black block

```
.alter
.lib 'cic018.l' ff
.temp -40
.alter
.lib 'cic018.l' ss
.temp 125
.alter
.lib 'cic018.l' sf
.temp 25
.alter
.lib 'cic018.l' fs
.temp 25
```

After simulation, it will produce sw0~sw4 (total 5 files) (TT 25 / FF -40 / SS 125 / SF 25 / FS 25)

Measure

- **Function from to**

Syntax.   .meas <Analysis> <Result_Var> **FUNC <**V(net)> **From To**

Ex: **.meas** Tran MaxValue MAX V(out) from=1n to=10n

Syntax.   .meas <Analysis> <Result_Var> **FUNC <**V(net)> **AT**

Ex: **.meas** DC DC_Gain DERIV V(out) at='0.5*3.3'

(½ VDD)

- *FUNC can be MIN, MAX, AVG, RMS, DERIV, INTEG*

- **PARAM**

Syntax.   .meas <Analysis> <Result_Var> **PARAM=('')**

Ex: **.meas** AC AC_GAIN PARAM=('Vdb(out)-Vdb(in)')

Measure

- **Trigger and Target**

  Syntax.   .meas <Analysis> <Result_Var> **TRIG** <V(netx)> <val=X> <Slew> **TARG** <V(nety)> <val=Y> <Slew>

  Ex:

  .meas tran DelayTime **Trig** V(in) val=0.9V rise=1 **Targ** V(out) + val=0.9V fall=1

  (Measure delaytime: Started pt. (Trig) starts from the 1^{st} rising edge of V(in) at 0.9V ; Ended pt. (Targ) ends ~~at the 1^{st} falling~~ edge when V(out)=0.9V)

- *Analysis can be DC, AC, Tran*
- *Val means value of V(net)*
- *Slew can be rise, fall, cross (special word “last”, ex: fall=last means the last falling edge)*

Measure

- **Find When**

  Syntax. .meas <Analysis> <Result_Var> **FIND** **<**V(net)> **WHEN**

  \+ condition

  Ex: .meas DC TriggerPt **FIND** V(Vin) **WHEN** V(Vout)=0.9

- **Find At**

  Syntax. .meas <Analysis> <Result_Var> **FIND** **<**V(net)> **AT**

  \+ condition

  Ex: .meas TRAN Volt **Find** V(out) **at**=10n

Analysis result file table

Ouput File Type	Extension
Output listing	.lis
Transient analysis results	.tr#
DC analysis results	.sw#
AC analysis results	.ac#
Transient analysis measurement results	.mt#
DC analysis measurement results	.ms#
AC analysis measurement results	.ma#
FFT analysis graph data files	.ft#
Output status files	.st#
Nets operation voltages	.ic#

HOW TO RUN HSPICE

How to run HSPICE

- Make sure all files are include in sp file (next page)
- Enter the folder of sp file, type hspice xxx.sp >! xxx.lis

  or hspice -i xxx.sp -o xxx.lis

  input xxx.sp **output xxx.lis**

- You can find some useful information in the .lis file
- If there is no error, terminal will print out job concluded

  otherwise, terminal will print out job aborted

  (you can find errors in .lis file)

- source /usr/cadtool/user_setup/08-hspice.csh

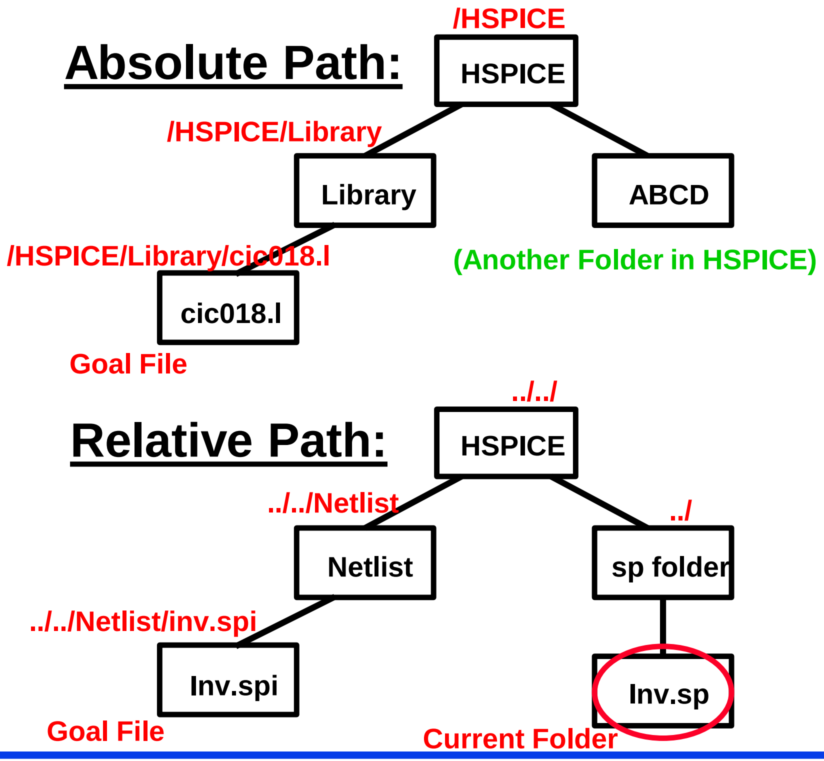
How to run HSPICE – Path Setting

- Make sure all files(spi file & lib file) are include in sp file

(Absolute Path)

```
*title
.prot
.lib "/HSPICE/Library/cic018.l" TT
.temp 25
.unprot
.inc "../../Netlist/inv.spi"
.option post
x1 IN OUT VDD GND inv
v1 VDD 0 1.8
v2 GND 0 0
v3 IN 0 0
.dc v3 0 1.8 0.01
.probe V(OUT)
.end
```

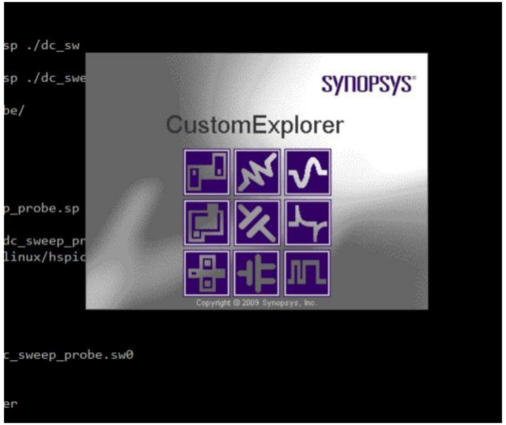
(Relative Path)

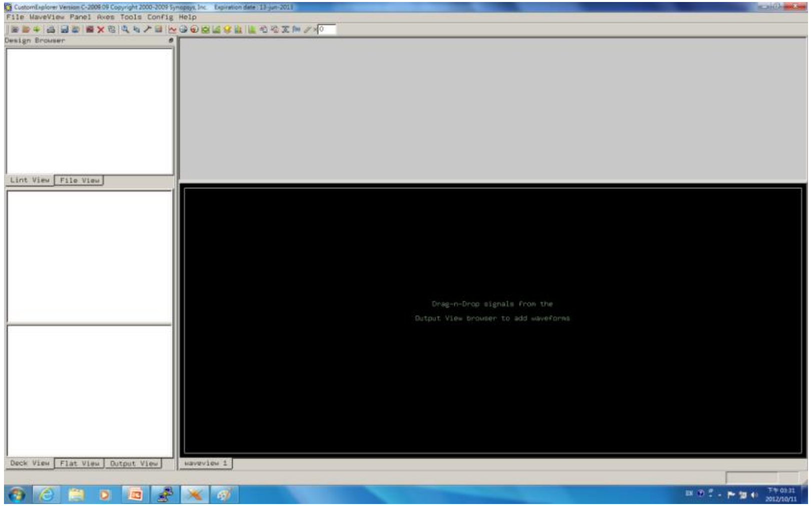


WAVEFORM

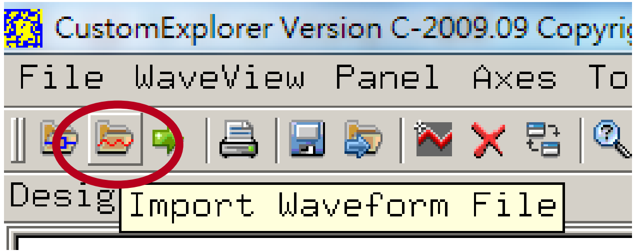
Waveform – Open Waveview

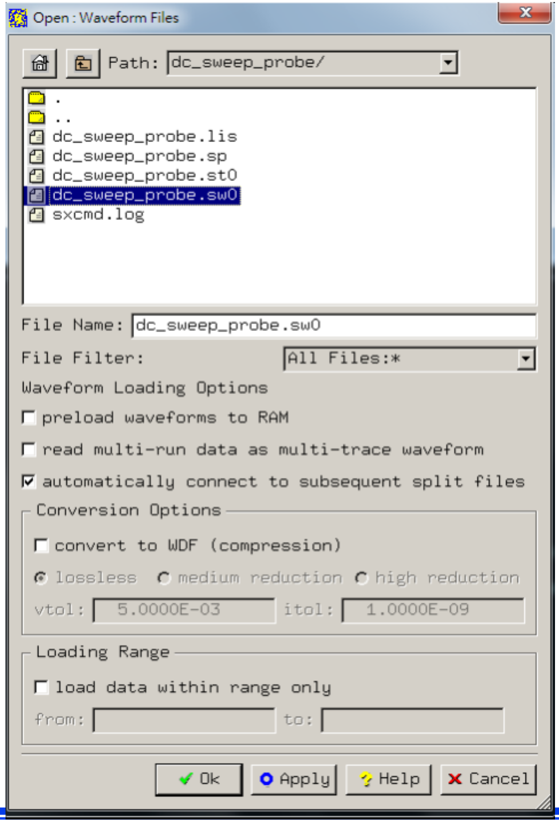
- Type wv & at terminal
- source /usr/cadtool/user_setup/08-customexplorer.csh



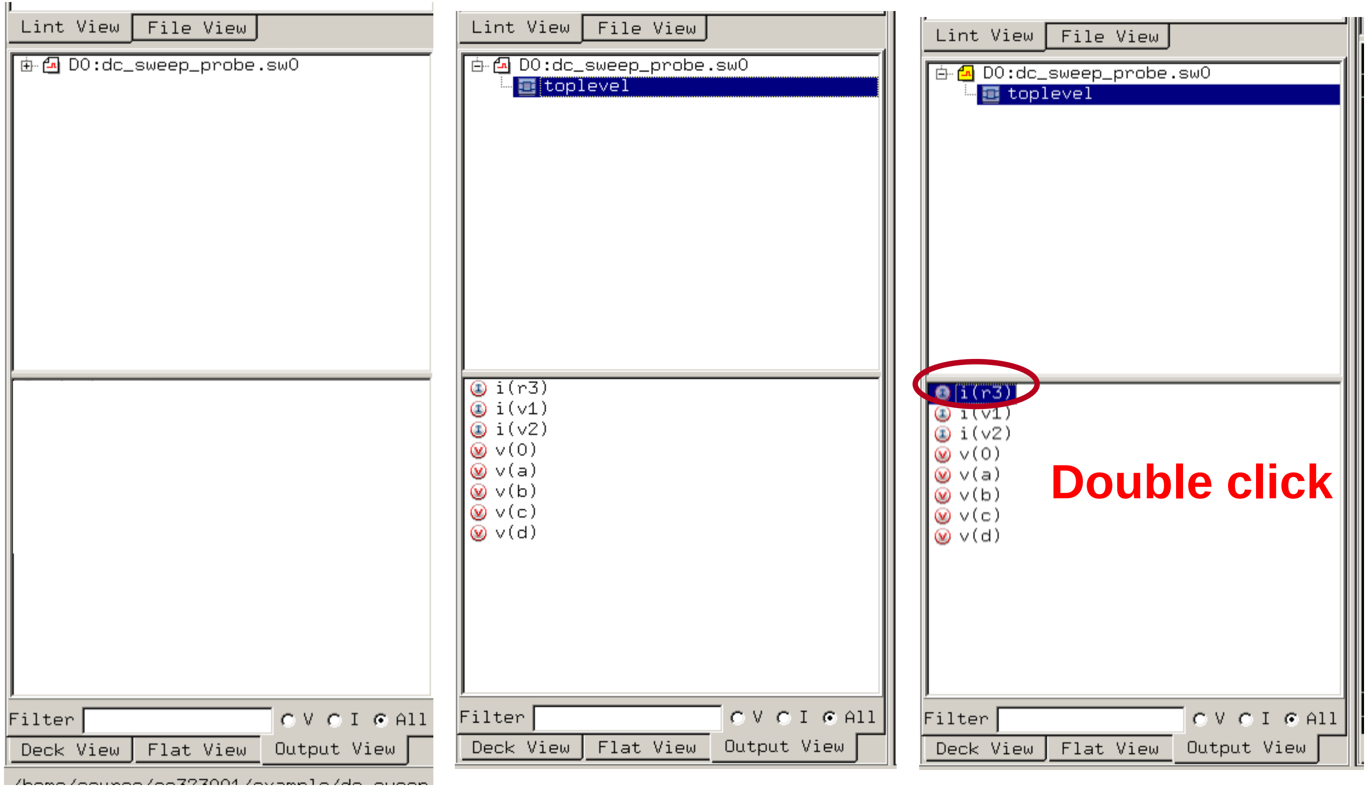


Waveform – Open File

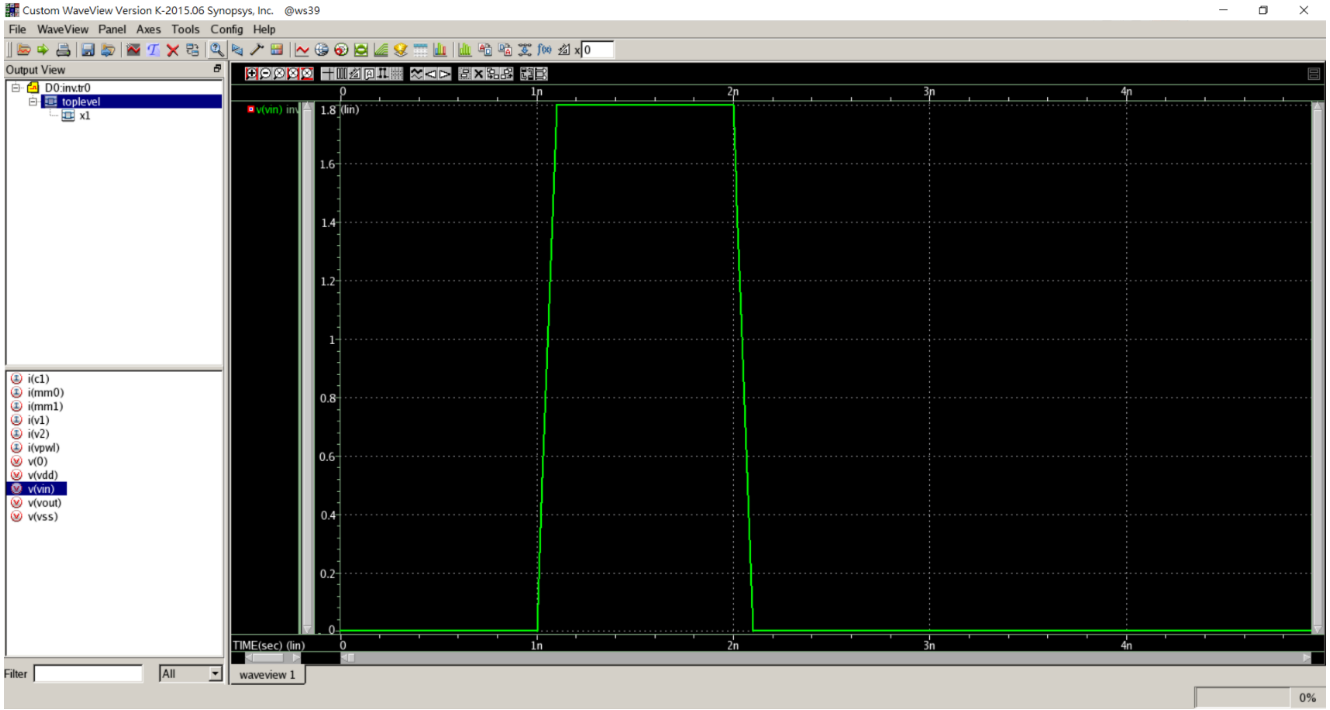




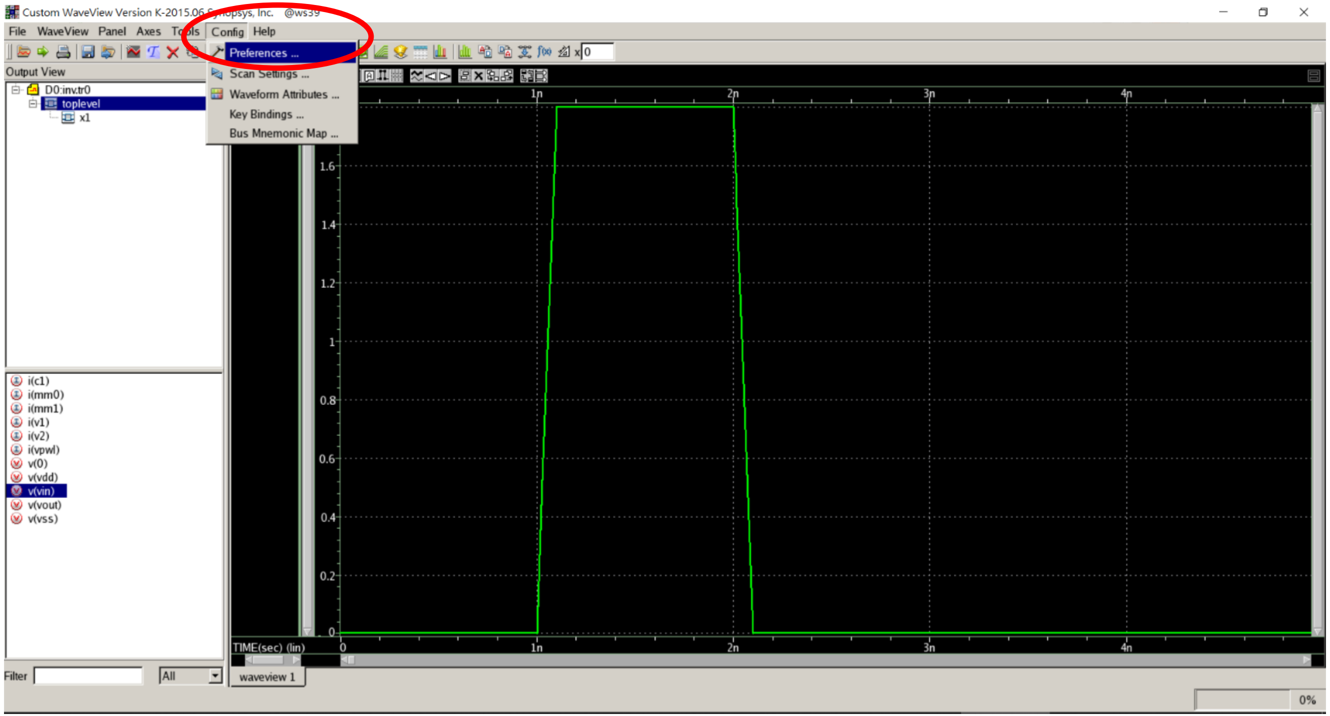
Waveform – Open File



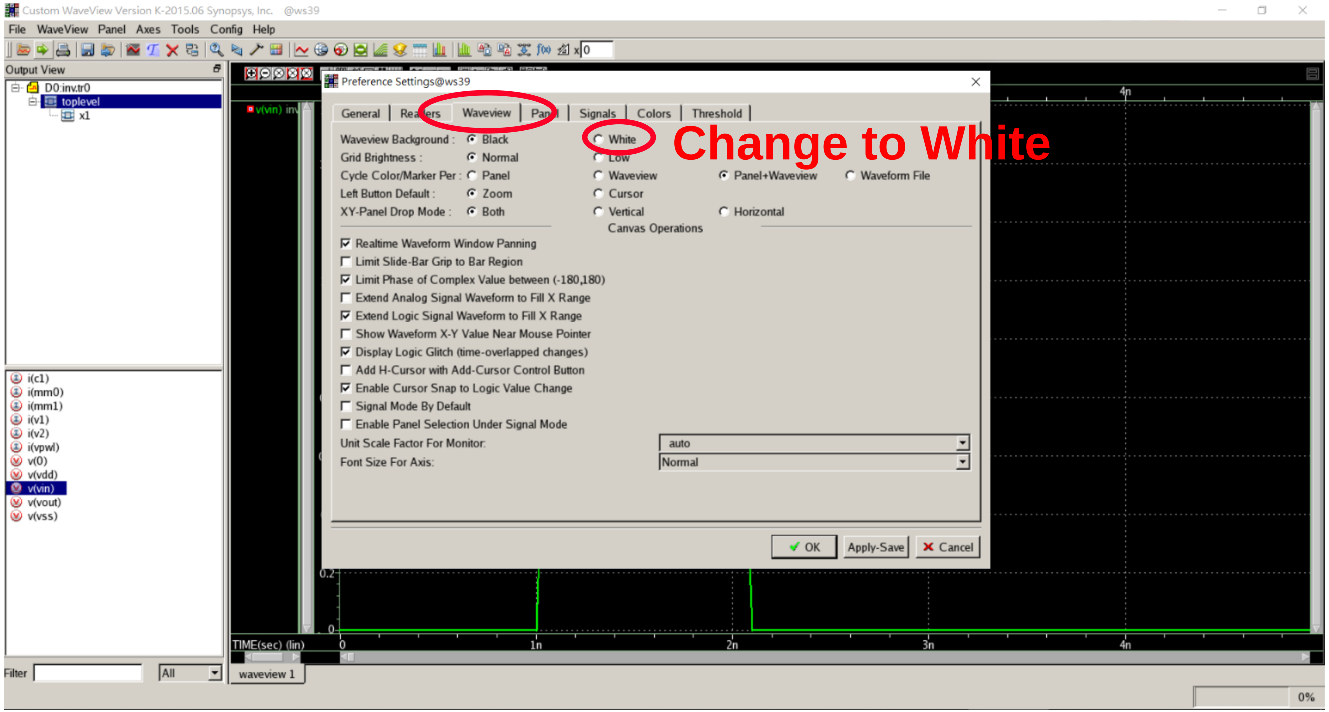
Waveform: change background color-1



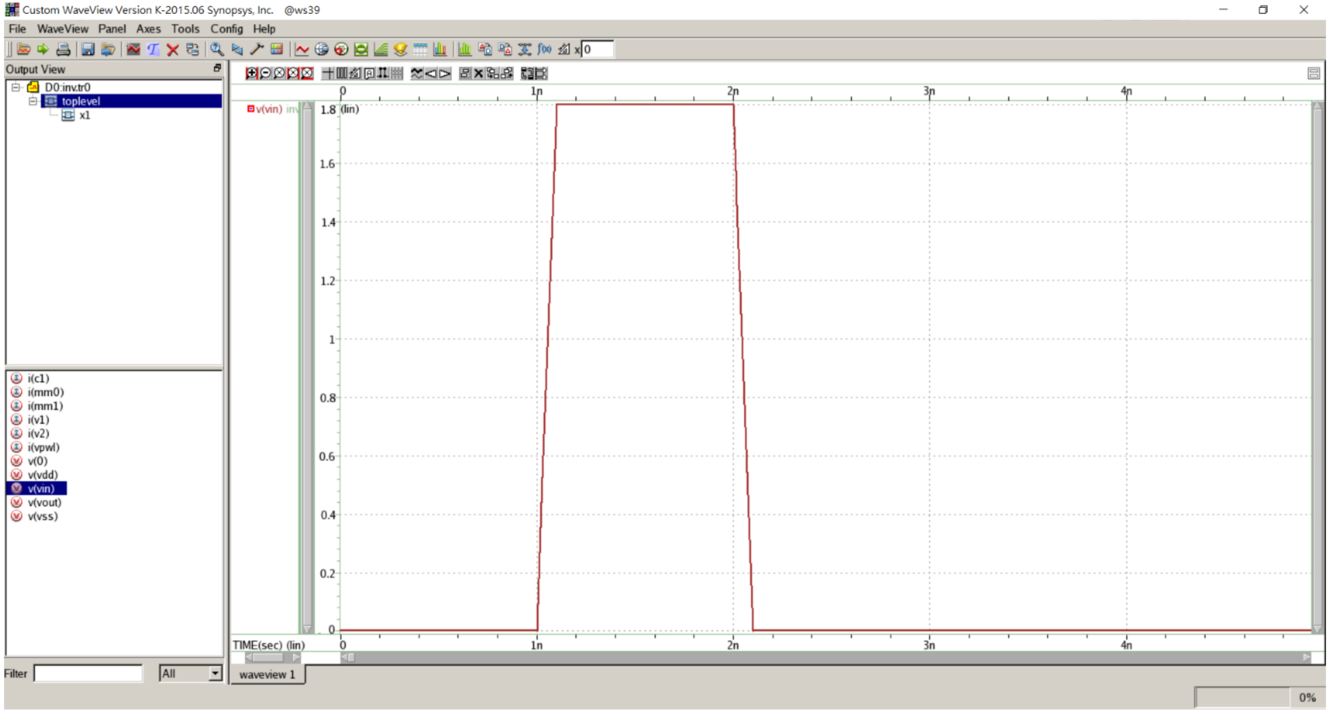
Waveform: change background color-2



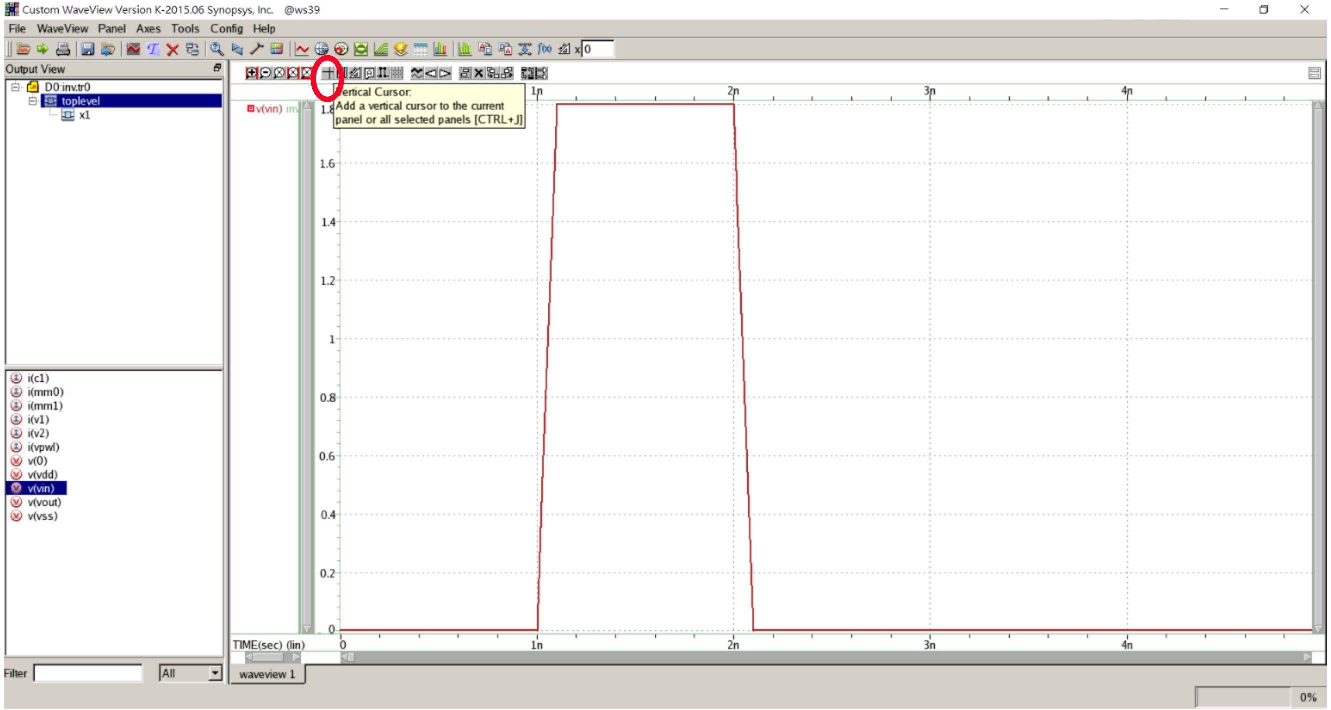
Waveform: change background color-3



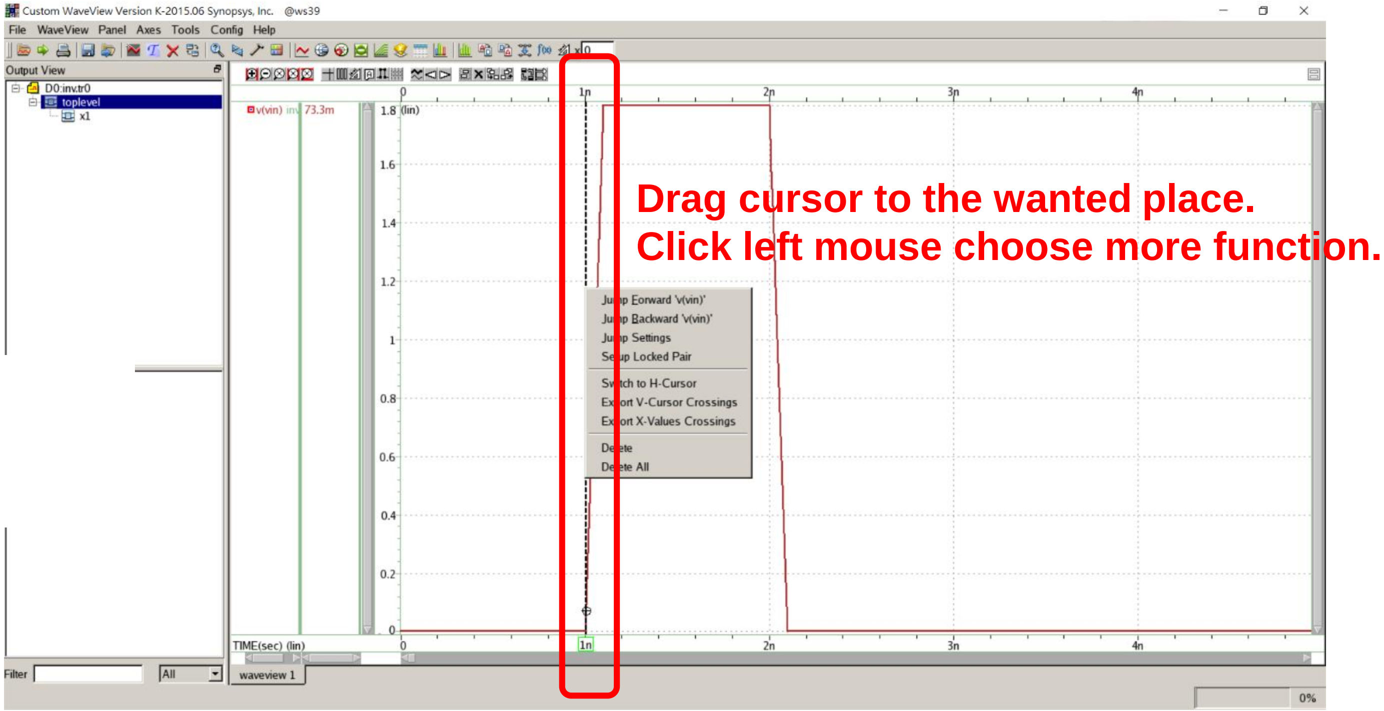
Waveform: change background color-4



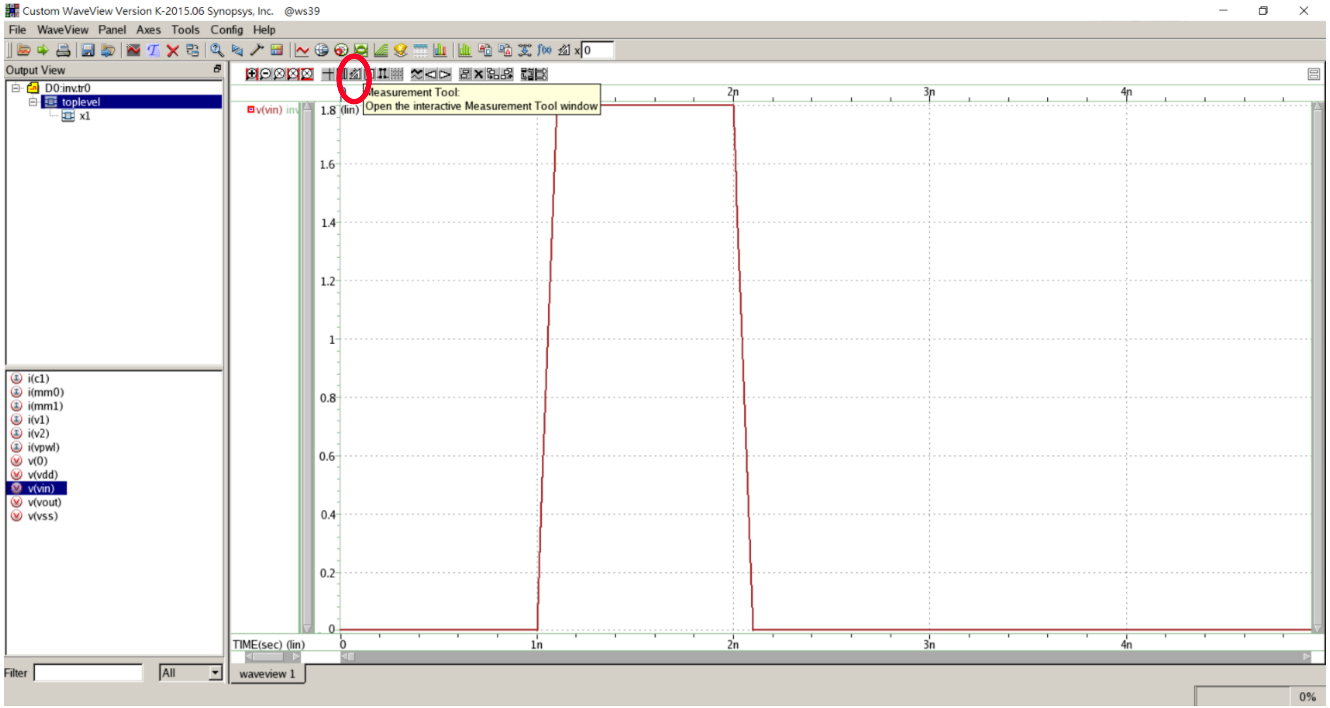
Waveform: Add Cursor - 1



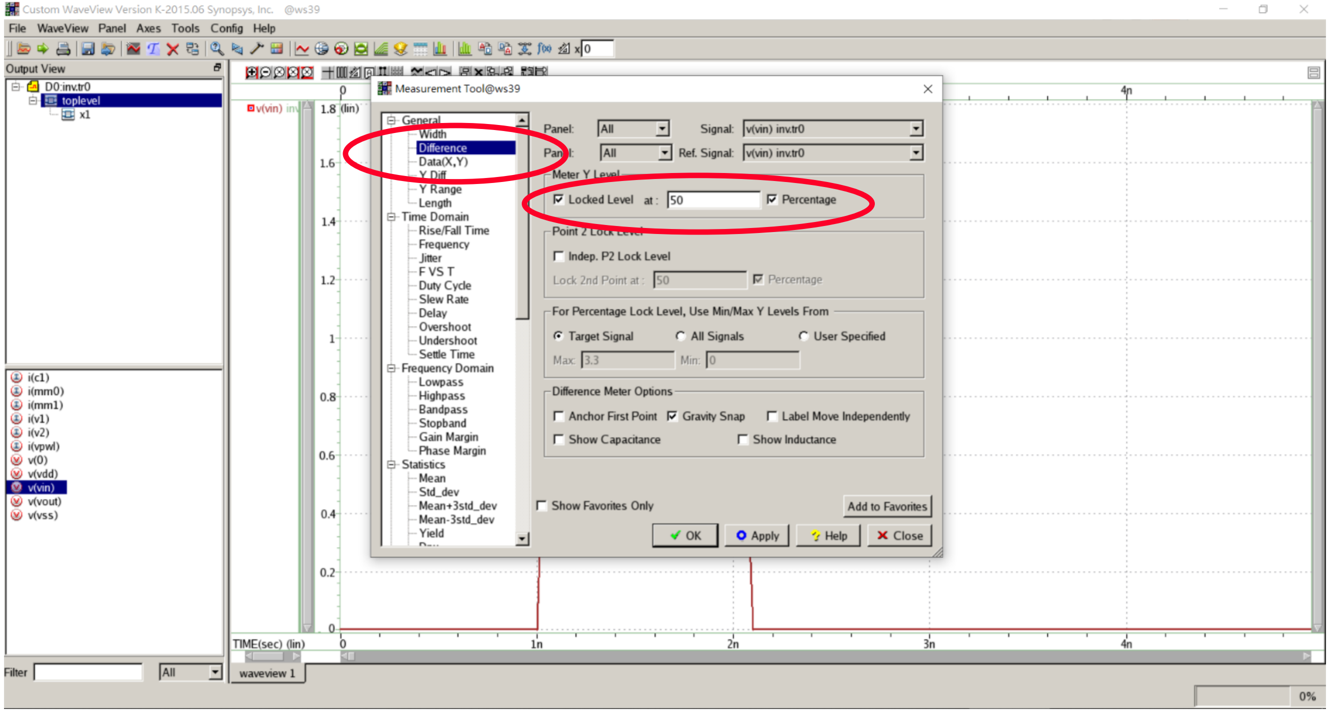
Waveform: Add Cursor



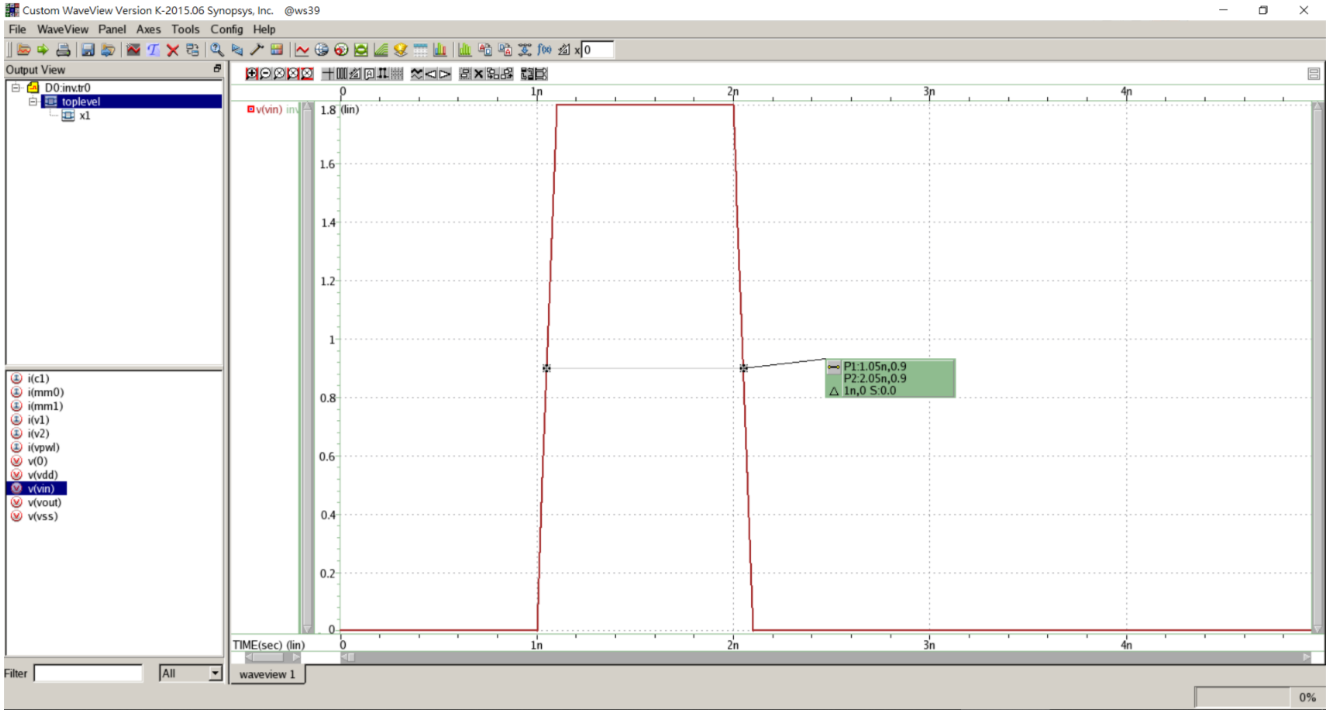
Waveform: Measurement tool



Waveform: Measurement tool



Waveform: Measurement tool



MONTE CARLO

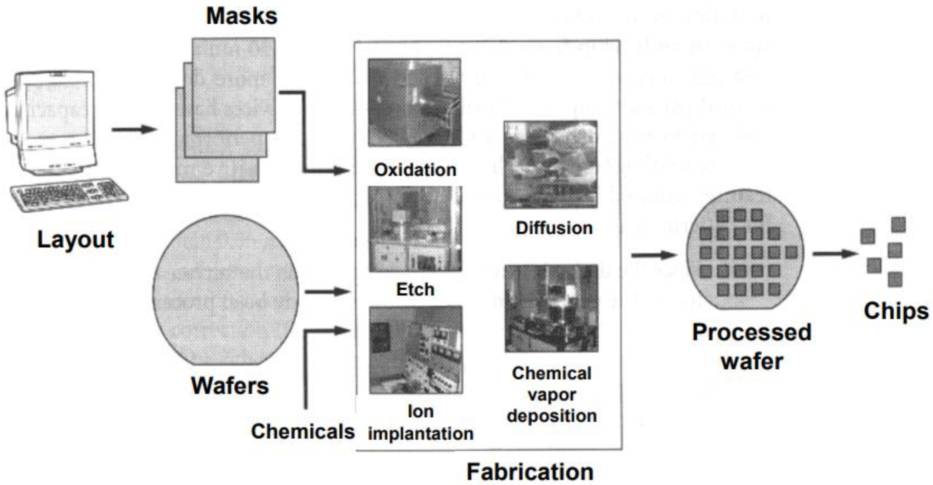
What is Monte Carlo ?

- There are process variations when producing ICs.

  (i.e ICs aren’t the same when producd by foundry)

  → run “Monte Carlo” by using Gaussian distribution

  to **simulate process variations conditions**.



Distribution for M.C. simulation in CIC018

- .PARAM X=AGAUSS(**nominal_val**, **abs_variation**, **sigma**) [in spi file]
  - **nominal_val:** Nominal value in Monte Carlo analysis.
  - **abs_variation:** AGAUSS vary the nominal_val by +/- abs_variation.
  - **sigma:** Specifies abs_variation at the sigma level. For example, if sigma=3, then the standard deviation is abs_variation divided by 3.
- MPUL Q QB VDD VDD p_18 w=250n l=180n m=1 **delvto=X** [in spi file]
  - **delvto**: Zero-bias threshold voltage shift. Default=0.
- .dc(tran) …….. **sweep monte=32000** [in sp file]
- AGAUSS distribution in cic018 stands for Vth distribution
  - Gauss distribution example in CIC018: AGAUSS(0,0.072,6)
- **Set different parameter(X, Y, Z ….) for each transistor mismatch!!**

Example

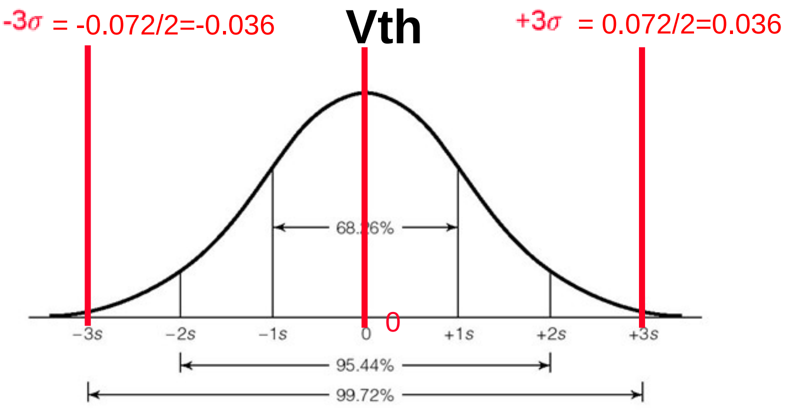
Inv.sp

```
*title
.prot
.lib "cic018.l" TT
.temp 25
.unprot
.inc "inv.spi"
.option post
x1 IN OUT VDD GND inv
C1 OUT 0 0.05p
v1 VDD 0 1.8
v2 GND 0 0
v3 IN 0 0
.dc v3 0 1.8 0.01 sweep monte=1024
.probe V(OUT)
.end
```

Inv.spi

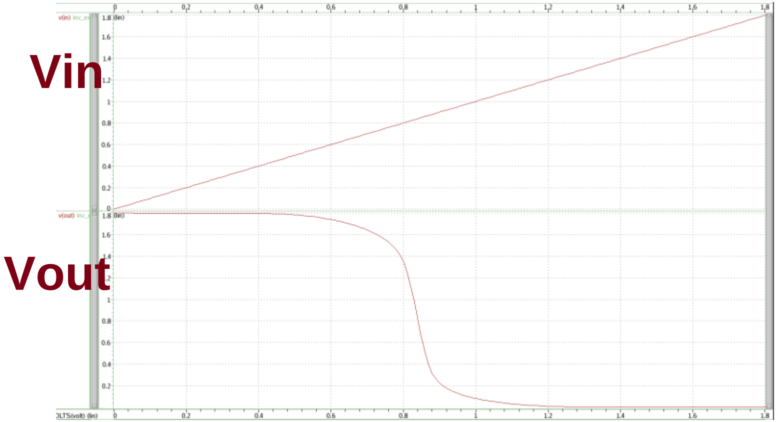
```
.PARAM VTH=AGAUSS(0,0.072,6)

.SUBCKT inv Vin Vout VDD VSS
MM1 Vout Vin VSS VSS n_18 W=1u L=0.18u m=1 delvto=VTH
MM0 Vout Vin VDD VDD p_18 W=2u L=0.18u m=1 delvto=VTH
.ENDS
```

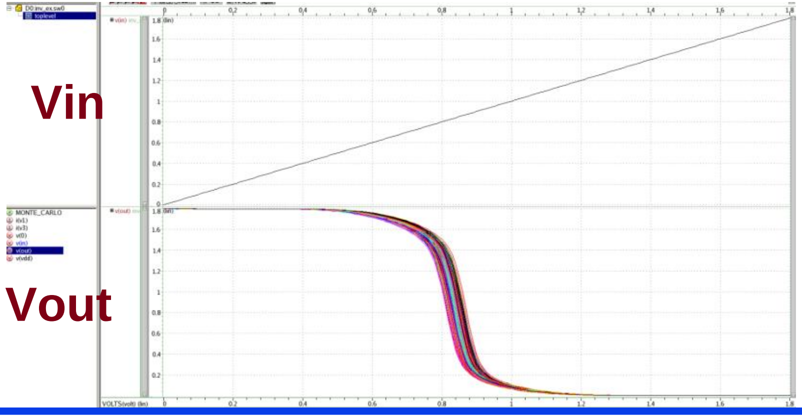


Example - Waveform

- Original Simulation Result
  (without redline code in p.53)



- Monte Carlo 1024 times
  Simulation Result



THE END