N16FFCLL V1.0 SRAM SPICE Model Usage Guide

Outline

- Model Overview
- Guideline for Eldo Usage
- Model Package
- Usage file and flag settings
- Usage of variation model
- Netlist for usage of variation model

Simulation Environment

- **Simulation Environment**
  - **Simulator:**
    - Hspice Version
      - ✓ I-2013.12-SP2-1
    - Spectre Version
      - ✓ MMSIM14.1_ISR12
    - Eldo Version
      - ✓ 15.1
  - **O/S:**
    - RH COMPILER/OS VERSION
      - ✓ gcc 4.1.2 / Red Hat Enterprise Linux AS release 4
    - `IN COMPILER/OS VERSION
      - ✓ Visual Studio 2008 / Windows 7
    - SUSE COMPILER/OS VERSION
      - ✓ gcc 4.1.0 / SUSE Linux Enterprise Server 10
    - SUN COMPILER/OS VERSION
      - ✓ SUNWspro (SUN Studio 11) cc 5.8 / Solaris 10 SPARC
    - SUN X86 COMPILER/OS VERSION
      - ✓ SUN Studio 12.1 cc 5.10 / Solaris 10 X86

Guideline for Eldo Usage

- **For users who use Eldo with Spice format netlist, the following command is needed for simulation.**
  - Add one more argument “-compat” to Eldo command line.

Command

```
ln24:/spice2> eldo –compat -i idsat.sp –o idsat.out
```

Command add “-compat”

Netlist

Netlist keeps unchanged

```
* netlist for Idsat of DUT
*-----------------------------------------------------
.temp 25
.option brief=1
.lib ‘usage.l’ TT_SRAM
.lib ‘usage.l’ pre_simu_sr
.option brief=0

Vds0 d0 0 0.85
xmdut0 d0 g s b nchpg_hcsr_mac nfin=2 l=0.02u
```

  - Please note if the native Eldo binary is wrapped in a shell script in the working system, user needs to add the argument “-compat” to Eldo command line in the script.

shell script

```
/AMS_aol_12.2a_patch3/bin/eldo -i $argv[1] -o $argv[2]

/AMS_aol_12.2a_patch3/bin/eldo -compat -i $argv[1] -o $argv[2]
```

Model Package

- The **TMI model package** provided by TSMC will contain the **model file**, **model usage files** and the **compiled shared libraries** for different OS platforms. Please put the overall files of the model package in the same directory to do the model simulation.

- All necessary option settings and the relative path of the compiled shared libraries have been set in the model usage files. Simulators will detect the settings in the model usage files and automatically link the correct TMI library for the corresponding platform. Just make sure that your simulators support TMI model process.
- Both ‘x’ and ‘m’ prefix of the devices are supported in TMI model.
- The message similar to the one shown below will be printed in the simulation output file if the simulation is through TMI model process.

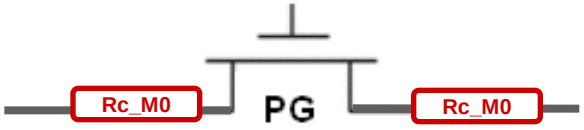
```
*********************************************************
** TMI Share Library          Version is xxxxxxxxxxxxxxx
*********************************************************
```

Usage file and flag settings

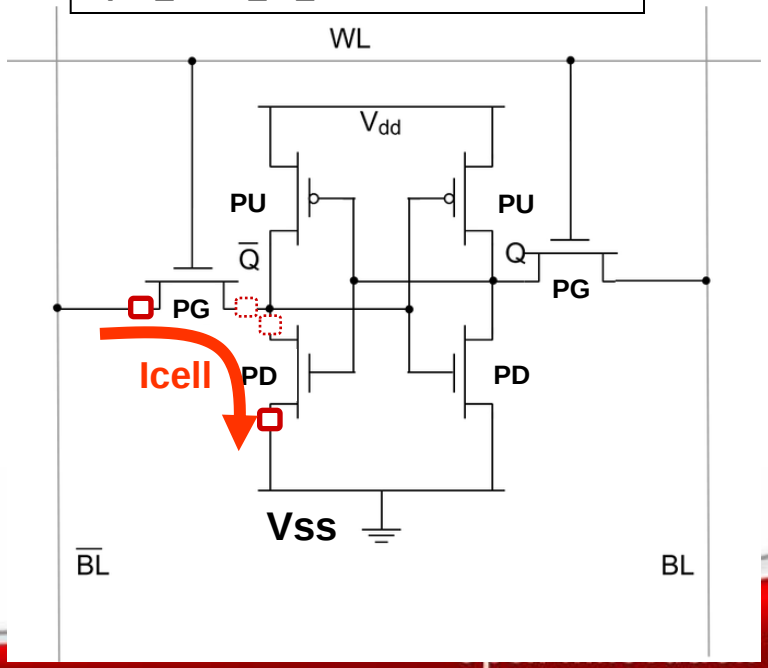
Usage File

- **There are two sets of corners in N16 SRAM usage file.**
  - Library “pre_simu_sr” have to be included for device level simulation, e.g. Idlin of PG.
  - Library “pre_simu_sr_cell” have to be included for cell level simulation, e.g. Icell of 6T SRAM.
  - Disabling the Rc_M0 of PG source site and PD drain site at cell level corners is the major difference of those two sets. By doing that, the Icell simulation at pre_simu_sr_cell stage can close to the post-simu result.
  - At post-simu stage, Rc_M0 is extracted by LPE. No difference between device corners and cell corners. (pre_simu_sr / pre_simu_sr_cell are forbidden)

1. Rc_M0 at both S/D are added at pre-simu stage of device corners by including “pre_simu_sr” lib.



2. Rc_M0 at PG source site and PD drain site are disabled at pre-simu stage of cell corners by including “pre_simu_sr_cell” lib.



Usage of Pre-layout Simulation

- The “Ccosflag”, “Ccodflag”, “Rcosflag”, “Rcodflag” and “Rgflag” are globally changed parameters for MOS device.
- The contact to poly cap, contact resistor and gate resistor are included when flags are set to 1 respectively for pre-layout simulation.
- User can include a “pre_simu_sr” or “pre_simu_sr_cell” library to set flags for pre-layout device level or cell level simulation respectively. (the pre_simu_sr and pre_simu_sr_cell lib are in usage.l)

pre_simu_sr

```
.lib pre_simu_sr
.param ccodflag_hcsr = 1
.param ccodflag_hdsr = 1
.param ccodflag_hpsr = 1
.param ccodflag_dpfsr = 1
.param ccosflag_hcsr = 1
.param ccosflag_hdsr = 1
.param ccosflag_hpsr = 1
.param ccosflag_dpfsr = 1
.param rcodflag_hcsr = 1
.param rcodflag_hdsr = 1
.param rcodflag_hpsr = 1
.param rcodflag_dpfsr = 1
.param rcosflag_hcsr = 1
.param rcosflag_hdsr = 1
.param rcosflag_hpsr = 1
.param rcosflag_dpfsr = 1
.param rgflag_hcsr = 1
.param rgflag_hdsr = 1
.param rgflag_hpsr = 1
.param rgflag_dpfsr =1
.endl pre_simu_sr
```

Netlist_device corners

```
* netlist for Idsat of DUT
*-------------------------------------------------------------
.temp 25.000
.lib 'usage.l' TT_SRAM
.lib 'usage.l' pre_simu_sr

Vgs g 0 0.850
Vss s 0 0.000
Vbs b 0 0.000

.op
.dc Vgs 0 0.850 0.850

Vds1 d1 0 0.85
xmdut1 d1 g s b nchpg_hcsr_mac nfin=2 l=0.020u

.meas idsat_temp1 find i(Vds1) when v(g)=0.85
.meas idsat1 param='-idsat_temp1*1e6'
.end
```

Please notice that the library “pre_simu_sr” (or “pre_simu_sr_cell”) need to be placed behind TT_SRAM to prevent from the redefine (flag change back to 0).

pre_simu_sr_cell

```
.lib pre_simu_sr_cell
.param ccodflag_hcsr = 1
.param ccodflag_hdsr = 1
.param ccodflag_hpsr = 1
.param ccodflag_dpfsr = 1
.param ccosflag_hcsr = 1
.param ccosflag_hdsr = 1
.param ccosflag_hpsr = 1
.param ccosflag_dpfsr = 1
.param cellflag_hcsr = 1
.param cellflag_hdsr = 1
.param cellflag_hpsr = 1
.param cellflag_dpfsr = 1
.param rcodflag_hcsr = 1
.param rcodflag_hdsr = 1
.param rcodflag_hpsr = 1
.param rcodflag_dpfsr = 1
.param rcosflag_hcsr = 1
.param rcosflag_hdsr = 1
.param rcosflag_hpsr = 1
.param rcosflag_dpfsr = 1
.param rgflag_hcsr = 1
.param rgflag_hdsr = 1
.param rgflag_hpsr = 1
.param rgflag_dpfsr = 1
.endl pre_simu_sr_cell
```

Netlist_cell corners

```
* netlist for Icell of hcsr
*-------------------------------------------------------------
.temp 25
.lib 'usage.l' TT_SRAM
.lib 'usage.l' pre_simu_sr_cell

.global vdd vss
.param pwr=0.85

XPpu1 bl_in  g1 vdd vdd pchpu_hcsr_mac nfin=1 l=0.02u
XPpu2 blb_in g2 vdd vdd pchpu_hcsr_mac nfin=1 l=0.02u
XNpd1 bl_in  g1 vss vss nchpd_hcsr_mac nfin=2 l=0.02u
XNpd2 blb_in g2 vss vss nchpd_hcsr_mac nfin=2 l=0.02u
XNpg1 bl  wl bl_in  vss nchpg_hcsr_mac nfin=2 l=0.02u
XNpg2 blb wl blb_in vss nchpg_hcsr_mac nfin=2 l=0.02u

vdd vdd 0 pwr
vss vss 0 0
vbl bl 0 pwr
vblb blb 0 pwr

vwl wl 0 pwl (0 0 0.2ns 0 0.3ns pwr)
v1 g2 bl_in 0
v2 blb_in g1 0
.nodeset v(bl_in)=0 v(blb_in)=pwr
.tran 0.05ns 0.8ns
.meas tran cell_iu find par('i(vbl)*(-1e6)') at '0.7ns'
.meas tran standby_ip find par('(i(vss)*1e12+i(vwl)*1e12)') at '0.1ns'
.end
```

N16FFC SRAM Flag Setting

- **Flag settings**
  - The usage file handles all these flags, so the users don't need to worry about them.
  - A flag in instance parameters has higher priority than a flag in model setup lib.

Global Flag in setup lib

	Flag name	Description	Default	Range	Note
SRAM	ccosflag_XXXsr	flag to enable PO to S side M0&OD outer fringing capacitance	0	1 or 0	Users can set "1" for pre-layout simulation
	ccodflag_XXXsr	flag to enable PO to D side M0&OD outer fringing capacitance	0	1 or 0	Users can set "1" for pre-layout simulation
	rcosflag_XXXsr	flag to enable the S side Rc_M0	0	1 or 0	Users can set "1" for pre-layout simulation
	rcodflag_XXXsr	flag to enable the D side Rc_M0	0	1 or 0	Users can set "1" for pre-layout simulation
	rgflag_XXXsr	flag to enable poly gate resistance	0	1 or 0	Users can set "1" for pre-layout simulation
	ngconflag_XXXsr	flag to switch gat contact number	1	1 or 2	Users can set this flag to be 1 : single-sided gate contact (default) 2 : dual-sided gate contacts
	igflag_XXXsr	flag to disable gate tunneling current components	1	1 or 0	0 : turn off gate tunneling current 1 : turn on gate tunneling current (default)
	lboff_flag_XXXsr	flag to enable lboff simulation with worst case scenario	0	1 or 0	0 : have typical case lboff simulation (default) 1 : have worst case lboff simulation
	sigma_factorn_XXXsr_npg sigma_factorn_XXXsr_npd sigma_factorp_XXXsr_ppu sigma_factorn_XXXrpsr_npg sigma_factorn_XXXrpsr_npd	flag to tighten the variation range	1	0 ~ 1	For a corner model, 2/3 is used to make a 2 sigma corner. For a statistical model, 2/3 is used to tighten the value of one sigma number to 2/3 of original number.
	mismatchflag_XXXsr	flag to enable mismatch/local variation model	1	1 or 0	Change flag globally. Will be disabled when totalflag=1
	globalflag_XXXsr	flag to enable global variation model	1	1 or 0	Change flag globally. Will be disabled when totalflag=1
	totalflag_XXXsr	flag to enable conventional corner model	1	1 or 0	Change flag globally.
	cellflag_XXXsr	flag to disable Rc_M0 or PG(source side) & PD(drain side)	0	1 or 0	"1" is for cell-level simulation. "0" is for device-level simulation.
	nicell_XXXsr	parameter to adjust worst bit Icell in the netlist	0	0 ~ 7	For worst bit Icell simulation. nicell=0 means SSg corner only. Users can set value based on their interested memory size and CDF%.

N16FFC Flag Setting

- **Flag settings**
  - Except the flag in Lib 'setup_sr' which can change the flag setting globally for all transistors when simulation, some flags are added as instance parameters for user to change the flag for specified transistors.
  - A flag in instance parameters has higher priority than a flag in model setup lib.

Flag in Instant parameters

	Flag name	Description	Default	Range	Note
SRAM	ccosflag (instance parameter only)	flag to enable PO to S side M0&OD outer fringing capacitance	0	1 or 0	
	ccodflag (instance parameter only)	flag to enable PO to D side M0&OD outer fringing capacitance	0	1 or 0	
	rcosflag (instance parameter only)	flag to enable the S side M0 resistance	0	1 or 0	
	rcodflag (instance parameter only)	flag to enable the D side M0 resistance	0	1 or 0	
	rgflag (instance parameter only)	flag to enable poly gate resistance	0	1 or 0	
	mismatchflag (instance parameter only)	flag to enable mismatch/local variation model	1	1 or 0	Change flag locally,should be specified for each transistor
	globalflag (instance parameter only)	flag to enable global variation model	1	1 or 0	Change flag locally,should be specified for each transistor
	totalflag (instance parameter only)	flag to enable conventional corner model	1	1 or 0	Change flag locally,should be specified for each transistor
	nicell (instance parameter only)	parameter to adjust worst bit Icell in the netlist	0	0 ~ 7	For worst bit Icell simulation (all SRAM cells). nicell=0 means SSg corner only. Users can set value based on their interested memory size and CDF%.
	wbifactor (instance parameter only)	parameter to active worst bit Icell in the netlist for dual port (DP) SRAM cell only	0	1 or 0	For dual port (DP) SRAM cell worst bit Icell simulation.(See page-21)

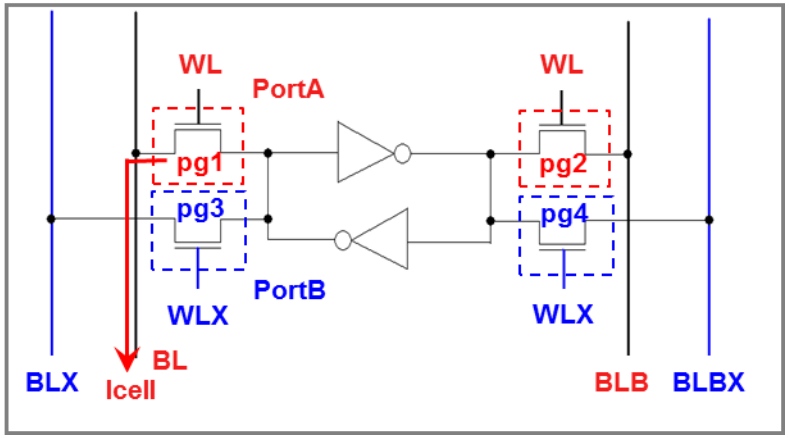
Usage of variation model

Common Variation Simulation Options (1)

- **Traditional total corner approach**
  - Assumes all transistors in the chip are moving toward the same direction, which is not realistic and may be too conservative.
  - In some circuits, this approach is too aggressive because it assumes no mismatch between transistors.
- **Global corner only (or with local Monte Carlo)**
  - As fast as traditional total corner simulation, but is more realistic.
  - Not suitable for circuits with very small number of transistors and mismatch-sensitive circuits (analog).
- **Local Monte Carlo Only**
  - For mismatch simulation.
- **Full Monte Carlo simulation (Global Monte Carlo + Local Monte Carlo)**
  - The same global variation applies to all devices, while each device gets random local variation.
  - Too slow.

Common Variation Simulation Options (2)

- **Worst bit Icell Simulation approach**
  - Defined to simulate the worst bit Icell current.
  - It is done by SSg corner simulation with a given "nicell" value. Users can set value based on their interested memory size and CDF%.
  - Worst bit Icell simulation for dual port (DP) SRAM cell has to add extra instance parameter "wbifactor = 1" at PortB PG devices if Icell is extracted by PortA PG devices and vice versa. (See page-21 for sample netlist)
  - Worst bit Icell simulation is only available for Vdd ~ 0.8*Vdd and 125C ~ -40C.

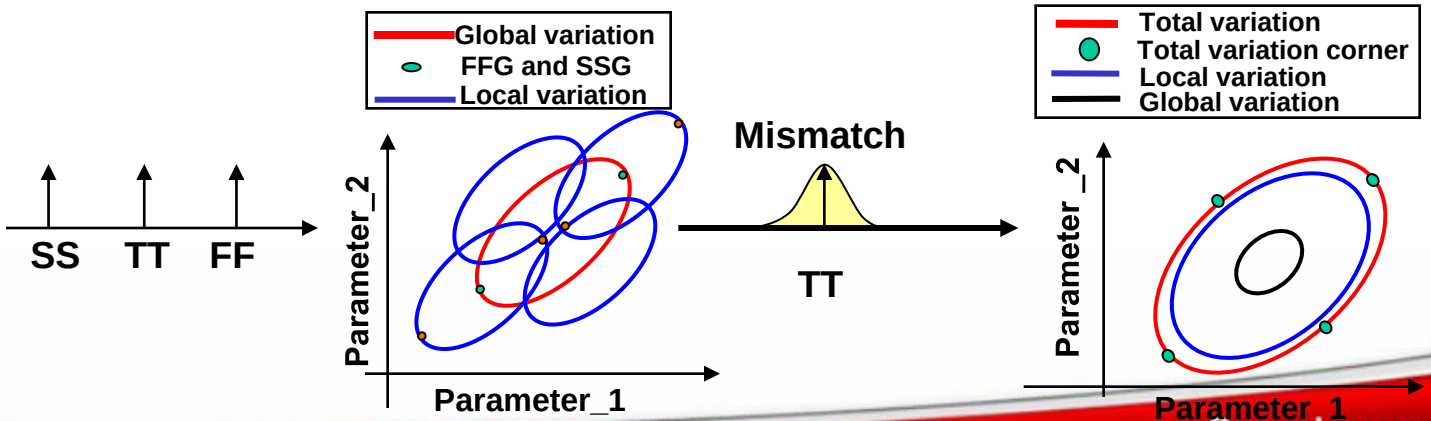


Dual port SRAM schematic circuit

Model Usage File

- **Pre-defined libs prepared in usage file**
  - Collect the must-used libs to prevent from misuse
    - 5 simulation cases are included in the usage file.
  - Add one more lib, .lib pre_simu_sr for device level and pre_simu_sr_cell for cell level, to address parameter setting (ex. ccoflag, cellflag).

Case	Global	Local	Lib to use in netlist	Additional Command need in netlist
Case1: Conventional Corner	Corner		TT_SRAM SS_SRAM FF_SRAM SF_SRAM FS_SRAM	N/A
Case2: Global Corner + Local MC	Corner	Monte Carlo	TTGlobalCorner_LocalMC_SRAM SSGlobalCorner_LocalMC_SRAM FFGlobalCorner_LocalMC_SRAM SFGlobalCorner_LocalMC_SRAM FSGlobalCorner_LocalMC_SRAM	Sweep Monte=1000 (optional) If this command has not been assigned, the result becomes global corner only.
Case3: Local MC Only	N/A	Monte Carlo	LocalMCOnly_SRAM	Sweep Monte=1000
Case4: global MC + LocalMC	Monte Carlo	Monte Carlo	GlobalMC_LocalMC_SRAM	Sweep Monte=1000
Case5: Worst bit Icell Corner	Corner		SSGlobalCorner_LocalMC_SRAM	.param nicell_XXX=YYY (YYY is the input value by users)



Netlist for usage of variation model

Model Usage File: Case1

- **Case1: Conventional fixed corner model**
  - It is identical to the existing TSMC fixed corner

```
.option  nomod ingold=2
+ newtol numdgt=7 relmos=1e-4 absmos=1e-8 relv=1e-4 relvdc=1e-4
.option MODSRH=0
*---------------------------------------------------------
.temp 25.000
.option brief=1
.lib 'usage.l' TT_SRAM
.option brief=0


Vgs g 0 0.850
Vss s 0 0.000
Vbs b 0 0.000

Vds0 d0 0 0.850
mdut0 d0 g s b nchpg_hcsr_mac  nfin=2  l=0.02u

.op
.dc vgs 0 0.850 0.850
.print
+ id_dut0_=par('abs(i(Vds0))')


.end
```

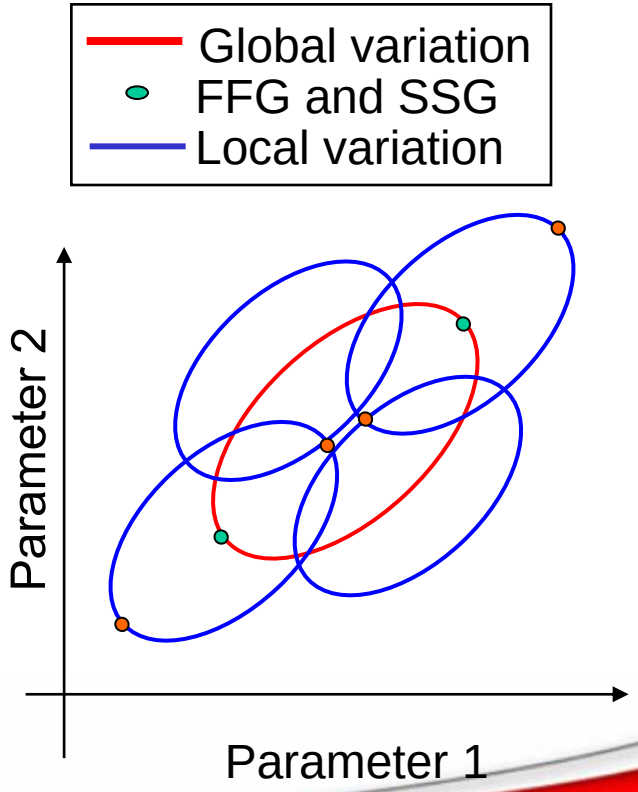
Set the devices' Fin number and Ldrawn

Model Usage File: Case2

- **Case2: Global variation only corner + local variation Monte Carlo**
  - Users can run mismatch model on top of each global variation only corner model.
  - Perform functional check on corner condition and understand design margin.
  - This methodology can help to check the contribution from local and global respectively.
  - Apply on mismatch/local variation sensitivity circuits: SRAM/Diff Amp/VCO/setup time/hold time characterization, … etc.

```
.option  nomod ingold=2
+ newtol numdgt=7 relmos=1e-4 absmos=1e-8 relv=1e-4 relvdc=1e-4
.option MODSRH=0
*---------------------------------------------------------
.temp 25.000
.option brief=1
.lib 'usage.l ' TTGlobalCorner_LocalMC_SRAM
.option brief=0

Vgs g 0 0.850
Vss s 0 0.000
Vbs b 0 0.000
Vds0 d0 0 0.850
xmdut0 d0 g s b nchpg_hcsr_mac  nfin=2 l=0.02u
.op
.dc vgs 0 0.850 0.850 sweep monte=1
.print
+ id_dut0_=par('abs(i(Vds0))')

.end
```

Set the devices' Fin number and Ldrawn



Model Usage File: Case3

- **Case3: Local variation Monte Carlo**
  - The same purpose as conventional mismatch model.

```
.option  nomod ingold=2
+ newtol numdgt=7 relmos=1e-4 absmos=1e-8 relv=1e-4 relvdc=1e-4
.option MODSRH=0
*---------------------------------------------------------
.temp 25.000
.option brief=1
.lib 'usage.l' LocalMCOnly_SRAM
.option brief=0

Vgs g 0 0.850
Vss s 0 0.000
Vbs b 0 0.000

Vds0 d0 0 0.850
xmdut0 d0 g s b nchpg_hcsr_mac nfin=2 l=0.02u
.op
.dc vgs 0 0.850 0.850 sweep monte=1
.print
+ id_dut0_=par('abs(i(Vds0))')
.end
```

Set the devices' Fin number and Ldrawn

Model Usage File: Case4

- **Case4: Global variation Monte Carlo + Local variation Monte Carlo**
  - Global variation: all trans share one set of random numbers for each MC run.
  - Local variation: Each trans applies different set of random numbers for each MC run.
  - The result is a distribution function which can predict the silicon distribution accurately.
  - Can meet the rule listed on RHS completely.

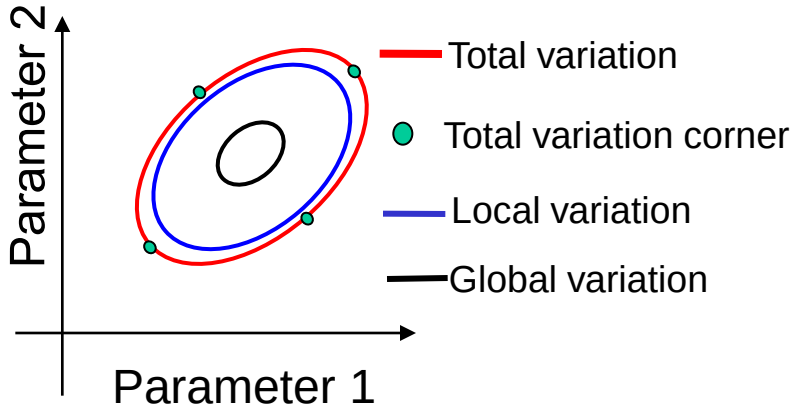
```
.option  nomod ingold=2
+ newtol numdgt=7 relmos=1e-4 absmos=1e-8 relv=1e-4 relvdc=1e-4
.option MODSRH=0
*---------------------------------------------------------
.temp 25.000
.option brief=1
.lib 'usage.l' GlobalMC_LocalMC_SRAM
.option brief=0

Vgs g 0 0.850
Vss s 0 0.000
Vbs b 0 0.000
Vds0 d0 0 0.850
xmdut0 d0 g s b nchpg_hcsr_mac  nfin=2 l=0.02u
```

Set the devices' Fin number and Ldrawn

```
.op
.dc vgs 0 0.850 0.850 sweep monte=1
.print
+ id_dut0_=par('abs(i(Vds0))')
.end
```

$$\sigma_{total}^{2} = \sigma_{global}^{2} + \sigma_{local}^{2}$$

global MC + local MC



Model Usage File: Case5-1

- **Case5-1: Worst bit Icell simulation by global parameter setting:**

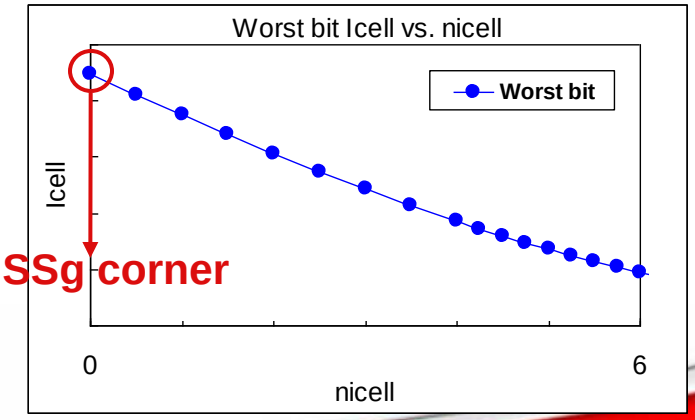
```
.option gmindc=1.0E-18 gmin=1.0E-18
.prot
.lib 'usage.l' SSGlobalCorner_LocalMC_SRAM
.lib 'usage.l' pre_simu_sr_cell
.unprot
.param nicell_hcsr=6
.global vdd vss
.param pwr=0.765
.temp -30
.param LPU=20N
.param LPD=20N
.param LPG=20N
```

Set the global parameter "nicell"

Set the devices' Ldrawn

```
XPpu1 bl_in g1 vdd vdd pchpu_hcsr_mac   nfin=1 l=LPU
XPpu2 blb_in g2 vdd vdd pchpu_hcsr_mac  nfin=1 l=LPU
XNpd1 bl_in g1 vss vss nchpd_hcsr_mac   nfin=2 l=LPD
XNpd2 blb_in g2 vss vss nchpd_hcsr_mac  nfin=2 l=LPD
XNpg1 bl  wl  bl_in  vss nchpg_hcsr_mac  nfin=2 l=LPG
XNpg2 blb wl blb_in vss nchpg_hcsr_mac  nfin=2 l=LPG

vdd vdd 0 pwr
vss vss 0 0
vbl bl 0 pwr
vblb blb 0 pwr

vwl wl 0 pwl (0 0 0.2ns 0 0.3ns pwr)
v1 g2 bl_in 0
v2 blb_in g1 0
.nodeset v(bl_in)=0 v(blb_in)=pwr
.tran 0.05ns 0.8ns
.meas tran cell_iu find par('i(vbl)*(-1e6)') at '0.7ns'
.end
```

Set the devices' Fin number and Ldrawn



Model Usage File: Case5-2

- **Case5-2: Worst bit Icell simulation by instance parameter setting:**

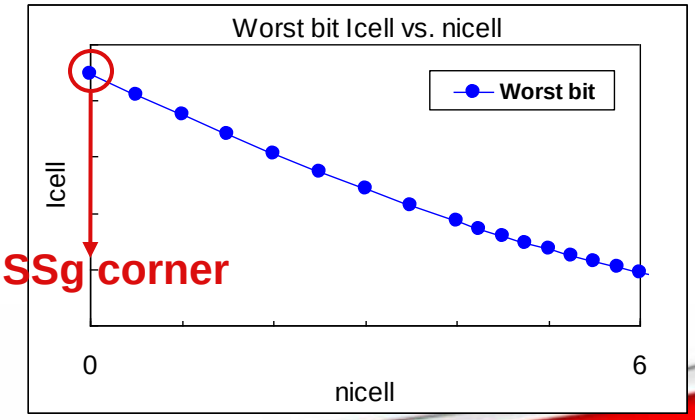
```
.option gmindc=1.0E-18 gmin=1.0E-18
.prot
.lib 'usage.l' SSGlobalCorner_LocalMC_SRAM
.lib 'usage.l' pre_simu_sr_cell
.unprot

.global vdd vss
.param pwr=0.765
.temp -30
.param LPU=20N
.param LPD=20N
.param LPG=20N
```

Set the devices' Ldrawn

Set the devices' Fin number and Ldrawn

```
XPpu1 bl_in g1 vdd vdd pchpu_hcsr_mac  nfin=1 l=LPU nicell=6
XPpu2 blb_in g2 vdd vdd pchpu_hcsr_mac nfin=1 l=LPU nicell=6
XNpd1 bl_in g1 vss vss nchpd_hcsr_mac  nfin=2 l=LPD nicell=6
XNpd2 blb_in g2 vss vss nchpd_hcsr_mac nfin=2 l=LPD nicell=6
XNpg1 bl  wl  bl_in  vss nchpg_hcsr_mac  nfin=2 l=LPG nicell=6
XNpg2 blb wl blb_in vss nchpg_hcsr_mac nfin=2 l=LPG nicell=6

vdd vdd 0 pwr
vss vss 0 0
vbl bl 0 pwr
vblb blb 0 pwr

vwl wl 0 pwl (0 0 0.2ns 0 0.3ns pwr)
v1 g2 bl_in 0
v2 blb_in g1 0
.nodeset v(bl_in)=0 v(blb_in)=pwr
.tran 0.05ns 0.8ns
.meas tran cell_iu find par('i(vbl)*(-1e6)') at '0.7ns'
.end
```

Set the instance parameter "nicell"



Model Usage File: Case5-3

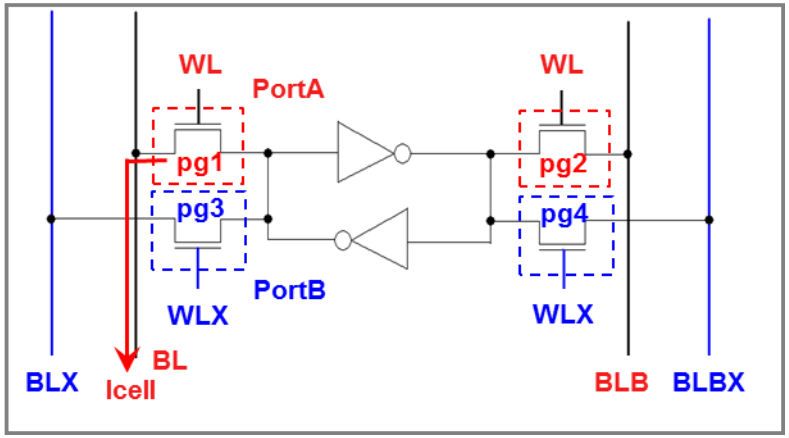
- **Case5-3: Worst bit Icell simulation for dual port SRAM cell (DP only)**
  - **It has to add extra instance parameter "wbifactor=1" at PortB PG devices if Icell is extracted by PortA PG devices and vice versa.**
  - **The usage of "nicell" is same as "case5-1 or case5-2".**

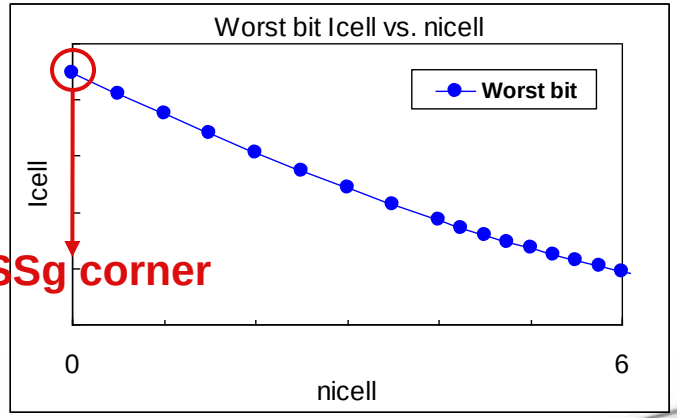
```
.option gmindc=1.0E-18 gmin=1.0E-18
.prot
.lib 'usage.l' SSGlobalCorner_LocalMC_SRAM
.lib 'usage.l' pre_simu_sr_cell
.unprot
.param nicell_dpfsr=6
.global vdd vss
.param pwr=0.765
.temp -30
xppu1 bl_in g1 vdd vdd pchpu_dpfsr_mac     nfin=1   l=0.02u
xPpu2 blb_in g2 vdd vdd pchpu_dpfsr_mac    nfin=1   l=0.02u
xNpd1 bl_in g1 vss vss nchpd_dpfsr_mac     nfin=4   l=0.02u
xNpd2 blb_in g2 vss vss nchpd_dpfsr_mac    nfin=4   l=0.02u
xNpg1 bl wl bl_in vss nchpg_dpfsr_mac      nfin=2   l=0.02u
xNpg2 blb wl blb_in vss nchpg_dpfsr_mac    nfin=2   l=0.02u
xNpg3 blx wlx bl_in vss nchpg_dpfsr_mac    nfin=2   l=0.02u  wbifactor=1
xNpg4 blbx wlx blb_in vss nchpg_dpfsr_mac  nfin=2   l=0.02u  wbifactor=1
vdd vdd 0 pwr
vss vss 0 0
vbl bl 0 pwr
vblb blb 0 pwr
vblx blx 0 pwr
vblbx blbx 0 pwr
vwl wl 0 pwl (0 0 0.2ns 0 0.3ns pwr)
vwlx wlx 0 pwl (0 0 0.2ns 0 0.3ns pwr)
v1 g2 bl_in 0
v2 blb_in g1 0
.nodeset v(bl_in)=0 v(blb_in)=pwr
.tran 0.05ns 0.8ns
.meas tran cell_iu find par('i(vbl)*(-1e6)') at '0.7ns'
.end
```

PortA

PortB

Extract Icell from PortA PG





Iboff Worst Case Model Usage

- **The “iboff_flag” is a globally changed parameter.**
- **The worst case Iboff scenario will be activated if user sets iboff_flag=1**
- **The “iboff_flag” is included in all total corners.**

```
* netlist for Idsat of DUT
*---------------------------------------------------------
.temp 25
.lib ‘usage.l' tt_sram
.param iboff_flag_hcsr=1

Vgs g 0 0.85
Vss s 0 0
Vbs b 0 0

Vds0 d0 0 0.85
xmdut0 d0 g s b nchpg_hcsr_mac nfin=2 l=0.02u
.op
.dc vgs 0 0.85 0.85
.print
+ id_dut0_=par('abs(i(Vds0))')
.end
```