

INVERT DEMO

S. de Graaf

Circuits and Systems Group
Faculty of Electrical Engineering,
Mathematics and Computer Science
Delft University of Technology
The Netherlands

Report EWI-ENS 06-01
May 11, 2006

Copyright © 2006 by the author.

Last revision: May 15, 2006.

1. INTRODUCTION

We got an e-mail from OptEM about this demo (see appendices), that the SPICE netlists were incorrect (see appendix I).

There was a problem with Netlist #1, because the diodes are wrong connected. I fixed this problem by changing the pin order in the technology file. The first pin of the junction caps is taken as the 'p' (anode) pin. To let it work, there must also be some changes to the *space* code. Because ground caps are swapped, when the first node is ground, the sortNr pointing to a 'p' node must also be changed into a 'n' node. Second, for substrate to ground caps, the 'p' node may not always be connected to substrate.

There was also a problem with Netlist #2, because the transistor parameters were not printed. This happened, because the nodes were swapped. The first node of these d/s-caps is always connected to ground and the second node always to the d/s node. The *xspice* program is searching for the 'n' node. When the 'n' node is ground, the "area" and "perim" attribute values are stored in the wrong net and can't be resolved afterwards. That's the reason, why the transistor parameters were missing. This is also repaired by changing the sortNr in the *space* code.

Note that the d/s-caps are modeled to be connected to ground. This is not the real case. This ground can be p-substrate or n-well. These d/s-caps cannot be used to extract netlist diodes, because no choice between 'p' and 'n' can be made. This method is only intended to get the transistor parameters.

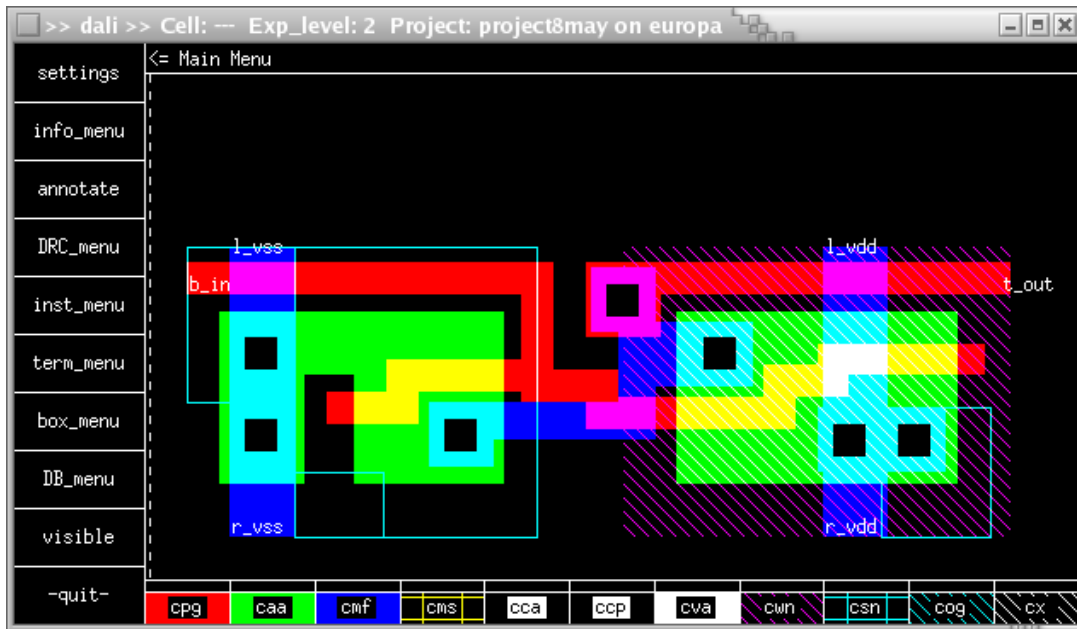
For pictures of the "invert" demo, see section 2. For more details about how *xspice* retrieves the transistor parameters, see section 3.

The following appendices are included:

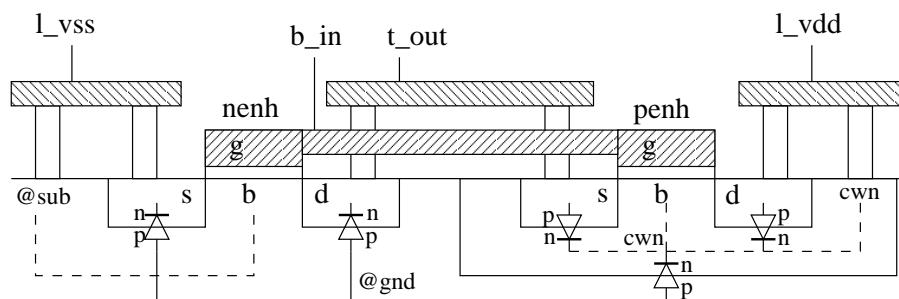
- Appendix A -- Demo Invert Files
- Appendix B -- Demo Invert README File
- Appendix C -- Demo Invert xspicerc File
- Appendix D -- Demo Invert jun.lib File
- Appendix E -- Demo Invert invert.cmd File
- Appendix F -- Demo Invert Results Step 3/4
- Appendix G -- Demo Invert Results Step 5/6
- Appendix H -- Demo Invert elem.s File
- Appendix I -- Email From OptEM
- Appendix J -- Changed Space Code
- Appendix K -- Technology File Changes

2. INVERT PICTURES

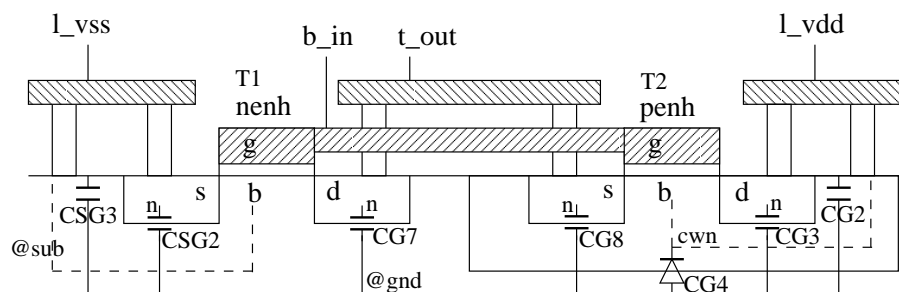
Picture of the layout of cell "invert" (90 degrees rotated):



Picture with junction diodes (see result of step 4, Netlist #1):



Picture with junction d/s caps (see result of step 6, Netlist #2):



3. HOW DO WE GET THE MOS PARAMETERS

I have looked in the *xspice* (*xsls*) source code and know now how it works. Function *prInst* in file "prInst.c" reads the circuit "mc" stream twice. The first time, it tries to find "ap" cell calls with function *is_ap*, and is also searching for transistor cell calls. The "area" and "perim" values are stored in a table with function *addAP*. Because *k=7*, a search with function *findNet* is done for net_name "n" and inst_name "_CGxx" (or "_CSGxx"). Function *findNet* return the net number *fN_nx* of this node. Net_name "n" must be connected to a d/s-node of a transistor. Function *addAP* stores this information under the net number in a data structure. This data structure contains the torname and the area and perim values. Note that the gate-width of a transistor is also stored in this data structure under the net number of the "s" and "d" node. Note that, when more drains or sources are with the same torname in the same net number the transistor count member is updated. Flag *is_caps* is set, if there are any "ap" cell calls found in the "mc" stream. This is almost always true, because there are almost always transistor devices.

The second time, when the "mc" stream is read, the "ap" cell calls (the special ground caps) are skipped. The flag *is_ap_mos* is set, when a transistor call is detected. The net numbers for the transistor "d" and "s" node are set on variables *dN_nx* and *sN_nx*. For *is_ap_mos*, function *prAP* tries to add the "ap" parameters to the transistor. The net numbers are used to find the area and perim and gate-width values in the net data structures. When there are no d/s-caps, zero values for "area" and "perim" are returned. In that case, no d/s parameters are added to the transistors. But this can also happen, when the d/s-caps values are stored under the wrong net number.

Conclusion, when a wrong net number is used, the correct values can not be found. When the net_names "n" and "p" are swapped by one of the special ground caps there is a problem. Thus, the technology compiler *tecc* must always use the correct pin order for writing these d/s-caps to the technology file. The first pin ("p") must be the ground node and the second pin ("n") must be the d/s-node.

Thus, the *space* extractor may also not change the order of the pins of these special d/s-caps. Also for the junction diodes it is important to maintain the specified pin order, else the anode and cathode are wrong connected. Maybe we can better look to the conductor type of the pins.

However, by implementing the BEM/FEM method, i have decided to use a certain order for capacitance element pins. Thus, when a ground or substrate pin is used, this pin is always the second pin. Thus, the pins are possibly swapped. For the d/s capacitors, we are sure that this happens. Thus, in that case it is also important to swap the *sortNr*. Because this *sortNr* points to an entry in the *capPolarityTab* table and tells which polarity must be used for the first pin (normally 'p' of coarse).

After fixing this swap problem, i detected that it still was not working in all cases. The caps between substrate and ground get always a fixed polarity order. That's wrong, because *space* may not force any order of the pins.

4. APPENDICES

APPENDIX A -- Demo Invert Files

```
europa(Linux) /u/01/01/simon/out/cacd/share/demo/invert $ ls -lt
-rw-r--r--  1 simon    space          409 May  2  2003 xspicerc
-rw-r--r--  1 simon    space          224 May  2  2003 jun.lib
-rw-r--r--  1 simon    space          464 May  2  2003 invert.cmd
-rw-r--r--  1 simon    space        2638 May  2  2003 README
-rw-r--r--  1 simon    space        4096 Dec 19  2002 invert.gds
```

APPENDIX B -- Demo Invert README File

Demonstration of extraction of an inverter with transistor bulk connections and drain/source regions. The latter are extracted as either non-linear junction capacitances, or drain/source area and perimeter information attached to the MOS transistors.

STEPS
=====

1. Make project, use scmos_n technology and lambda = 0.2 (micron)
% mkpr -p scmos_n -l 0.2 projectname

Rest assumes that you and your files are in the project directory.

2. Use cgi to put invert layout into database.
% cgi invert.gds
3. Perform an extraction of the circuit, including coupling capacitances. The drain/source regions are extracted as non-linear junction capacitances that have parameters 'area' and 'perimeter'.
% space -C -Sjun_caps=area-perimeter invert
4. Look in the file xspicerc to see how the junction capacitances are printed in the netlist output as diodes with parameters 'area' and 'pj', and next get the SPICE circuit description.
% xspice -a invert
5. An alternative is to extract the drain/source regions as 'area and perimeter' information that is attached to the MOS transistors. This achieved by modifying the transistor definitions in the element definition file. Therefore, first copy the element definition file from the process directory to the local file elem.s:
% cp \$ICDPATH/share/lib/process/scmos_n/space.def.s elem.s
% chmod +w elem.s

Change the lines

```
nenh : cpg caa  csn : cpg caa : @sub  # nenh MOS
penh : cpg caa !csn : cpg caa : cwn   # penh MOS
```

into

```
nenh : cpg caa  csn : cpg caa (!cpg caa cs) : @sub # nenh MOS
penh : cpg caa !csn : cpg caa (!cpg caa !cs) : cwn  # penh MOS
```

to describe the drain/source regions.

Change the lines

```
cond_pa : caa !cpg !csn : caa : 70 : p    # p+ active area
cond_na : caa !cpg  csn : caa : 50 : n    # n+ active area
into
cond_pa : caa !cpg !csn : caa : 0  : p    # p+ active area
cond_na : caa !cpg  csn : caa : 0  : n    # n+ active area
to avoid complaints by tecc about the fact the resistance for drain/
source regions should be zero (that resistance is already modeled in
the transistor model).
```

And remove the 'ndif' and 'pdif' capacitance lists.

Next, run tecc to compile the new element definition file:

```
% tecc elem.s
```

and run space again with the new element definition file.

```
% space -C -Sjun_caps=area-perimeter -E elem.t invert
```

6. Next watch the SPICE output again.

```
% xspice -a invert
```

APPENDIX C -- Demo Invert xspicerc File

```
include_library spice3f3.lib
include_library jun.lib

model nenh_0 nenh nmos ()
model penh_0 penh pmos ()
model ndif ndif d ()
model nwell nwell d ()
model pdif pdif d ()

bulk nmos 0.0
bulk pmos 5.0

params ndif { area=$area pj=$perim }
params nwell { area=$area pj=$perim }
params pdif { area=$area pj=$perim }
```

APPENDIX D -- Demo Invert jun.lib File

```
model ndif d (is=2u cjo=100u vj=0.8 m=0.5)
model nwell d (is=2u cjo=100u vj=0.8 m=0.5)
model pdif d (is=10u cjo=500u vj=0.8 m=0.5)
```

APPENDIX E -- Demo Invert invert.cmd File

```
/*
/* Command file for simulating the inverter circuit using simeye/nspice.
*/

option sigunit = 1.000000e-11
option simperiod = 60
set b_in = 1*10 h*30 l*20
set l_vdd = h*20
set l_vss = l*20
option level = 3
plot b_in t_out

/* spice commands:

*%
trise 0.05n
tfall 0.05n
tstep 0.005n
vlow 0
vhigh 5
*%
*%

*/
```

APPENDIX F -- Demo Invert Results Step 3/4

```

% space -C -Sjun_caps=area-perimeter invert
% xspice -au invert
invert

* Generated by: xspice 2.39 25-Jan-2006
* Date: 11-May-06 10:12:18 GMT
* Path: /u/52/52/work/simon/CACD/demo/demo/invert/project8may
* Language: SPICE

* circuit invert b_in t_out l_vdd r_vdd l_vss r_vss
vnet1 l_vdd r_vdd 0
vnet2 l_vss r_vss 0
m1 t_out b_in l_vss l_vss nenh_0 w=6.8u l=1.2u
m2 l_vdd b_in t_out l_vdd penh_0 w=12.4u l=1.2u
c1 l_vdd b_in 795.12e-18
c2 l_vdd t_out 3.04892f
d1 t_out l_vdd pdif area=20.48p pj=14.8u
c3 l_vdd GND 249.6e-18
d2 GND l_vdd nwell area=155.52p pj=50.4u
c4 b_in t_out 495.12e-18
c5 b_in GND 3.21264f
c6 b_in l_vss 282.72e-18
c7 t_out GND 987.68e-18
d3 GND t_out ndif area=16.16p pj=11.2u
c8 l_vss GND 774.4e-18
d4 GND l_vss ndif area=27.92p pj=22.6u
* end invert

+ ucrit=10k delta=200m xj=500n vmax=50k neff=1 rsh=0 nfs=0 js=2u cj=100u
+ cjsw=300p mj=500m mjsw=300m pb=800m cgdo=300p cgso=300p)
+ ucrit=10k delta=200m xj=500n vmax=50k neff=1 rsh=0 nfs=0 js=10u cj=500u
+ cjsw=600p mj=500m mjsw=300m pb=800m cgdo=300p cgso=300p)

% dbcat -cs mc invert
=> circuit/invert/mc
cell:"nenh" inst:"_T1" attr:"w=6.8000e-06;l=1.2000e-06" dim:0
cell:"penh" inst:"_T2" attr:"w=1.2400e-05;l=1.2000e-06" dim:0
cell:"cap" inst:"_C1" attr:"v=7.951200e-16" dim:0
cell:"cap" inst:"_C2" attr:"v=3.048920e-15" dim:0
cell:"pdif" inst:"_C3" attr:"area=2.048e-11;perim=1.48e-05" dim:0
cell:"cap" inst:"_CG1" attr:"v=2.496000e-16" dim:0
cell:"nwell" inst:"_CG2" attr:"area=1.5552e-10;perim=5.04e-05" dim:0
cell:"cap" inst:"_C4" attr:"v=4.951200e-16" dim:0
cell:"cap" inst:"_CG3" attr:"v=3.212640e-15" dim:0
cell:"cap" inst:"_CS1" attr:"v=2.827200e-16" dim:0
cell:"cap" inst:"_CG4" attr:"v=9.876800e-16" dim:0
cell:"ndif" inst:"_CG5" attr:"area=1.616e-11;perim=1.12e-05" dim:0
cell:"cap" inst:"_CSG1" attr:"v=7.744000e-16" dim:0
cell:"ndif" inst:"_CSG2" attr:"area=2.792e-11;perim=2.26e-05" dim:0

```


APPENDIX G -- Demo Invert Results Step 5/6

```
% space -C -Sjun_caps=area-perimeter -E elem.t invert
% xspice -au invert
invert

* Generated by: xspice 2.39 25-Jan-2006
* Date: 11-May-06 10:15:27 GMT
* Path: /u/52/52/work/simon/CACD/demo/demo/invert/project8may
* Language: SPICE

* circuit invert b_in t_out l_vdd r_vdd l_vss r_vss
vnet1 l_vdd r_vdd 0
vnet2 l_vss r_vss 0
m1 t_out b_in l_vss l_vss nenh_0 w=6.8u l=1.2u ad=16.16p as=27.92p pd=11.2u
+ ps=22.6u nrs=0.603806 nrd=0.349481
m2 l_vdd b_in t_out l_vdd penh_0 w=12.4u l=1.2u ad=23.36p as=20.48p pd=10.8u
+ ps=14.8u nrs=0.133195 nrd=0.151925
c1 l_vdd b_in 795.12e-18
c2 l_vdd t_out 3.04892f
c3 l_vdd GND 249.6e-18
d1 GND l_vdd nwell area=155.52p pj=50.4u
c4 b_in t_out 495.12e-18
c5 b_in GND 3.21264f
c6 b_in l_vss 282.72e-18
c7 t_out GND 987.68e-18
c8 l_vss GND 774.4e-18
* end invert

+ ucrit=10k delta=200m xj=500n vmax=50k neff=1 rsh=0 nfs=0 js=2u cj=100u
+ cjsw=300p mj=500m mjsw=300m pb=800m cgdo=300p cgso=300p)
+ ucrit=10k delta=200m xj=500n vmax=50k neff=1 rsh=0 nfs=0 js=10u cj=500u
+ cjsw=600p mj=500m mjsw=300m pb=800m cgdo=300p cgso=300p)

% dbcat -cs mc invert
=> circuit/invert/mc
cell:"nenh" inst:"_T1" attr:"w=6.8000e-06;l=1.2000e-06" dim:0
cell:"penh" inst:"_T2" attr:"w=1.2400e-05;l=1.2000e-06" dim:0
cell:"cap" inst:"_C1" attr:"v=7.951200e-16" dim:0
cell:"cap" inst:"_C2" attr:"v=3.048920e-15" dim:0
cell:"cap" inst:"_CG1" attr:"v=2.496000e-16" dim:0
cell:"$nenh$ds" inst:"_CG2" attr:"area=7.84e-12;perim=5.6e-06" dim:0
cell:"$penh$ds" inst:"_CG3" attr:"area=2.336e-11;perim=1.08e-05" dim:0
cell:"nwell" inst:"_CG4" attr:"area=1.5552e-10;perim=5.04e-05" dim:0
cell:"cap" inst:"_C3" attr:"v=4.951200e-16" dim:0
cell:"cap" inst:"_CG5" attr:"v=3.212640e-15" dim:0
cell:"cap" inst:"_CS1" attr:"v=2.827200e-16" dim:0
cell:"cap" inst:"_CG6" attr:"v=9.876800e-16" dim:0
cell:"$nenh$ds" inst:"_CG7" attr:"area=1.616e-11;perim=1.12e-05" dim:0
cell:"$penh$ds" inst:"_CG8" attr:"area=2.048e-11;perim=1.48e-05" dim:0
cell:"cap" inst:"_CSG1" attr:"v=7.744000e-16" dim:0
cell:"$nenh$ds" inst:"_CSG2" attr:"area=2.792e-11;perim=2.26e-05" dim:0
cell:"$penh$ds" inst:"_CSG3" attr:"area=8.4e-12;perim=5.8e-06" dim:0
% dbcat -cs net invert
=> circuit/invert/net
net:"l_vdd" inst:"" attr:"cd=1" subnets:9
  subnet[0]: "r_vdd" inst:"" attr:""
  subnet[1]: "p" inst:"_C1" attr:""
```

```

    subnet[2]: "p" inst: "_C2" attr: ""
    subnet[3]: "b" inst: "_T2" attr: ""
    subnet[4]: "d" inst: "_T2" attr: ""
    subnet[5]: "p" inst: "_CG1" attr: ""
    subnet[6]: "n" inst: "_CG2" attr: ""
    subnet[7]: "n" inst: "_CG3" attr: ""
    subnet[8]: "n" inst: "_CG4" attr: ""
net: "b_in" inst: "" attr: "cd=2" subnets: 6
    subnet[0]: "p" inst: "_C3" attr: ""
    subnet[1]: "n" inst: "_C1" attr: ""
    subnet[2]: "g" inst: "_T2" attr: ""
    subnet[3]: "g" inst: "_T1" attr: ""
    subnet[4]: "p" inst: "_CG5" attr: ""
    subnet[5]: "p" inst: "_CS1" attr: ""
net: "t_out" inst: "" attr: "cd=3" subnets: 7
    subnet[0]: "n" inst: "_C3" attr: ""
    subnet[1]: "n" inst: "_C2" attr: ""
    subnet[2]: "d" inst: "_T1" attr: ""
    subnet[3]: "s" inst: "_T2" attr: ""
    subnet[4]: "p" inst: "_CG6" attr: ""
    subnet[5]: "n" inst: "_CG7" attr: ""
    subnet[6]: "n" inst: "_CG8" attr: ""
net: "SUBSTR" inst: "" attr: "cd=0" subnets: 8
    subnet[0]: "l_vss" inst: "" attr: ""
    subnet[1]: "r_vss" inst: "" attr: ""
    subnet[2]: "s" inst: "_T1" attr: ""
    subnet[3]: "b" inst: "_T1" attr: ""
    subnet[4]: "n" inst: "_CS1" attr: ""
    subnet[5]: "p" inst: "_CSG1" attr: ""
    subnet[6]: "n" inst: "_CSG2" attr: ""
    subnet[7]: "n" inst: "_CSG3" attr: ""
net: "GND" inst: "" attr: "cd=4" subnets: 11
    subnet[0]: "n" inst: "_CG1" attr: ""
    subnet[1]: "p" inst: "_CG2" attr: ""
    subnet[2]: "p" inst: "_CG3" attr: ""
    subnet[3]: "p" inst: "_CG4" attr: ""
    subnet[4]: "n" inst: "_CG5" attr: ""
    subnet[5]: "n" inst: "_CG6" attr: ""
    subnet[6]: "p" inst: "_CG7" attr: ""
    subnet[7]: "p" inst: "_CG8" attr: ""
    subnet[8]: "n" inst: "_CSG1" attr: ""
    subnet[9]: "p" inst: "_CSG2" attr: ""
    subnet[10]: "p" inst: "_CSG3" attr: ""

```

APPENDIX H -- Demo Invert elem.s File

```

# space element definition file for scmos_n example process
# with transistor bulk connections and substrate terminals
# for substrate contacts and nmos bulk connections, and
# with information for 3D capacitance extraction and
# substrate resistance extraction.
#
# masks:
# cpg - polysilicon interconnect      ccp - contact metal to poly
# caa - active area                  cva - contact metal to metal2
# cmf - metal interconnect           cwn - n-well
# cms - metal2 interconnect          csn - n-channel implant
# cca - contact metal to diffusion    cog - contact to bondpads

unit resistance      1      # ohm
unit c_resistance    1e-12 # ohm um^2
unit a_capacitance   1e-6  # aF/um^2
unit e_capacitance   1e-12 # aF/um
unit capacitance     1e-15 # fF

maxkeys 13

colors :
  cpg  red
  caa  green
  cmf  blue
  cms  gold
  cca  black
  ccp  black
  cva  black
  cwn  glass
  csn  glass
  cog  glass
  @sub pink

conductors :
  # name      : condition      : mask : resistivity : type
  cond_mf : cmf                : cmf  : 0.045       : m   # first metal
  cond_ms : cms                : cms  : 0.030       : m   # second metal
  cond_pg : cpg                : cpg  : 40          : m   # poly interconnect
  # cond_pa : caa !cpg !csn : caa  : 70          : p   # p+ active area
  # cond_na : caa !cpg  csn : caa  : 50          : n   # n+ active area
  cond_pa : caa !cpg !csn : caa  : 0           : p   # p+ active area
  cond_na : caa !cpg  csn : caa  : 0           : n   # n+ active area
  cond_well : cwn              : cwn  : 0           : n   # n well

fets :
  # name : condition      : gate d/s : bulk
  # nenh : cpg caa  csn : cpg  caa : @sub # nenh MOS
  # penh : cpg caa !csn : cpg  caa : cwn  # penh MOS
  nenh : cpg caa  csn : cpg  caa (!cpg caa csn) : @sub # nenh MOS
  penh : cpg caa !csn : cpg  caa (!cpg caa !csn) : cwn  # penh MOS

```

```

contacts :
# name      : condition          : lay1 lay2 : resistivity
cont_s : cva cmf cms           : cmf  cms  : 1    # metal to metal2
cont_p : ccp cmf cpg           : cmf  cpg  : 100   # metal to poly
cont_a : cca cmf caa !cpg cwn !csn
        | cca cmf caa !cpg !cwn csu
        : cmf  caa  : 100   # metal to active area
cont_w : cca cmf cwn csu       : cmf  cwn  : 80   # metal to well
cont_b : cca cmf !cwn !csu     : cmf  @sub : 80   # metal to subs

junction capacitances ndif :
# name      : condition          : mask1 mask2 : capacitivity
# acap_na : caa                 !cpg csu !cwn : @gnd caa : 100 # n+ bottom
# ecap_na : !caa -caa !-cpg -csu !-cwn : @gnd -caa : 300 # n+ sidewall

junction capacitances nwell :
acap_cw : cwn                  : @gnd cwn : 100 # bottom
ecap_cw : !cwn -cwn           : @gnd -cwn : 800 # sidewall

junction capacitances pdif :
# acap_pa : caa                 !cpg !csu cwn : caa cwn : 500 # p+ bottom
# ecap_pa : !caa -caa !-cpg !-csu cwn -cwn : -caa cwn : 600 # p+ sidewall

capacitances :
# polysilicon capacitances
acap_cpg_sub : cpg              !caa !cwn : cpg @gnd : 49
acap_cpg_cwn : cpg              !caa cwn : cpg cwn : 49
ecap_cpg_sub : !cpg -cpg !cmf !cms !caa !cwn : -cpg @gnd : 52
ecap_cpg_cwn : !cpg -cpg !cmf !cms !caa cwn : -cpg cwn : 52

# first metal capacitances
acap_cmf_sub : cmf              !cpg !caa !cwn : cmf @gnd : 25
acap_cmf_cwn : cmf              !cpg !caa cwn : cmf cwn : 25
ecap_cmf_sub : !cmf -cmf !cms !cpg !caa !cwn : -cmf @gnd : 52
ecap_cmf_cwn : !cmf -cmf !cms !cpg !caa cwn : -cmf cwn : 52

acap_cmf_caa : cmf              caa !cpg !cca : cmf caa : 49
ecap_cmf_caa : !cmf -cmf caa !cms !cpg : -cmf caa : 59

acap_cmf_cpg : cmf              cpg !ccp : cmf cpg : 49
ecap_cmf_cpg : !cmf -cmf cpg !cms : -cmf cpg : 59

```

APPENDIX I -- Email From OptEM

I have few issues with the invert demo
please see the **highlight** message below for detail.

Netlist #1:

invert

* Generated by: xspice 2.39 25-Jan-2006

* Date: 3-May-06 22:53:31 GMT

* Language: SPICE

* circuit invert b_in t_out l_vdd r_vdd l_vss r_vss

vnet1 l_vdd r_vdd 0

vnet2 l_vss r_vss 0

m1 t_out b_in l_vss l_vss nenh_0 w=6.8u l=1.2u

m2 l_vdd b_in t_out l_vdd penh_0 w=12.4u l=1.2u

c1 l_vdd b_in 795.12e-18

c2 l_vdd t_out 3.04892f

d1 t_out l_vdd pdif area=20.48p pj=14.8u

/// Where is other pdif diode? It was skipped because it was l_vdd to l_vdd?

c3 l_vdd GND 249.6e-18

d2 l_vdd GND nwell area=155.52p pj=50.4u

/// Why forward bias the nwell diode???

c4 b_in t_out 495.12e-18

c5 b_in GND 3.21264f

c6 b_in l_vss 282.72e-18

c7 t_out GND 987.68e-18

d3 t_out GND ndif area=16.16p pj=11.2u

/// Why the diode node connection were switched, this forward bias the drain of nmos???

c8 GND l_vss 774.4e-18

d4 GND l_vss ndif area=27.92p pj=22.6u

/// Why not skip this diode like the other pdif diode since the it is GND to l_vss?

* end invert

Netlist #2

invert

* Generated by: xspice 2.39 25-Jan-2006

* Date: 3-May-06 22:55:00 GMT

* Language: SPICE

* circuit invert b_in t_out l_vdd r_vdd l_vss r_vss

vnet1 l_vdd r_vdd 0

vnet2 l_vss r_vss 0

m1 t_out b_in l_vss l_vss nenh_0 w=6.8u l=1.2u as=27.92p ps=22.6u nrs=0.603806

```
/// What happened to as, pd, nrd parameter??
m2 l_vdd b_in t_out l_vdd penh_0 w=12.4u l=1.2u
/// What happened to all parameter??
c1 l_vdd b_in 795.12e-18
c2 l_vdd t_out 3.04892f
c3 l_vdd GND 249.6e-18
d1 l_vdd GND nwell area=155.52p pj=50.4u
c4 b_in t_out 495.12e-18
c5 b_in GND 3.21264f
c6 b_in l_vss 282.72e-18
c7 t_out GND 987.68e-18
c8 GND l_vss 774.4e-18
* end invert
```

APPENDIX J -- Changed Space Code

```

Index: gettech.cc
=====
RCS file: /users/space/main/cvsroot/CACD/src/space/extract/gettech.cc,v
retrieving revision 4.97
retrieving revision 4.98
diff -r4.97 -r4.98
951a952
>             if (capPolarityTab[cap -> sortNr] == 'p') cap -> sortNr += 1;

Index: out.c
=====
RCS file: /users/space/main/cvsroot/CACD/src/space/lump/out.c,v
retrieving revision 4.101
retrieving revision 4.102
diff -r4.101 -r4.102
879,880d878
<             if (defGndCapOtherPol[0] != cnet.net_name[0]) setGndCapPol ();
<
881a880
>             if (defGndCapOtherPol[0] != cnet.net_name[0]) setGndCapPol ();
902,903d900
<             if (defSubCapOtherPol[0] != cnet.net_name[0]) setSubCapPol ();
<
904a902
>             if (defSubCapOtherPol[0] != cnet.net_name[0]) setSubCapPol ();
956c954
<             cnet.net_name[0] = (capPolarityTab[i] != 'n')? 'n' : 'p';
---
>             cnet.net_name[0] = (capPolarityTab[i] != 'n')? 'p' : 'n';
1101c1099
<             cnet.net_name[0] = (capPolarityTab[i] == 'n')? 'n' : 'p';
---
>             cnet.net_name[0] = (capPolarityTab[i] != 'n')? 'n' : 'p';

```

APPENDIX K -- Technology File Changes

```

Index: space.def.s
=====
RCS file: /users/space/main/cvsroot/CACD/src/process/process/scmos_n/space.def.s,v
retrieving revision 1.6
retrieving revision 1.7
diff -r1.6 -r1.7
66,69c66,69
< junction capacitances ndif :
< # name      : condition                : mask1 mask2 : capacitivity
<   acap_na   : caa          !cpg  csn !cwn : caa @gnd : 100 # n+ bottom
<   ecap_na   : !caa -caa !-cpg -csn !-cwn : -caa @gnd : 300 # n+ sidewall
---
> junction capacitances ndif :
> # name      : condition                : mask1 mask2 : capacitivity
>   acap_na   : caa          !cpg  csn !cwn : @gnd  caa : 100 # n+ bottom
>   ecap_na   : !caa -caa !-cpg -csn !-cwn : @gnd -caa : 300 # n+ sidewall
71,73c71,73
< junction capacitances nwell :
<   acap_cw   : cwn                        : cwn @gnd : 100 # bottom
<   ecap_cw   : !cwn -cwn                  : -cwn @gnd : 800 # sidewall
---
> junction capacitances nwell :
>   acap_cw   : cwn                        : @gnd  cwn : 100 # bottom
>   ecap_cw   : !cwn -cwn                  : @gnd -cwn : 800 # sidewall

```