BNL-65754
Informal Report

TOWARDS A MORE ACCURATE EXTRACTION OF THE SPICE NETLIST FROM MAGIC BASED LAYOUTS*

Gianluigi De Geronimo
Brookhaven National Laboratory
Upton, NY 11973-5000

August, 1998

*This research was supported under Contract No. DE-AC02-98CH10886.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, not any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency, contractor, or subcontractor thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency, contractor or subcontractor thereof.

Printed in the United States of America
Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road
Springfield, VA 22161

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

Index

Introduction

The extraction of the SPICE netlist form MAGIC based layouts is investigated. It is assumed that the layout is fully coherent with the corresponding mask representation [1].

The process of the extraction can be made in three steps:

step 1: extraction of .EXT file from layout, through MAGIC command *extract*
step 2: extraction of the netlist from .EXT file through *ext2spice* extractor
step 3: correction of the netlist through *ext2spice.corr* program.

Each of these steps introduces some approximations, most of which can be optimized, and some errors, most of which can be corrected. Aim of this work is the description of each step, of the approximations and errors on each step, and of the corresponding optimizations and corrections to be made in order to improve the accuracy of the extraction.

The HP AMOS14TB 0.5µm process with linear capacitor and silicide block options and the corresponding SCN3MLC_SUBM.30.tech27 technology file will be used in the following examples.

This work was supported by the US Department of Energy, Contract No. DE-AC02-98CH10886.

Step 1 - From the Layout to the .EXT file

The MAGIC command *extract* provides the extraction of the layout according to some information contained in the technology file. The technology file is composed of 17 sections (*tech, version, planes, types, contact, styles, compose, connect, cifoutput, cifinput, mzrouter, drc, extract, wiring, router, plowing, plot*) and the sections of interest for the extraction are the 8th (*connect*) and the 13th (*extract*).

The *connect* and the *extract* sections contained in the SCN3MLC_SUBM.30.tech27 Technology File are reported in Appendix A. See MAGIC Maintainer's Manual #2 (Technology File Manual) for the formalism used in these sections.

The information contained in the .EXT file

The circuit is always extracted by using three elements: the **node** element, the **capacitance between nodes** element, and the **FET** element.

[1] It is good practice to write into cif format (:cif MAGIC command) and to read from the written cif format (:cif read MAGIC command) the layout before the extraction.

a) Node Element

Each λ×λ tile of the layer is part of a node. The node is determined through the information contained in the *connect* section of the technology file. If each tile is *connected* to the adjacent one, both tiles are part of the same node. To each node element corresponds a line in the .EXT file which looks as follows:

node *nodename R C X Y layer a1 p1 a2 p2 a3 p3 ....*

where:

nodename is the label of that node. MAGIC assigns an arbitrary name if there is no label on that node.

R is the resistance of the node and it is calculated according to the formula:

$$\begin{cases} R = \sum_i \mathrm{R}_i \\ \mathrm{R}_i = \dfrac{\mathrm{RSQ}_i}{2}\left[\dfrac{\mathrm{P}_i^2}{4\mathrm{A}_i} - 2 + \sqrt{\left(\dfrac{\mathrm{P}_i^2}{4\mathrm{A}_i} - 2\right) - 4}\right] \end{cases} \tag{1}$$

where RSQ$_i$ is the resistance per square of the ith layer, and its value in mOhm/square (default=0) is given in the *extract* section under the corresponding keyword *resist* (giving the precedence to the upper plane in case a layer is represented in more than one plane, as it is for the contacts), A$_i$ and P$_i$ are respectively the total area and the total perimeter of the ith layer. The value of *R* is expressed in Ohm and it is approximated to the nearest integer. This formula gives an exact value only for rectangular sections composed of a single layer. The resistance is always calculated along the longest direction of the rectangle. In all other cases *R* represents a bad approximation and for this reason *R* is ignored during Step 2 of the extraction. As a consequence, the values reported in the *extract* section under the keyword *resist* are not determinant for the extraction.

C is the capacitance between the node and the substrate and it is calculated according to the formula:

$$\begin{cases} C = \sum_i \mathrm{C}_i \\ \mathrm{C}_i = \mathrm{ACAP}_i \times \mathrm{A}_i + \mathrm{PCAP}_i \times \mathrm{P}_i \end{cases} \tag{2}$$

where ACAP$_i$ and PCAP$_i$ are respectively the capacitance per λ^2 associated to the area and the capacitance per λ associated to the perimeter (fringing) of the ith layer, and their values in aF (default=0) are given in the *extract* section respectively under the keywords *areacap* and *perimcap* (see the Manual for the formalism), A$_i$ and P$_i$ are respectively the total area and the total perimeter of the ith layer. The value of *C* is expressed in aF and it is approximated to the nearest integer. In most cases *C* represents a good approximation and for this reason it is taken into account during Step 2 of the extraction. As a consequence, the values reported in the *extract* section under the keywords *areacap* and *perimcap* are determinant for the final extraction and their value must be accurately verified and updated through the General Process Specifications and the Beta Technology Files and the MOSIS Parametric Test Results.

X Y are the coordinates of the node in units of λ. For single layer nodes, the low corner of the most left side is assumed as origin. For multiple layer nodes, the low left corner of the most down among the most left contacts is assumed as origin.

layer is the name of one of the layers of the node. For single layer nodes, it is the name of that layer. For multiple layer nodes, it is the name of the contact the corner of which is assumed as origin.

a1 p1 a2 p2 a3 p3 ... are respectively the area and the perimeter of the layers of the node which belong to the resist-classes 1, 2, 3 ... where resist-class 1 is the first line with keyword *resist* in the *extract* section, class 2 is the second line with keyword *resist* and so on. In the technology file used here for the examples, there are 6 resist-classes. The values *a1, p1, a2, p2,* ... are used for the evaluation of the resistance of the node but not for the evaluation of its capacitance. As a consequence these values are not determinant for the final extraction.

b) Capacitance Element

If two or more nodes are present in the layout, the capacitance between each couple of nodes (if > 0) is extracted, and reported through the capacitance element. The corresponding line in the .EXT file looks as follows:

cap *nodename1 nodename2 C*

where:

nodename1 nodename2 are the labels of the two nodes; MAGIC uses an the arbitrary names if there is no label.

C is the capacitance between the two nodes, in aF.

The capacitance between two nodes is evaluated only in the following cases:

- a layer of one node overlaps a layer of the other node and the two layers belong to different planes; the corresponding keyword in the *extract* section is *overlap* (aF/λ^2), and includes the possibility of a shield layer between the two layers (see Manual); if this capacitance is extracted for a given area, an equal area is subtracted in the evaluation of the node capacitance to substrate for the layer belonging to the upper plane, which is correct;

- a layer of one node is adjacent to a layer of the other node and the two layers belong to different planes; the corresponding keyword in the *extract* section is *sideoverlap* (aF/λ) and includes the possibility of a shield between the two layers; if this capacitance is extracted for a given perimeter, an equal perimeter is subtracted in the evaluation of the node capacitance to substrate for the layer belonging to the upper plane, which is correct;

- a layer of one node is close, but not adjacent, to a layer of the other node and the two layers belong to the same plane; the corresponding keyword in the *extract* section is *sidewall* (aF, and must be calculated by using the factor $2\times\lambda_w/\lambda_d$ where λ_d is the distance and λ_w the width); the capacitance is extracted only if the distance between the adjacent layers is below the value reported under the *sidehalo* keyword at the beginning of the *extract* section.

c) FET Element

If two nodes are connected through a specific layer, and it exists a corresponding *fet* line with the specified layer indicated in the 2nd element under the corresponding keyword *fet* in the *extract* section (see Manual), a FET is extracted, the gate of which is the specific layer. The corresponding line in the .EXT file looks as follows:

fet *model X Y X+1 Y+1 a p snode gatenodename gatewidth 0 nodename1 width1 jlabel1 nodename2 width2 jlabel2*

where:

model is the name of the model of the FET, as specified in the 5th element under the corresponding keyword *fet* in the *extract* section.

X Y X+1 Y+1 are the Cartesian coordinates and the Cartesian coordinates plus 1 of the low left corner gate layer in units of λ.

a p are respectively the area and the perimeter of the gate

snode is the label of the node at which the substrate must be considered connected, as specified in the 6th element under the corresponding keyword *fet* in the *extract* section. The [7th] facultative element under the corresponding keyword *fet* in the *extract* section considers the connection of the body to a well, when the well is superposed to the channel of the FET (see Manual).

gatenode is the label of the gate node; ***gatewidth*** is the width of the gate contact (i.e. the length of the channel of the FET) in units of λ.

nodemname1, nodename2 are respectively the labels of the drain and of the source, the layer (and number) of which are specified in the 3rd (and 4th) element under the corresponding keyword *fet* in the *extract* section (see the Manual), ***width1, width2*** are respectively the widths of the drain and of the source (i.e. the lwidth of the channel of the FET) in units of λ and ***jlabel1, jlabel2*** are respectively the labels at the drain and at the source junction (in the layout these labels must be followed by the "$" symbol). According to the Magic extraction criteria, during Step 1 the drain junction of each finger is the one at the left or below the gate. This criteria for the choice of the drain is not determinant for the final extraction because, as it will be discussed, during Step 2 the side with the *jlabel* S$ is assumed as source and the one with the *jlabel* D$ is assumed as drain, disregarding the orientation of the gate.

The errors contained in the .EXT file

Three **errors** are made during Step 1 in the evaluation of the capacitance for the node and capacitance element.

Type 1 Error. An over-estimation is made by the extractor when two or more layers of the same node and belonging to different planes are overlapped. The error is due to the fact that the capacitance of the upper layers to the substrate is in most cases added, but its value should be zero because of the shield of the lower layer.
This is the case for example of all the contact layers (pc, ndc, pdc, nsc, psc, cwc), which are characterized by the fact that lie both in the *active* plane and in the *m1* plane (see Manual). In the evaluation of the capacitance of the area to the substrate, both the planes are considered, but only the *active* plane should be considered because the *m1* plane is shielded by the *active* plane.
In the evaluation of the capacitance of the perimeter to the substrate, both the planes are considered, but only the *active* plane should be considered because the *m1* plane is shielded, except when the contact layers are along the perimeter of the node.
Both problems could be corrected by suitably modifying few lines of the *extract* section, as shown. The line [2]

```
areacap (m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1
2.700
```

must be replaced with the line

```
areacap (m1,m2c/m1)/metal1   2.700
```

which means that no capacitance to substrate is calculated for the part of the contact lying in the metal1 plane.

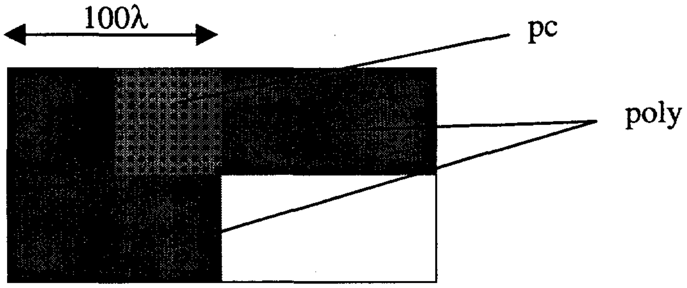
The line

```
perimc (ndc/a,nsc/a,pdc/a,psc/a,pc/a,cwc/a)/active (ndiff,nsd,pdiff,psd,poly,cwnsd)/active
-9.300
```

(note the minus sign) must be added, which means that, if the contact is not along the perimeter of the node, the capacitance of that perimeter to the substrate is adjusted to be zero.

[2] Don't be surprised if sometimes the name of a layer is reported more than once in the same group of a line in the Technology File (for example cwc/m1 appears twice in this case). The origin of this redundancy is at the Technology File generation level (ask Anand Kandasamy for details), but it is without consequence for the extraction.

Example 1



From .EXT file:

node "a_82_n45#" 7 136410 82 -45 p 0 0 0 0 15000 600 2500 200 0 0 0 0

resistance from Eq.(1):

$$\frac{\left(\frac{P1^2}{4\cdot A1}-2\right)+\sqrt{\left(\frac{P1^2}{4\cdot A1}-2\right)^2-4}}{2}\cdot 1.8+\frac{\left(\frac{P2^2}{4\cdot A2}-2\right)+\sqrt{\left(\frac{P2^2}{4\cdot A2}-2\right)^2-4}}{2}\cdot 0.07=6.788$$

capacitance from Eq.(2): $A1\cdot 7.92+P1\cdot 15+A2\cdot 2.7+P2\cdot 9.3=136410\,\blacksquare$

which is over-estimated. After the modification of the technology file, it results from .EXT file:

node "a_82_n45#" 7 128265 82 -45 p 0 0 0 0 15000 600 2500 200 0 0 0 0

capacitance from Eq.(2): $A1\cdot 7.92+P1\cdot 15+(P2-50-50-50)\cdot 9.3=128265\,\blacksquare$

and it is a better approximation.

It is worth noting that this is only a partial solution of the problem, and it is subject to errors. For example when a *m1* line connected to the contact flows above the active layer of that contact, its capacitance to the substrate is erroneously added. An alternative solution not subject to errors will be proposed in the next section.

Type 2 Error. In the case of *contacts, metals* and *poly* which flow over a *well* and are connected to that well (i.e. are parts of the same node), the area and perimeter capacitances to substrate of the metal and poly as well as the part of the perimeter capacitance to substrate of the contact adjacent to the well are added (no overlap or sideoverlap to the well and consequent subtraction occurs, because the well is part of the same node). This is an error because all these layers are shielded by the well.

To give a simple example, the linear capacitors can be considered, where the perimcap capacitance to substrate of *cwc* (the well contact) along the perimeter which is not adjacent to the *cwnsd* (the n-diffusion) as well as the area and perimeter capacitance to substrate of a metal connected to the *cwc* are erroneously added.
In the case of linear capacitors, a partial solution of the problem can be obtained by introducing the following additional changes in the *extract* section.

The line:

```
perimc (m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1
~(m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1   9.300
```

must be replaced with the line (the layer *cwc/m1* is deleted):

```
perimc (m1,ndc/m1,nsc/m1,pdc/m1,psc/m1,pc/m1,m2c/m1)/metal1
~(m1,ndc/m1,nsc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1   9.300.
```

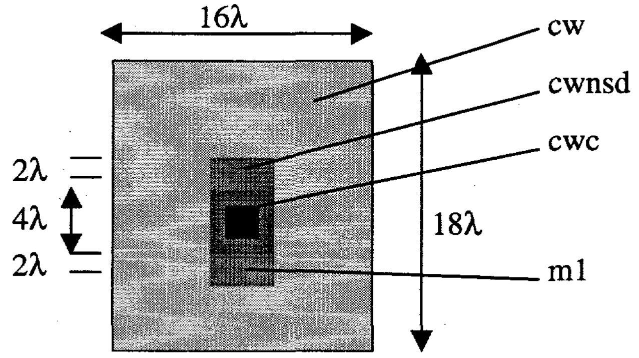
The line added to correct the Error 1

```
perimc (ndc/a,nsc/a,pdc/a,psc/a,pc/a,cwc/a)/active (ndiff,nsd,pdiff,psd,poly,cwnsd)/active
-9.300
```

must be replaced with the line:

```
perimc (ndc/a,nsc/a,pdc/a,psc/a,pc/a)/active (ndiff,nsd,pdiff,psd,poly)/active   -9.300
```

<u>Example 2</u>



From .EXT file:

node "w_246_n38#" 0 2606.88 246 -38 cw 0 0 0 0 0 0 24 20 0 0 0 0

capacitance from Eq.(2): $A1 \cdot 8.46 + P1 \cdot 0 + A2 \cdot 2.7 + P2 \cdot 9.3 = 2606.88$ ▪

which is over-estimated. After the further modification of the technology file, it results from .EXT file:

```
node "w_246_n38#" 0 2532.48 246 -38 cw 0 0 0 0 0 0 24 20 0 0 0 0
```

capacitance from Eq.(2): $A1 \cdot 8.46 + P1 \cdot 0 + A2 \cdot 2.7 + (P2 - 4 - 4) \cdot 9.3 = 2532.48$.

and it is a better approximation.

It is worth noting that an over-estimating error due to the capacitance of the metal to substrate is still present. The solution proposed in the next section will solve also this problem.

Type 3 Error. In the case of linear capacitors, the sideoverlap capacitance between *wcap* (the active area of the capacitor) and *cwnsd* (the diffusion) and the sideoverlap capacitance between *poly* and *capwell* (the well of the capacitor) are considered under the same *sideoverlap* keyword. This is an error because they are characterized by different thickness of the oxide and then subject to two different values of aF/λ. To solve this problem the following changes must be made.

The line:

```
sideoverlap (poly,pres,wcap,pc/a,wcap,pc/a)/active
~(poly,pres,wcap,pc/a,wcap,pc/a)/active ~space/w  15
```

must be replaced with the line:

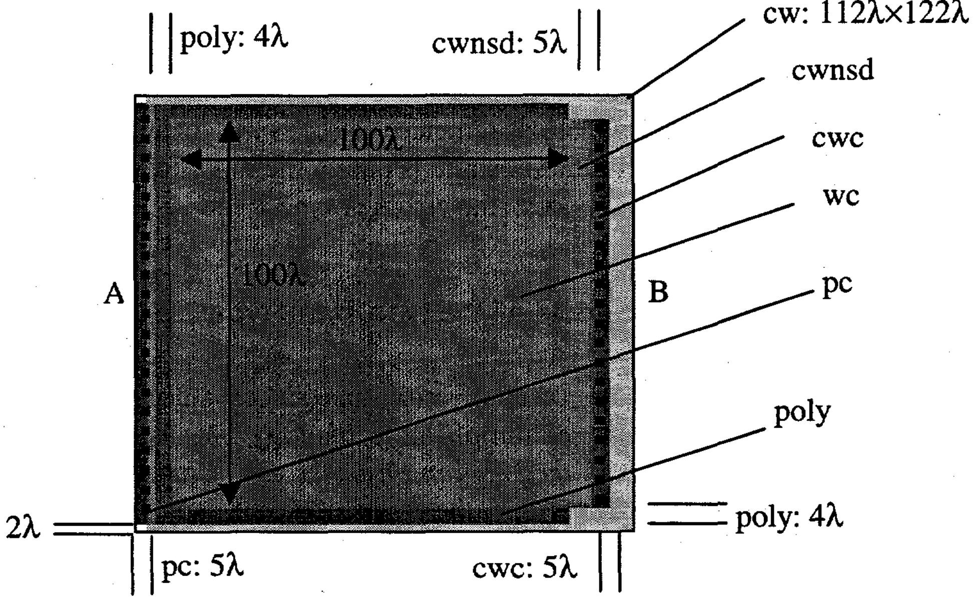
```
sideoverlap (poly,pres,wcap,pc/a,wcap,pc/a)/active
~(poly,pres,wcap,pc/a,wcap,pc/a,cwnsd)/active ~space/w  15
```

and the line

```
sideoverlap (wcap)/active (cwnsd)/active ~space/w  40
```

must be added, where the value 40 aF/λ can be deduced from a simulation, from a suitably equation or from the MOSIS Parametric Test Results (see Appendix B).

Example 3



From .EXT file:

```
node "A" 3 5336.28 141 -18 pc 0 0 0 0 11772 434 540 226 0 0 0 0
node "B" 1 115597 144 -20 cw 0 0 0 0 0 0 500 210 0 0 0 0
cap "A" "B" 2.09531e+06
```

Capacitances from Eq.(2): $A1\cdot 7.92 + P1\cdot 15 + P1\cdot 9.3 = 5336.28$

$$A2\cdot 8.46 = 115597.44$$

which are correct.

Internodal capacitance: $A3\cdot 207.9 + (A4 - A3)\cdot 7.92 + P4\cdot 15 + P5\cdot 9.3 = 2095305.36$

which is under-estimated due to Error 3.

After the further modification of the technology file, it results from the .EXT file:

```
node "A" 3 5336.28 141 -18 pc 0 0 0 0 11772 434 540 226 0 0 0 0
node "B" 1 115597 144 -20 cw 0 0 0 0 0 0 500 210 0 0 0 0
cap "A" "B" 2.09781e+06
```

Internodal capacitance:

$$A3\cdot 207.9 + (A4 - A3)\cdot 7.92 + (P4 - 100)\cdot 15 + P5\cdot 9.3 + 100\cdot 40 = 2097805.36$$

which is a better (indeed with negligible difference in this case) approximation.

It is worth noting that, in this case, the solution adopted is effective in the correction of the error in the evaluation of the internodal capacitance. On the other hand, the absence of errors in the evaluation of the capacitance to the substrate is due to the modifications introduced in the *extract* section of the technology file to correct the errors of type 1 and 2.

A solution which globally corrects all the type 1 and type 2 errors

It has been shown that the three types of errors introduced during Step 1 can be partially corrected by modifying the *extract* section of the technology file. Apart from the correction of the error of type 3, which *must be introduced in any case* and it represents a global correction, the other modifications correct only partially the errors of type 1 and 2.

A different approach to globally correct **all** the errors of type 1 and 2 is now discussed. This approach is based on the introduction of an additional plane (we will call it "substrate") and of an additional layer (we will call it "substrate" or "ss") which lies in the additional substracte plane. By using this additional layer the capacitances to substrate are now evaluated as *overlap* and *sideoverlap* capacitances, instead of *areacap* and *perimc* capacitances. The main advantage is that the shielding layers (available only with the *overlap* and *sideoverlap* keywords) can now be used for the capacitances to substrate. Another advantage is that the new layer, which must be present in all the layout under extraction, can be divided into areas thus allowing the simulation of the resistance of the substrate from one point to another of the layout. This additional layer must be also invisible at the CIF extraction level. The modifications which must be made in the technology file to introduce and use the additional plane and layer are discussed in the Appendix C. A label SS can be given to the additional node the layer of which is ss.

Examples

By applying the modified technology to **Example 1** it follows from the .EXT file:

without the ss layer (note the zero value capacitance)

```
node "a_82_n172#" 7 0 82 -172 p 0 0 0 0 15000 600 2500 200 0 0 0 0
```

With the ss layer (note that the capacitance to substrate is now represented by the capacitance element, while the node elements are ignored during the step 2 of the extraction because their capacitance is zero)

```
node "a_82_n172#" 7 0 82 -172 p 0 0 0 0 15000 600 2500 200 0 0 0 0
node "SS" 0 0 74 -184 ss 0 0 0 0 0 0 0 0 0 0 0 0
cap "a_82_n172#" "SS" 128265
```

By applying the modified technology to **Example 2** it follows from the .EXT file:

Without the ss layer (note the zero value capacitance)

```
node "w_246_n38#" 0 0 246 -38 cw 0 0 0 0 0 0 24 20 0 0 0 0
```

With the ss layer (again note that the capacitance to substrate is now represented by the capacitance element)

```
node "w_246_n38#" 0 0 246 -38 cw 0 0 0 0 0 0 24 20 0 0 0 0
node "SS" 0 0 245 -39 ss 0 0 0 0 0 0 0 0 0 0 0 0
cap "w_246_n38#" "SS" 2436.48
```

and now the result is now correct: $A1 \cdot 8.46 = 2436.48$.

By applying the modified technology to **Example 3** it follows from the .EXT file:

Without the ss layer (note the zero value capacitance)

```
node "A" 3 0 141 -18 pc 0 0 0 0 11772 434 540 226 0 0 0 0
node "B" 1 0 144 -20 cw 0 0 0 0 0 0 500 210 0 0 0 0
cap "A" "B" 2.09781e+06
```

With the ss layer (again note that the capacitances to substrate is now represented by the capacitance element)

```
node "A" 3 0 141 -18 pc 0 0 0 0 11772 434 540 226 0 0 0 0
node "B" 1 0 144 -20 cw 0 0 0 0 0 0 500 210 0 0 0 0
node "SS" 0 0 134 -25 ss 0 0 0 0 0 0 0 0 0 0 0 0
cap "A" "SS" 5336.28
cap "A" "B" 2.09781e+06
cap "B" "SS" 115597
```

Step 2 - From the .EXT file to the .CIR file (Ext2spice)

The Ext2spice program provides the extraction of the Spice netlist from the Magic.EXT file. The Ext2spice program here considered was released by Lloyd Clonts (Oak Ridge National Laboratory) in June 23rd 1998 [3].

The Ext2spice program

The Ext2spice program runs with the command line:

ext2spice [options] *nomefile.ext*

and the output (*-g* option) are two files:

nomefile.cir	Spice netlist containing options, sources, library lines and the line .include *nomefile.spice*
nomefile.spice	Spice netlist containing the elements extracted from the .EXT file

The netlist containing the elements extracted from the .EXT file will be discussed, element by element, in the next sections. Information about the options can be found in the Ext2spice manual (Appendix D). Ext2spice uses a support file called Ext.defs (Appendix E) the role of which will be discussed in the next sections. In the Ext.defs file are also contained the Spice command lines which will appear in the .cir file.

Capacitor Extraction

The capacitors from the node element and from the capacitance element of the .EXT file are reported in the .spice file.

In particular, to each node element in the .EXT file

node *nodename R C X Y layer a1 p1 a2 p2 a3 p3* ....

it corresponds in the .spice file the element

Cx *newnodename $G_Vsub! C*

[3] In the original release (1QT 1998) some errors were found, which have been corrected by Lloyd under our stimulus, generating the new release here discussed.

where Cx is an arbitrary name and *newnodename* is equal to *nodename* if the node has a label or an arbitrary node number (indeed much shorter than the arbitrary node names used by Magic in the .EXT file) if there is no label on that node. The line "Vsubstr $G_Vsub! 0 DC 0V" is added by Ext2spice (*-D* option) to insure the connection of the global substrate node $G_Vsub!.

To each capacitance element in the .EXT

> cap *nodename1 nodename2 C*

it corresponds, in the .spice file, the element

> Cx *newnodename1 newnodename2 C*

where Cx is the name of the element according to the general device labeling (see Ext2spice Manual) or an arbitrary name if there is no label on that element; *newnodename1* (*newnodename2*) is equal to *nodename1* (*nodename2)* if the node has a label or an arbitrary node numbers (much shorter than the arbitrary node name used in the .EXT file) if there is no label on that node.

It is worth pointing out that a capacitance is reported only if it is not floating and if its value is higher that the value indicated with the option *-c cmin* (default minimum: 1fF). All the values of these capacitances (linear, parasitics, ...) are reported as well. The accuracy in their value is related to the accuracy of their extraction during Step 1.

FET Extraction

All the FETs from the fet element of the .EXT file are reported in the .spice file.

In particular, to each fet element in the .EXT file (with some exceptions discussed in the next sections, such as in the case of the extraction of resistors)

> fet *model X Y X+1 Y+1 a p snode gatenodename gatewidth 0 nodename1 width1 jlabel1 nodename2 width2 jlabel2*

it corresponds in the .spice file the element

> Mx *newdrainnodename newgatenodename newsourcenodename substratenodename newmodel [M=m] W=w L=l AD=ad PD=pd AS=as PS=ps [NRG=nrg NRD=nrd NRS=nrs]*

If the *-mm* option is used, multifinger FETs are merged in a single FET by adding the *M=m* term, where m is the number of fingers.

Mx is the name of the element according to the general device labeling (see Ext2spice Manual) or an arbitrary name if there is no label on that element;

newgatenodename is equal to *gatenodename* if the node has a label or an arbitrary node numbers (much shorter than the arbitrary node name used by Magic in the .EXT file) if there is no label on that node.

newdrainnodename newsourcenodename respectively the drain (to which the area AD and the perimeter PD are associated) and the source name (to which the area AD and the perimeter PD are associated), are equal to *nodename1* and *nodename2* with a order which, if *jlabel1* and *jlabel2* are zero, depends on the orientation of the FET. The drain node is chosen according to Magic extraction criteria where the drain junction of each finger is the one at the left or below the gate. On the contrary, if one *jlabel1* and/or *jlabel2* is labeled as D (or S) (remember that in the layout these labels must be followed by the "$" symbol), the corresponding junction will be assumed as drain (or source). Attention must be paid to the fact that, for multifinger FETs, the junction label *jlabel* must lie in the same finger where is placed the label of the device, otherwise it will be ignored !

substratenodename is equal to *snode*

> In order to recognize the substrate node used in Ext2spice ($G_Vsub!) and to use the additional ss layer, the following changes must be made for n-channel FETs in the *fet* lines of the *extract* section:
>
> ```
> # REPLACED fet  nfet ndiff,ndc 2 nfet Gnd! pwell 15 324
>  fet  nfet ndiff,ndc 2 nfet $$G_Vsub! pwell,ss 15 324
> # REPLACED fet  nfet ndiff,ndc 1 nfet Gnd! pwell 15 324
>  fet  nfet ndiff,ndc 1 nfet $$G_Vsub! pwell,ss 15 324
> ```
>
> A problem arises when the body must be connected to pwell. When the MAGIC extractor must choose between ss and pwell, it chooses ss (i.e. the layer with the lower planeorder), which is an error. A way to solve this problem is to delete the layer ss from the areas where nfet and pwell are simultaneously present. On the other hand, for most of the technologies that we use, the connection of the body of an n-channel FET to a pwell different from the p substrate is not allowed (i.e. the body of n-channel FETs is always connected to the substrate).

newmodel is equal to the model defined in the Ext.defs file with the lines:

> def *model* *newmodel* M n
> def *model* *newmodel* M p
>
> where *model* is equal to *name* of the *fet* element in the .EXT file (see Tehcnology File Manual) and it is different for n-channel and p-channel devices

w is the gate width, ant it is equal to *width1* (=*width2*) *l* is the gate length, ant it is equal to *gatewidth*

ad, pd, as and *ps* are the areas and perimeters of drain and source respectively, and the layer to be used in this evaluation is the one indicated under the *resistclass* keyword in the extract section of the technology file. In particular:

> the drain/source layer for n-channel devices is defined by using the Ext.defs line *nrclass n*, where n is the nth *resistclass* line in the extract section;
>
> the drain/source layer for p-channel devices is defined by using the Ext.defs line *prclass n*, where n is the nth *resistclass* line in the extract section.

Note that, in order to correctly extract the areas and perimeters in our case, the original lines in the Ext.defs file:

```
prclass 1
nrclass 2
nwclass 3
pwclass 4
plrclass 5
```

must be replaced with the lines (we are not interested in the nwclass, pwclass and plrclass):

```
prclass 2
nrclass 1
nwclass 9
pwclass 9
plrclass 9
```

Note that, because the area and perimeter capacitances of the diffusions (*ndiff, pdiff*) and of the corresponding contacts (*ndc, pdc*) are included the FET model, the corresponding parasitics in the node and capacitance elements of the .EXT file must be set to zero. In order to achieve this result, the layers *ndiff, pdiff, ndc/a, pdc/a, ndc/m1* and *pdc/m1* must be deleted from all the *areacap, perimcap, overlap, sideoverlap* (not *sidewall*) lines in the *extract* section which contain these layers.

Note also that, according to the *extract* section lines:

```
resist (ndiff,anres,ndc/a,nsd,nsc/a)/active 1900
resist (pdiff,apres,pdc/a,psd,psc/a)/active 2000
```

the layers *anres, nsd, nsc/a, apres, psd, psc/a* take part in the calculus of the areas and perimeters. In order to avoid this error (consider for example the case of a FET diffusion connected to a n-well with a nsc/a all around it: the area and perimeter calculated will be strongly over-estimated!), the two lines must be replaced with the lines:

```
resist (ndiff,ndc/a)/active 1900
resist (pdiff,pdc/a)/active 2000
```

Ext2spice calculates correctly the areas also for multifinger FETs and for FETs which share a diffusion. For equal width FETs sharing a diffusion, it calculates the total area and divides it by the total number of fingers. For different FETs having different width and sharing a diffusion, the calculus is made in order to assign to each finger the same area per unit of width. For example, in the case of two FETs with widths W1 and W2 and number of fingers NF1 and NF2, it results for the area per finger AF2 of the second FET:

$$AF2 = \frac{Atot\dfrac{W2}{W1+W2}}{NF1\dfrac{W1}{W1+W2}+NF2\dfrac{W2}{W1+W2}} \tag{3}$$

where Atot is the total diffusion area. With an analog procedure, the perimeter is correctly calculated for multifinger FETs and for FETs which share a diffusion:

$$PF2 = \frac{Ptot\dfrac{W2}{W1+W2}}{NF1\dfrac{W1}{W1+W2}+NF2\dfrac{W2}{W1+W2}} \tag{4}$$

If the *-mn* option is used, the number of squares of the gate NRG, drain NRD and source NRS of the FET are calculated for the evaluation of the corresponding parasitic resistances.

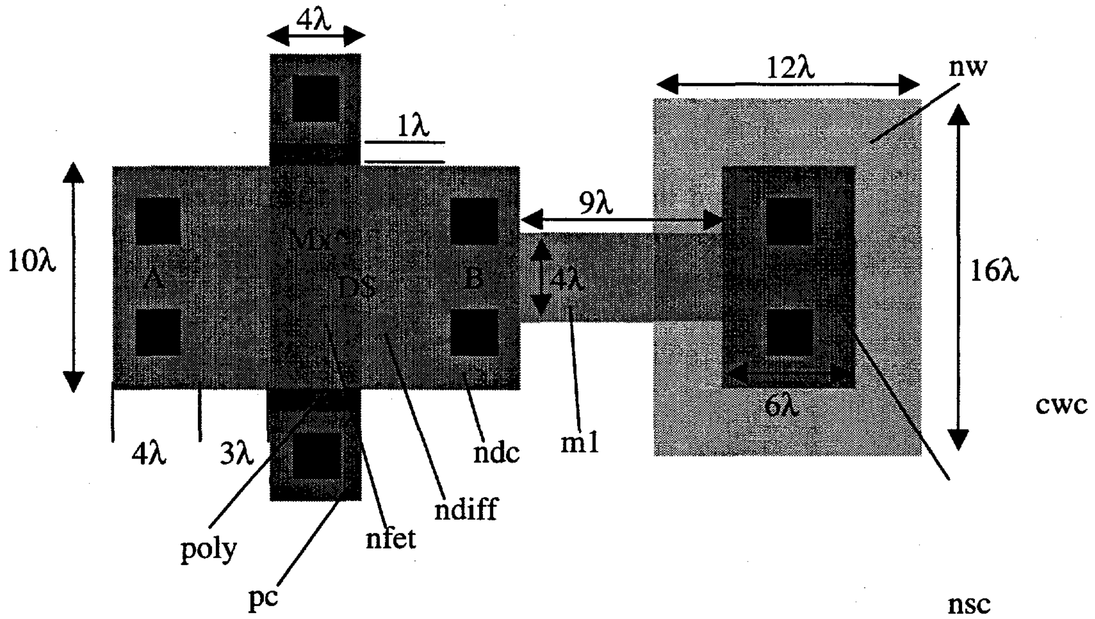
The parameter NRG is calculated only for the FET active area (nfet, pfet) and in the drain to source direction, which can be considered a bad approximation. Moreover, in the case of multifinger FETs, the NRG of a finger is equal to the number of squares of the active area of that finger divided by M, which is an error. Finally, it must be considered that SPICE uses the same resistance per square of the diffusion to calculate RG, RD and RS, which is another error.

The parameters NRD and NRS are calculated in the drain to source direction by using the drain and source areas and perimeters respectively. This approach introduces an approximation which is strongly dependent on the geometry of the diffusion areas. It introduces also an over-estimation because also the areas of the diffusion contacts are included in the calculus. Moreover, in the case of multifinger FETs, the NRD (NRS) of a finger is equal to the sum of the number of squares of the drain (source) divided by M^2, which is an error.

It is worth noting that, during the simulation, SPICE considers the RS, RG, RD as values per finger, and it divides these values by M for the simulation of the overall FET, which is the correct approach.

The Step 3 will provide a more accurate extraction of the parasitic resistances RG, RD and RS, by correcting the errors and by including the resistance of the contact.

Example 4



From the .EXT file (we use the technology file modified as in Step 1, in its version with the ss layer) it results:

```
node "A" 3 0 59 31 ndc 70 34 0 0 0 0 40 28 0 0 0 0
node "C" 9 0 66 26 pc 0 0 0 0 80 48 32 32 0 0 0 0
node "B" 12 0 83 28 nw 130 66 0 0 0 0 136 70 0 0 0 0
node "SS" 0 0 55 23 ss 0 0 0 0 0 0 0 0 0 0 0 0
cap "C" "SS" 1080
cap "A" "SS" 167.4
cap "B" "SS" 1930.92
fet nfet 66 31 67 32 40 28 "Gnd!" "C" 8 "Mx" "A" 10 0 "B" 10 "D"
```

The capacitances of A and B nodes to substrate are over-estimated (for example the capacitance of the A node must be zero, because included in the FET model through the source area and perimeter) and the body of the FET is connected to Gnd! while should be connected to the substrate SS. After the further modifications of the technology file, it results from the .EXT file:

```
node "A" 3 0 59 31 ndc 70 34 0 0 0 0 40 28 0 0 0 0
node "C" 9 0 66 26 pc 0 0 0 0 80 48 32 32 0 0 0 0
node "B" 3 0 83 28 nw 70 34 0 0 0 0 136 70 0 0 0 0
node "SS" 0 0 55 23 ss 0 0 0 0 0 0 0 0 0 0 0 0
cap "C" "SS" 1080
cap "B" "SS" 1800.72
fet nfet 66 31 67 32 40 28 "SS" "C" 8 "Mx" "A" 10 0 "B" 10 "D"
```

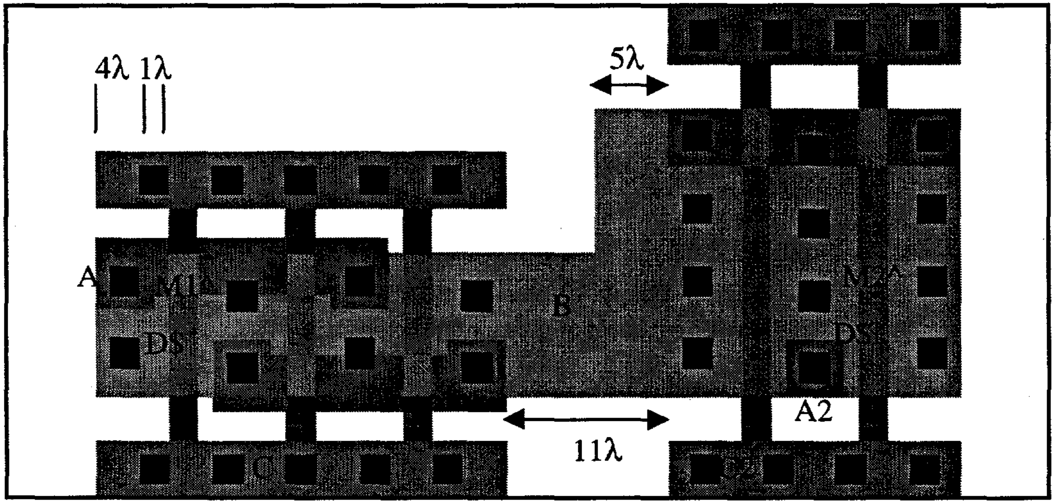
which is correct both in terms of capacitances and of connections.

Now, from the .spice file (*-mn* option) it results:

```
Mx B C A SS CMOSN W=3.00U L=1.20U AD=6.30P PD=10.20U AS=6.30P PS=10.20U
+ NRG=0.40 NRD=0.70 NRS=0.70
C1 B SS 1.8F
C2 C SS 1.1F
```

and the capacitances, areas and perimeters are correctly extracted, while the NRG, NRD and NRS parameters are affected by the previously discussed errors.

Example 5



In this example two n-channel FETs (M1, gate 10λ×2λ, 3fingers and M2, gate 20λ×2λ, 2fingers) share the source.

From the .EXT file (we use the technology file modified as in Step 1 and in Step 2, in its version with the ss layer) it results:

```
node "A2" 7 0 107 31 ndif 120 52 0 0 0 0 80 48 16 16 0 0
node "A" 13 0 61 31 ndc 110 62 0 0 0 0 80 56 88 52 0 0
node "B" 26 0 68 31 ndif 470 164 0 0 0 0 240 152 168 100 0 0
node "C" 61 0 61 24 pc 0 0 0 0 320 212 224 128 0 0 0 0
node "C2" 60 0 100 24 pc 0 0 0 0 264 192 160 96 0 0 0 0
node "SS" 0 0 59 21 ss 0 0 0 0 0 0 0 0 0 0 0 0
cap "A" "SS" 243.96
cap "B" "C2" 248.48
cap "C" "SS" 5598
cap "A" "B" 627.33
cap "A2" "SS" 57.6
cap "B" "C" 248.48
cap "B" "SS" 399.96
cap "C2" "SS" 4075.68
cap "A2" "B" 359.28
cap "A2" "C2" 35.68
cap "A" "C" 248.48
fet nfet 113 31 114 32 40 44 "SS" "C2" 4 "M2" "A2" 20 "D" "B" 20 0
fet nfet 105 31 106 32 40 44 "SS" "C2" 4 0 "B" 20 0 "A2" 20 0
fet nfet 82 31 83 32 20 24 "SS" "C" 4 0 "A" 10 0 "B" 10 0
fet nfet 74 31 75 32 20 24 "SS" "C" 4 0 "B" 10 0 "A" 10 0
fet nfet 66 31 67 32 20 24 "SS" "C" 4 "M1" "A" 10 "D" "B" 10 0
```

Now, from the .spice file (*-mm* and *-mn* options) it results:

```
M1 A C B SS CMOSN M=3 W=3.00U L=0.60U AD=3.30P PD=6.20U AS=6.04P PS=7.03U
+ NRG=0.07 NRD=0.12 NRS=0.22
M2 A2 C2 B SS CMOSN M=2 W=6.00U L=0.60U AD=5.40P PD=7.80U AS=12.09P PS=14.06U
+ NRG=0.05 NRD=0.07 NRS=0.17
C1 A C 0.2F
C2 A2 C2 0.0F
C3 A2 B 0.4F
C4 C2 SS 4.1F
C5 B SS 0.4F
C6 B C 0.2F
C7 A2 SS 0.1F
C8 A B 0.6F
C9 C SS 5.6F
C10 B C2 0.2F
C11 A SS 0.2F
```

and the capacitances, areas and perimeters are correctly extracted, while the NRG, NRD and NRS parameters are again affected by errors. Note that in the perimeter calculation, the parts adjacent to the channel are included, which is a reasonable approach.

From Eq.(3) it results for the shared area:

$$\frac{(10\cdot6+10\cdot11+20\cdot10+20\cdot5)\cdot\frac{10}{20+10}}{3\cdot\frac{10}{10+20}+2\cdot\frac{20}{10+20}}\cdot.3\cdot.3 = 6.043$$

$$\frac{(10\cdot6+10\cdot11+20\cdot10+20\cdot5)\cdot\frac{20}{20+10}}{3\cdot\frac{10}{10+20}+2\cdot\frac{20}{10+20}}\cdot.3\cdot.3 = 12.086$$

From Eq.(4) it results for the shared perimeter:

$$\frac{(10+5+10+5+10+11+11+10+10+10+20+20+20+6+6)\cdot\frac{10}{20+10}}{3\cdot\frac{10}{10+20}+2\cdot\frac{20}{10+20}}\cdot .3 = 7.029$$

$$\frac{(10+5+10+5+10+11+11+10+10+10+20+20+20+6+6)\cdot\frac{20}{20+10}}{3\cdot\frac{10}{10+20}+2\cdot\frac{20}{10+20}}\cdot .3 = 14.057$$

Resistor Extraction

The Magic extractor doesn't have any additional element for the extraction of the resistors. As a consequence, the three basic elements (the *node* element, the *capacitance* element and the *fet* element) must be used for this purpose. A very simple technique for the extraction of the resistors is to extract these devices as particular FETs. This technique will be discussed here and applied to the case of a poly resistors with silicide block (*sb*).

A poly resistor is composed of a polysilicon layer (*poly*) (the great part of which is subject to the silicide block *sb* (*pres*)), and by two polycontacts (*pc*). Till now, when a poly resistor is extracted, the two contacts and the *poly* appears as parts of the same node. The reason is the line:

```
poly,pres,wcap,nfet,wcap,pfet,pc/a    poly,pres,wcap,nfet,wcap,pfet,pc/a
```

which is present in the *connect* section of the technology file and which "tells" the extractor that, when *poly*, *pres* and *pc* are in contact, they it must be considered connected each other (i.e. parts of the same node).

In order to extract the *pres* layer as a FET, this line must be replaced with the line:

```
poly, wcap,nfet,wcap,pfet,pc/a    poly, wcap,nfet,wcap,pfet,pc/a
```

and the line:

```
fet  pres poly 2 polyres $$G_Vsub! ss,nwell,pwell 0 0
```

must be added in the *extract* section.

Ext2spice recognizes this "fet" as a resistance if the Ext.defs file contains the lines:

```
def polyres        RPOLY  R r
distrib polyres    RSQ  0.000 0 ecapp     SQ
distrib ecapp      CPAR  0.0 0
```

where RSQ is the resistance per square of *pres* (90Ω in this technology) and CPAR is the parasitic capacitance (in fF/μ^2) used in the *distributed model* (*-d* option, see Ext2spice Manual). The number of squares SQ of the extracted resistance is then calculated by Ext2spice, from the corresponding fet line in the .EXT file, by using the "channel" (*pres*) area and the "channel" width, according to the equation:

$$SQ = \frac{A}{W^2} \tag{5}$$

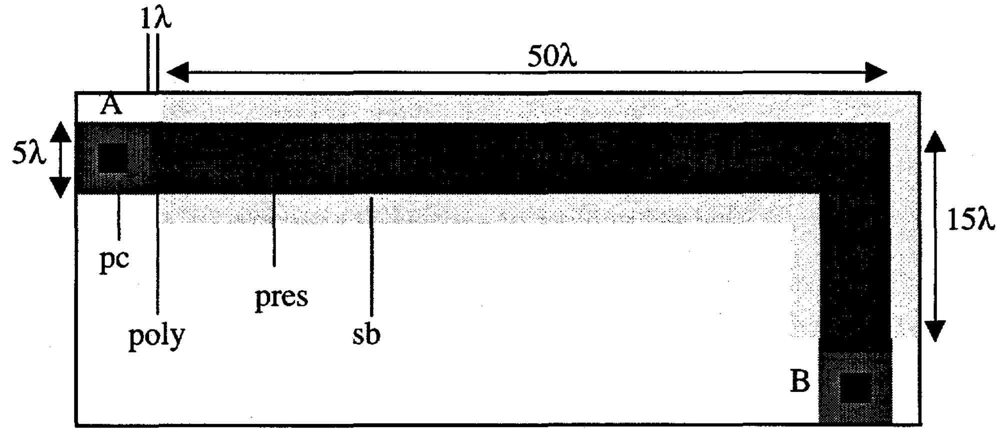
where W is the minimum between the widths of the "drain" and "source" connections (the *poly* adjacent to *pres* in this case). It turns out that the width of these connections is determinant for the correct extraction of the resistance.

In most cases the exracted SQ represents a good approximation of the value of the resistor. On the other hand, for a more accurate approximation:

- each corner along *pres* must be weighted with a factor 0.55 [4];
- the real width of *pres* should be used, which is in general smaller that the drawn width;
- in the distributed case, the perimeter component of the parasitic capacitance is not included;
- the resistance of the *poly* near the *pc* should be added;
- the resistance of the *pc* should be added.

The Step 3 will provide a more accurate extraction, by including all these additional terms

Example 6



[4] R. L. Geiger, P. E. Allen, N. R. Strader, *VLSI*, McGraw-Hill, 1990, p.80

From the .EXT file (we use the technology file modified as in Step 1, in its version with the ss layer) it results:

```
node "a_n21_31#" 0 0 -21 31 sb 0 0 0 0 0 0 0 0 0 0 0 0
node "A" 26 0 -27 33 pc 0 0 0 0 360 154 50 40 0 0 0 0
equiv "A" "B"
node "a_n21_38#" 0 0 -21 38 sb 0 0 0 0 0 0 0 0 0 0 0 0
node "SS" 0 0 -35 13 ss 0 0 0 0 0 0 0 0 0 0 0 0
cap "A" "SS" 5440.2
```

and the two nodes A and B are "short circuited" (equiv "A" "B").

After the further modifications of the technology file, it results from the .EXT file:

```
node "B" 2 0 24 17 pc 0 0 0 0 30 17 25 20 0 0 0 0
node "a_n21_31#" 0 0 -21 31 sb 0 0 0 0 0 0 0 0 0 0 0 0
node "a_n21_33#" 18 0 -21 33 pres 0 0 0 0 300 120 0 0 0 0 0 0
node "A" 2 0 -27 33 pc 0 0 0 0 30 17 25 20 0 0 0 0
node "a_n21_38#" 0 0 -21 38 sb 0 0 0 0 0 0 0 0 0 0 0 0
node "SS" 0 0 -35 13 ss 0 0 0 0 0 0 0 0 0 0 0 0
cap "A" "SS" 632.1
cap "B" "SS" 632.1
cap "a_n21_33#" "SS" 4176
fet polyres -21 33 -20 34 300 130 "SS" "a_n21_33#" 0 "R1" "A" 5 0 "B" 5 0
```

and the resistor is extracted as a FET while the parasitic capacitances of the two contacts (from *pc* and *poly*) are correctly extracted.

From the .spice file (*-mm* and *-mn* options) it results:

```
        R1 A B RPOLY 12.00
        * R1=1080.0 (width=1.50U)
        C1 B SS 0.6F
        C2 A SS 0.6F
        ************************************************************
        ** Model Definitions for PSPICE
        ************************************************************
        .MODEL RPOLY RES (R=90.000)
```

and the number of squares SQ is correctly extracted, apart the factor 0.55 which must be attributed to the corner.

Or, in the distributed case (*-mm, -mn* and *-d* options)

```
        XR1 A B SS D_RPOLY params: SQ=12.00 A=27.00
        * R1=1080.0 (width=1.50U)
        C1 B SS 0.6F
        C2 A SS 0.6F
        .SUBCKT D_RPOLY 1 2 3 params: SQ={SQ} A={A}
```

```
XR1 1 4 3 E_RPOLY params: SQ={SQ/5} A={A/5}
XR2 4 5 3 E_RPOLY params: SQ={SQ/5} A={A/5}
XR3 5 6 3 E_RPOLY params: SQ={SQ/5} A={A/5}
XR4 6 7 3 E_RPOLY params: SQ={SQ/5} A={A/5}
XR5 7 2 3 E_RPOLY params: SQ={SQ/5} A={A/5}
.ENDS
**********************************************************
** Unit element model for RPOLY
**********************************************************
.SUBCKT E_RPOLY 1 2 3 params: SQ={SQ} A={A}
R1 1 2 RPOLY {SQ}
C1 1 3 CAPP {A/2}
C2 2 3 CAPP {A/2}
.ENDS
**********************************************************
** Model Definitions for PSPICE
***********************************************************
.MODEL RPOLY RES (R=90.000)
.MODEL CAPP CAP (C=0.088FF)
```

and also the area A of *pres* for the evaluation of the parasitic capacitance associated to *pres* is correctly extracted.

In this example, the limits in the accuracy of the extraction are the ones previously discussed.

Step 3 - Corrections and Improvements (ext2spice.corr)

The Ext2spice.corr program provides the correction and improvement of the spice netlist generated by the Ext2spice program. In particular:

- eliminates, if present, all the capacitors the value of which is 0,0F [5];
- calculates the value of **resistors**, both in the non-distributed and in the distribted case, by including in the evaluation the corners, the resistance of the poly adiacent the contact, the resistance of the contact, the perimeter component of the parasitic capacitance and the difference between the real and the drawn width; for each resistor the result of these calculations is also reported as a comments;
- converts the NRG, NRD and NRS parameters of the **FETs** into RG, RD and RS parameters, providing also a more accurate approximation, by including the resistance of the contacts and a separate coefficient for the resistance per square of polysilicon, ndiffusion and pdiffusion;
- locates the **linear capacitors** and provides these capacitors with a model that includes their dependence on the voltage and on the temperature.

Finally:

- by the introduction of an additional layerfor each metal and by some further modification of the Technology file and of the Ext.defs file, it provide the extraction of the **metal lines as resistors**, including all the related parasitic capacitances.

Ext2spice.corr

The code of the Ext2spice.corr program is reported in the Appendix F. The program can be invoked in cascade to Ext2spice by using the following script:

```
#!/bin/csh
rm -f $1.cir
~ ext2spice -mm -mn  -c .001 -g -d -D -S $1
cp $1.spice $1.spice.old
~ ext2spice.corr $1.spice.old > $1.spice
```

The old .spice file is saved as *nomefile.spice.old*

[5] This function is necessary because is some cases the 0 value capacitors are not filtered by Ext2spice.

Improvements in the extraction of the Resistors

(a) - Corners

In order to take into account in the calculus of the value of the resistor all the corners present along *pres*, the number of corners of each resistor must be known. This number can be included in the label of each resistor. When Ext2spice.corr finds a resistor with a label R*xx*_L*n* (the label R*xx*_L*n*^ must be attached to *pres*), extracts *n* as the number of corners for that resistor. If the term _L*n* is omitted from the label, the number of corners of that resistor is assumed to be zero. Ext2spice.corr provides also the necessary modifications in the .spice file in order to to take into account this new parameter, both in the case of the non-distributed and distributed model.

(b) - Resistance of the contact and of the poly adjacent to the contact

The resistance of the contact *pc* and of the poly adjacent the contact are included in the calculus. For the adjacent poly, the minimum length 1λ is assumed (plus 1λ for the distance between the real contact and the poly). The value of the resistance of the poly contact *PC* [Ohm] and of the resistance per square of the poly *RSQPOLY* [Ohm/square] must be given at the beginning of the Ext2spice.corr code. These values can be found in the General Process Specifications, in the Technology File of the Process or in the Mosis Parametric Test Results.

(c) - Perimeter component of the parasitic capacitance

In the case of the distributed model, the parasitic capacitance associated to the resisor is extracted. But Ext2spice provides only the area component of this capacitance. In order to extract also the perimeter component, the .MODEL lines are replaced with .PARAM lines and the capacitance is calculated according to the values of the parameters. The value of the areacap *RACAP* [F/m^2] and perimcap *RPCAP* [F/m] must be given at the beginning of the Ext2spice.corr code. These values can be found in the General Process Specifications, in the Technology File of the Process or in the Mosis Parametric Test Results.

Attention must be paid when the additional ss layer is used and *pres* lies is in a well, because the parasitic capacitance is calculated from *pres* to the layer ss and not from *pres* to the well, which is an error. The problem is that when the MAGIC extractor must choose between ss and well, it chooses ss (i.e. the layer with the lower planeorder). The solution is to remove the ss layer from all the areas where *pres* is overlapped to a well.

(d) - Real and drawn width

Ext2spice.corr provides the correction through the parameter *RDW*, which is the delta-width. The value of *RDW* [m] must be given at the beginning of the Ext2spice.corr code. This value can be found in the General Process Specifications, in the Technology File of the Process or in the Mosis Parametric Test Results.

(e) - Calculus of the value of the resistance

The value of the resistance is finally calculated through the equation:

$$R = 2PC + 2RSQPOLY\frac{2LAMBDA}{W} + \frac{RSQ(SQ - 0.45NL)}{1 - \frac{RDW}{W}} \qquad (6)$$

where W is the width of *pres* and NL is the number of corners. In the non-distributed case W is not available and a minimum width *WMIN* is assumed.
The values of *LAMBDA* [m], *RSQ* [Ohm/square] and *WMIN* [m] must be given at the beginning of the Ext2spice.corr code. These values can be found in the General Process Specifications, in the Technology File of the Process or in the Mosis Parametric Test Results.

Example 7

After the relabeling of the resistance of the **Example 6** as R1_L1^, it follows from Ext2spice.corr in the non-distributed case:

```
R100 A B 1056.3
C1 B 1 0.6F
C2 A 1 0.6F
```

and in the distributed case:

```
XR1 A B 1 D_RPOLY params: SQ=12.00 A=27.00 NL=1
* XR1=1056.3 (W=1.5e-06, Weff=1.5e-06, Cpar=6.687e-15)
C1 B 1 0.6F
C2 A 1 0.6F
********************************************************
* Global Node Definitions
********************************************************
Vsubstr  $G_Vsub! 0 DC 0V
********************************************************
* Distributed Model Definitions
********************************************************
** Multiple element model for RPOLY
********************************************************
.SUBCKT D_RPOLY 1 2 3 params: SQ={SQ} A={A} NL={NL}
XR1 1 4 3 E_RPOLY params: SQ = {SQ/5} A = {A/5}
XR2 4 5 3 E_RPOLY params: SQ = {SQ/5} A = {A/5}
XR3 5 6 3 E_RPOLY params: SQ = {SQ/5} A = {A/5}
XR4 6 7 3 E_RPOLY params: SQ = {SQ/5} A = {A/5}
XR5 7 2 3 E_RPOLY params: SQ = {SQ/5} A = {A/5}
.ENDS
********************************************************
** Unit element model for RPOLY
********************************************************
.SUBCKT E_RPOLY 1 2 3 params: SQ={SQ} A={A} NL={NL}
R1 1 2 {2*PC+2*RSQPOLY*2*LAMBDA/SQRT(A*1p/SQ)+RSQ*(SQ-NL*0.45)/(1-
                                                 RDW/SQRT(A*1p/SQ))}
C1 1 3 {RACAP*A*1p/2+RPCAP/2*2*SQRT(A*1p*SQ)}
C2 2 3 {RACAP*A*1p/2+RPCAP/2*2*SQRT(A*1p*SQ)}
.ENDS
********************************************************
```

```
.PARAM RSQPOLY=3.5 PC=7
************************************************************
.PARAM RSQ=90 RDW=0 SQ=1 A=1p NL=0
.PARAM RACAP=8.5e-05 RPCAP=1.22e-10
```

Improvements in the extraction of the FETs

In order to give a better approximation of the parasitic resistances, their values are directly calculated by Ext2spice.corr. On each FET line, the new expressions for RG, RD and RS replaces the old expressions for NRG, NRD, NRS.

(a) - Parasitic resistance RG

The following equation is used for the evaluation of RG:

$$\mathrm{RG} = \mathrm{RSQPOLY}\left(\frac{1}{\mathrm{M}\cdot\mathrm{NRG}} + \frac{4\mathrm{LAMBDA}}{\mathrm{L}}\right)\frac{1}{4} + \mathrm{PC}\frac{1}{2}\frac{\mathrm{CLT}\cdot\mathrm{LAMBDA}}{\mathrm{L}} \tag{7}$$

The equation follows by considering the gate of a finger connected at both sides thus giving two resistors (remember that Ext2spice evaluates NRG in the wrong direction and it divides NRG per finger by M) in parallel for each finger. The term $4\lambda/L$ takes into account the resistance of the poly external to the active area. The last term approximates the contact resistance by using the parameter CLT, which is the number of contacts per lambda. The calculus is subject to the approximation made on the original NRG. The minimum dimension 1λ for the *poly* adjacent to the *pc* is assumed. The contact resistance results over-estimated for small W. The value of *CLT* must be given at the beginning of the Ext2spice.corr code and is set by the design rules.

(b) - Parasitic resistances RD and RS

The following equations are used to approximate the values of RD and RS.

For single finger n-chennal FETS:

$$\mathrm{RD} = \mathrm{NRSQ}\left(\mathrm{NRD} - \frac{3\cdot\mathrm{LAMBDA}}{\mathrm{W}}\right) + \mathrm{NDC}\frac{2\cdot\mathrm{CLT}\cdot\mathrm{LAMBDA}}{\mathrm{W}} \tag{8a}$$

$$\mathrm{RS} = \mathrm{NRSQ}\left(\mathrm{NRS} - \frac{3\cdot\mathrm{LAMBDA}}{\mathrm{W}}\right) + \mathrm{NDC}\frac{2\cdot\mathrm{CLT}\cdot\mathrm{LAMBDA}}{\mathrm{W}} \tag{8b}$$

and for single finger p-channel FETs:

$$\mathrm{RD} = \mathrm{PRSQ}\left(\mathrm{NRS} - \frac{3\cdot\mathrm{LAMBDA}}{\mathrm{W}}\right) + \mathrm{PDC}\frac{2\cdot\mathrm{CLT}\cdot\mathrm{LAMBDA}}{\mathrm{W}} \tag{8c}$$

$$\mathrm{RS} = \mathrm{PRSQ}\left(\mathrm{NRS} - \frac{3\cdot\mathrm{LAMBDA}}{\mathrm{W}}\right) + \mathrm{PDC}\frac{2\cdot\mathrm{CLT}\cdot\mathrm{LAMBDA}}{\mathrm{W}} \qquad (8d)$$

For multifinger n-chennal FETS:

$$\mathrm{RD} = \mathrm{NRSQ}\left(\mathrm{NRD}\cdot\mathrm{M}^2 - \frac{\mathrm{M}\cdot\mathrm{LAMBDA}}{\mathrm{W}} - \frac{2\cdot\mathrm{LAMBDA}}{\mathrm{W}}\right)\frac{1}{\mathrm{M}} + \mathrm{NDC}\frac{2\cdot\mathrm{CLT}\cdot\mathrm{LAMBDA}}{\mathrm{W}} \qquad (9a)$$

$$\mathrm{RS} = \mathrm{NRSQ}\left(\mathrm{NRS}\cdot\mathrm{M}^2 - \frac{\mathrm{M}\cdot\mathrm{LAMBDA}}{\mathrm{W}} - \frac{2\cdot\mathrm{LAMBDA}}{\mathrm{W}}\right)\frac{1}{\mathrm{M}} + \mathrm{NDC}\frac{2\cdot\mathrm{CLT}\cdot\mathrm{LAMBDA}}{\mathrm{W}} \qquad (9b)$$

and for multifinger p-channel FETs:

$$\mathrm{RD} = \mathrm{PRSQ}\left(\mathrm{NRS}\cdot\mathrm{M}^2 - \frac{\mathrm{M}\cdot\mathrm{LAMBDA}}{\mathrm{W}} - \frac{2\cdot\mathrm{LAMBDA}}{\mathrm{W}}\right)\frac{1}{\mathrm{M}} + \mathrm{PDC}\frac{2\cdot\mathrm{CLT}\cdot\mathrm{LAMBDA}}{\mathrm{W}} \qquad (9c)$$

$$\mathrm{RS} = \mathrm{PRSQ}\left(\mathrm{NRS}\cdot\mathrm{M}^2 - \frac{\mathrm{M}\cdot\mathrm{LAMBDA}}{\mathrm{W}} - \frac{2\cdot\mathrm{LAMBDA}}{\mathrm{W}}\right)\frac{1}{\mathrm{M}} + \mathrm{PDC}\frac{2\cdot\mathrm{CLT}\cdot\mathrm{LAMBDA}}{\mathrm{W}} \qquad (9d)$$

The M*λ/W-2*λ/W terms take into account the resistance of the diffusion part of the real contact. The XDC/(W/CLT/λ) term approximates the contact resistance and XDC must be NDC for n-channel and PDC for p-channel. The calculus is subject to the approximation on the original NRS (or NRD). The minimum distance between the channel and *ndc* (or *pdc*) is assumed. The contact resistance results overestimated for small L. Note that, during the simulation, SPICE considers the RS, RG, RD as values per finger, dividing these values by M for the simulation of the overall FET, which is the correct approach.
The values of *NRSQ* and *PRSQ* [Ohm/square], respectively the resistance per square of the ndiffusion and of the pdiffusion, as well as the values of *NDC* and *PDC* [Ohm], respectively the resistance of the contacts *ndc* and *pdc*, must be given at the beginning of the Ext2spice.corr code. These values can be found in the General Process Specifications, in the Technology File of the Process or in the Mosis Parametric Test Results.

Example 8

For the FET of **Example 4** it follows:

```
Mx B C A SS CMOSN W=3.00U L=1.20U AD=6.30P PD=10.20U AS=6.30P PS=10.20U
+ RG=7.4375 RD=3 RS=3
C1 B SS 1.8F
C2 C SS 1.1F
```

and for the FET of **Example 5** it follows:

```
M1 A C B SS CMOSN M=3 W=3.00U L=0.60U AD=3.30P PD=6.20U AS=6.04P PS=7.03U
```

```
+ RG=14.6667 RD=4.48333 RS=5.23333
M2 A2 C2 B SS CMOSN M=2 W=6.00U L=0.60U AD=5.40P PD=7.80U AS=12.09P PS=14.06U
+ RG=19.25 RD=2.1 RS=2.6
C1 A C 0.2F
C3 A2 B 0.4F
C4 C2 SS 4.1F
C5 B SS 0.4F
C6 B C 0.2F
C7 A2 SS 0.1F
C8 A B 0.6F
C9 C SS 5.6F
C10 B C2 0.2F
C11 A SS 0.2F
```

Improvements in the extraction of the Linear Capacitors

The improvement in the extraction of the linear capacitors consists in the addition of the dependence of the capacitance on the voltage and on the temperature. In order to achieve this result, all the linear capacitors must be modeled by using a .MODEL statement in the .spice file. Till now it was not possible to distinguish between parasitic and linear capacitors. But adding the following line in the *extract* section of the technology file:

```
fet  wcap cwnsd 1 wcap $$G_Vsub! 0 0
```

each linear capacitor will be represented both as *cap* and as *fet* element in the .EXT file.

If the following lines are added in the Ext.defs support file,

```
def wcap       LCAP       C c
distrib wcap      0.475  0.0 0
```

Ext2spice will extract, for each linear capacitor, a "duplicated" capacitor (apart the value) and models each duplicate with LCAP.

Ext2spice.corr deletes, by comparison, all these "duplicates" and models the corresponding original capacitors (the value of which is more accurate) through the model LCAP. The original and correct values of the capacitors are preserved and the .MODEL line for LCAP is added. The values of the voltage parameters *VC1* and *VC2* and of the temperature parameters *TC1* and *TC2* used in the .MODEL statement can be found in the General Process Specifications or in the Mosis Parametric Test Results.

Example 9

For the linear capacitor of **Example 3** it follows:

```
C2 A B LCAP 2097.8F
*************************************************************
```

```
* Global Node Definitions
**********************************************************
.MODEL LCAP CAP (C=1 VC1=0.001 VC2=0.0005 TC1=0 TC2=0 )
```

Extraction of the resistance of the Metal lines

The extraction of the resistance of the metal lines requires that, as in the case of poly resistors, a part or all the line becomes visible as a *fet* to the MAGIC extractor. A possible and effective approach is the introduction of an additional layer for each metal, which must be invisible at the .CIF extraction level and which represents the active area of this "fet". In order to preserve all the parasitic capacitances related to the lines, the active area of the "fet" must be as small as possible, i.e. must be a square along the line. The resistance of the line is then extracted by evaluating the number of squares of the "drain" and of "source" of this “fet” (which equals with good approximation the number of squares of the line). The use of more squares suitably distributed along the line allows a more accurate extraction.

In order to use the additional layers, the following changes must be made in the Technology File.

In the *types* section, three additional layers (*m1res*, *m2res* and *m3res*) for the three metals (*m1*, *m2* and *m3*) must be added:

```
metal1  m1res,m1r
metal2  m2res,m2r
metal3  m3res,m3r
```

In the *styles* section, to "see" the additional layers at the MAGIC layout level; the lines

```
m1res  20
m1res  12
m1res  13
m2res  21
m2res  12
m2res  13
m3res  22
m3res  12
m3res  13
```

must be added.

In the *connect* section, to set the connections, the lines:

```
m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1
                        m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1
m2,m2c/m2,m3c/m2,m3c/m2  m2,m2c/m2,m3c/m2,m3c/m2
m3,m3c/m3  m3,m3c/m3
```

must be replaced with the lines:

```
m1,m1res,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1
                             ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1
m2,m2res,m2c/m2,m3c/m2,m3c/m2  m2c/m2,m3c/m2,m3c/m2
m3,m3res,m3c/m3  m3c/m3
```

In the *cifoutput* section, to let the layer be invisible at the .CIF level (i.e to let it be "like" the original metal):

```
layer CMF m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1
labels m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1

layer CMS m2,m2c/m2,m3c/m2,m3c/m2
labels m2,m2c/m2,m3c/m2,m3c/m2

layer CMT m3,m3c/m3
labels m3,m3c/m3
```

must be replaced with the lines:

```
layer CMF m1,m1res,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1
labels m1,m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1

layer CMS m2,m2res,m2c/m2,m3c/m2,m3c/m2
labels m2,m2,m2c/m2,m3c/m2,m3c/m2

layer CMT m3,m3res,m3c/m3
labels m3,m3,m3c/m3
```

In the *extract* section, to enable the extraction of the "fet" diffusion areas, the *resistclass* lines must be modified as follows (this change does not affect the calculus of the real FET areas, if the Ext.defs support file is modified as described later):

```
# REPLACED resist (ndiff,anres,ndc/a,nsd,nsc/a)/active 1900
 resist ((ndiff,ndc/a)/active,(pdiff,pdc/a)/active) 0
# REPLACED resist (pdiff,apres,pdc/a,psd,psc/a)/active 2000
 resist (m1/metal1,m2/metal2,m3/metal3) 0
```

while the old *resistclass* lines containing *m1*, *m2* and *m3* must be deleted.

Finally, to enable the "fet" extraction, the following lines

```
fet  m1res m1,ndc/m1,nsc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1 2 m1res $$G_Vsub! 0 0
fet  m2res m2,m2c/m2,m3c/m2,m3c/m2 2 m2res $$G_Vsub! 0 0
fet  m3res m3,m3c/m3 2 m3res $$G_Vsub! 0 0
```

must be added. Note that all the contacts have been included in the evaluation of the areas of the fets for the extraction of the resistances.

In order to extract the new "fet" as a resistance, the following lines must be added in the Ext.defs support file. Basically the idea is to extract the areas of the new "fet" by considering these as p-channel FETs. The areas of the real p-channel FETs is extracted by considering these as n-channel FETs. This approach, relevant only at the Ext2spice level, has no consequence for the final correct extraction of the areas of all real FETs.

The line:

```
def pfet  CMOSP  M p
```

must be replaced with the line:

```
def pfet  CMOSP  M n
```

and the lines:

```
def m1res       RM1   M p
def m2res       RM2   M p
def m3res       RM3   M p
```

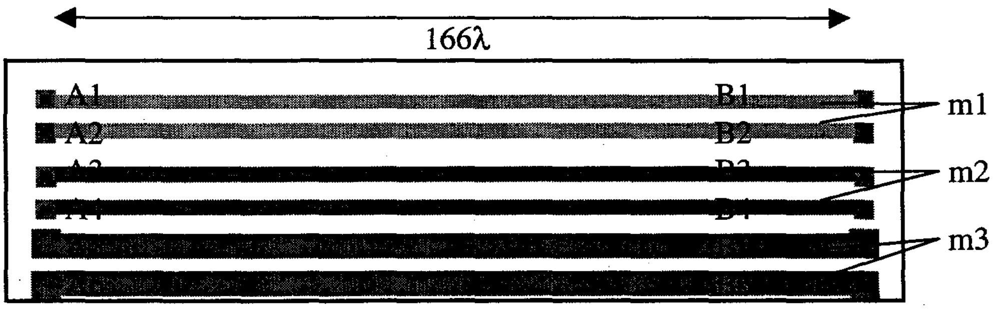
must be added.

Now Ext2spice.corr provides the accurate exraction of the resistances associated to the metal lines from the "drain" and "source" areas of these "fets".

The values of *RSQM1*, *RSQM2*, and *RSQM3* [Ohm/square], respectively the resistance per square of *m1*, *m2* and *m*3, must be given at the beginning of the Ext2spice.corr code. These values can be found in the General Process Specifications, in the Technology File of the Process or in the Mosis Parametric Test Results.

Drawback of this technique is that the line, at the MAGIC layout level, doesn't appear connected so that the check of the connections must be made before this modification, or the lines in the *connect* section must be suitably switched.

Example 10



In this example we have three couples of minimum wide lines in the three different metals (*m1*, *m2*, *m3*). A small square of the correct type (respectively *m1res*, *m2res*, *m3res*) has been introduced in the middle of each line, thus dividing the line in two parts. Tthe square is not visible here because the picture is extracted from the .CIF file, and, as required, the layers (*m1res*, *m2res*, *m3res*) are not visible at the .CIF level. From Step 2, with regards to the fet elements, it follows:

```
M1 A1 4 B1 $G_Vsub! RM1 W=0.90U L=0.00U AD=21.87P PD=50.40U AS=22.14P PS=51.00U
+ NRG=1.00 NRD=27.00 NRS=27.33
M2 A2 7 B2 $G_Vsub! RM1 W=0.90U L=0.00U AD=21.87P PD=50.40U AS=22.14P PS=51.00U
+ NRG=1.00 NRD=27.00 NRS=27.33
M3 A3 10 B3 $G_Vsub! RM2 W=0.90U L=0.00U AD=21.87P PD=50.40U AS=22.14P PS=51.00U
+ NRG=1.00 NRD=27.00 NRS=27.33
M4 A4 13 B4 $G_Vsub! RM2 W=0.90U L=0.00U AD=21.87P PD=50.40U AS=22.14P PS=51.00U
+ NRG=1.00 NRD=27.00 NRS=27.33
M5 A5 16 B5 $G_Vsub! RM3 W=1.50U L=0.00U AD=34.65P PD=55.80U AS=40.50P PS=63.60U
+ NRG=1.00 NRD=15.40 NRS=18.00
M6 A6 19 B6 $G_Vsub! RM3 W=1.50U L=0.00U AD=34.65P PD=55.80U AS=40.50P PS=63.60U
+ NRG=1.00 NRD=15.40 NRS=18.00
```

And from Step 3:

```
RM1 A1 B1 3.87333
RM2 A2 B2 3.87333
RM3 A3 B3 3.87333
RM4 A4 B4 3.87333
RM5 A5 B5 1.72
RM6 A6 B6 1.72
C2 A6 SS 1.0F
C5 B5 SS 1.1F
C8 A5 SS 1.0F
C9 B2 SS 2.7F
C10 B6 B5 0.8F
C11 B2 B1 0.4F
C13 A2 SS 2.7F
C14 A6 A5 0.7F
C16 B4 SS 1.4F
C19 A2 A1 0.4F
C20 B4 B3 0.4F
C21 A4 SS 1.4F
C22 SS B1 2.2F
C23 B3 SS 1.4F
C24 B6 SS 1.1F
C25 A4 A3 0.4F
C26 A3 SS 1.4F
C28 SS A1 2.2F
```

The accuracy os this evaluation, which includes the parasitic capacitances between the lines lying in the same plane, can be easily verified.

Conclusions

With a relatively simple modification of the Technology File (see Appendix G for a summary of the modifications), and with the use of the Magic Extract function, of the Ext2spice program (and its support file Ext.defs) and of the Ext2spice.corr program, it is possible to extract the SPICE netlist from a MAGIC based layout with an improved accuracy. The limit in this approach is that every technology requires the modification of the corresponding Technology File (together with some minor changes in the two programs). The extraction of the capacitances is strongly dependent on the accuracy of the parameters from the Technology File, which must be verified by the simulations (for example by using the Maxwell simulator) and updated by using the beta versions released by MOSIS.

Appendix A - Connect and extract sections

```
connect
  nwell,cwell,nsc,cwc,nsd,cwnsd,cwc/a  nwell,cwell,nsc,cwc,nsd,cwnsd,cwc/a
  pwell,psc,psd  pwell,psc,psd
  m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1
                       m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1
  m2,m2c/m2,m3c/m2,m3c/m2  m2,m2c/m2,m3c/m2,m3c/m2
  m3,m3c/m3  m3,m3c/m3
  ndiff,anres,ndc/a  ndiff,anres,ndc/a
  pdiff,apres,pdc/a  pdiff,apres,pdc/a
  ndiff,anres,nsd  ndc/a,nsc/a,psd,psc/a
  pdiff,apres,psd  pdc/a,psc/a,nsd,nsc/a
  poly,pres,wcap,nfet,wcap,pfet,pc/a  poly,pres,wcap,nfet,wcap,pfet,pc/a
  gc  poly,pres,wcap
  gv1  m1,m2
  gc  ndiff,anres,pdiff,apres,nsd,psd,m1
  m3,m3c/m3  m3,m3c/m3
  gv2  m2,m3
  pad  m1,m2,m3
end
```

```
extract
 style HP0.5um(hpcmos14tb)from:N81D
 cscale 1
 lambda 30
 step   100
 sidehalo 8
 planeorder well 0
 planeorder implant 1
 planeorder select 2
 planeorder active 3
 planeorder metal1 4
 planeorder metal2 5
 planeorder metal3 6
 planeorder oxide 7
 planeorder xp 8
 planeorder comment 9
 planeorder contact 10
 planeorder via1 11
 planeorder via2 12

 resist (ndiff,anres,ndc/a,nsd,nsc/a)/active 1900
 resist (pdiff,apres,pdc/a,psd,psc/a)/active 2000
 resist (poly,pres,wcap,pc/a,wcap,pc/a,nfet,pfet)/active 1800
 resist (m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1,m2c/m1)/metal1  70
 resist (m2,m2c/m2,m3c/m2,m3c/m2)/metal2  70
 resist (m3,m3c/m3,pad)/metal3  50

 contact ndc 4 2000
contact pdc 4 2000
contact pc 4 1700
contact m2c 4 700
contact m3c 4 360


#wellcap
 overlap wcap capwell  207.900

#nwell,cwell,pwell
 areacap (nwell,cwell)/well  8.460
 areacap (cwell)/well  8.460

#ndiff
 areacap (anres)/active  48.330
 overlap (anres)/active ~space/w  48.330
 perimc (anres)/active ~(anres)/active  67.800
 sideoverlap (anres)/active ~(anres)/active ~space/w  67.800
# MODEL HANDLES THIS: areacap (ndiff,anres,ndc/a)/active  48.330
# MODEL HANDLES THIS: overlap (ndiff,anres,ndc/a)/active ~space/w  48.330
# MODEL HANDLES THIS: perimc (ndiff,anres,ndc/a)/active ~(ndiff,anres,ndc/a,nfet,pfet)/active  67.800
# MODEL HANDLES THIS: sideoverlap (ndiff,anres,ndc/a)/active ~(ndiff,anres,ndc/a,nfet,pfet)/active
                    ~space/w  67.800

#pdiff
 areacap (apres)/active  84.600
 overlap (apres)/active ~space/w  84.600
 perimc (apres)/active ~(apres)/active  67.500
 sideoverlap (apres)/active ~(apres)/active ~space/w  67.500
# MODEL HANDLES THIS: areacap (pdiff,apres,pdc/a)/active  84.600
# MODEL HANDLES THIS: overlap (pdiff,apres,pdc/a)/active ~space/w  84.600
# MODEL HANDLES THIS: perimc (pdiff,apres,pdc/a)/active ~(pdiff,apres,pdc/a,nfet,pfet)/active  67.500
# MODEL HANDLES THIS: sideoverlap (pdiff,apres,pdc/a)/active ~(pdiff,apres,pdc/a,nfet,pfet)/active
                    ~space/w  67.500

#polycap

#poly
 sidewall (poly,pres,wcap,pc/a,wcap,pc/a)/active ~(poly,pres,wcap,pc/a,wcap,pc/a)/active
                    ~(poly,pres,wcap,pc/a,wcap,pc/a)/active (poly,pres,wcap,pc/a,wcap,pc/a)/active  2.136
 areacap (poly,pres,wcap,pc/a,wcap,pc/a)/active  7.920
 overlap (poly,pres,pc/a,pc/a)/active ~space/w  7.920
 perimc (poly,pres,wcap,pc/a,wcap,pc/a)/active ~(poly,pres,wcap,pc/a,wcap,pc/a)/active  15
 sideoverlap (poly,pres,wcap,pc/a,wcap,pc/a)/active ~(poly,pres,wcap,pc/a,wcap,pc/a)/active ~space/w  15
# MODEL HANDLES THIS: overlap (nfet)/active (ndiff,anres,ndc/a)/active 324.360
# MODEL HANDLES THIS: sideoverlap (nfet)/active ~(nfet)/active (ndiff,anres,ndc/a)/active  15
# MODEL HANDLES THIS: overlap (pfet)/active (pdiff,apres,pdc/a)/active 308.160
# MODEL HANDLES THIS: sideoverlap (pfet)/active ~(pfet)/active (pdiff,apres,pdc/a)/active  15

#poly2

#metal1
 sidewall (m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1
                    ~(m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1
                    ~(m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1
                    (m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1  6.690
 areacap (m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1  2.700

#metal-sub blocked by ~space/a,~space/c
 overlap (m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1 ~space/w  2.700
                ~space/a
 perimc (m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1
                ~(m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1  9.300
 sideoverlap (m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1
                ~(m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1 ~space/w
                9.300 ~space/a

#metal-diff blocked by ~space/c
 overlap (m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1
                (ndiff,anres,ndc/a)/active  2.790
 sideoverlap (m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1
                ~(m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1
                (ndiff,anres,ndc/a)/active  9.300
 overlap (m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1
                (pdiff,apres,pdc/a)/active  2.790
 sideoverlap (m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1
                ~(m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1
                (pdiff,apres,pdc/a)/active  9.300

#metal-polycap blocked by poly

#metal-poly blocked by poly2
 overlap (m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1
                (poly,pres,wcap,pc/a,wcap,pc/a,nfet,pfet)/active  5.400
 sideoverlap (m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1
                ~(m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1
                (poly,pres,wcap,pc/a,wcap,pc/a,nfet,pfet)/active  15
 sideoverlap (poly,pres,wcap,pc/a,wcap,pc/a,nfet,pfet)/active
                ~(poly,pres,wcap,pc/a,wcap,pc/a,nfet,pfet)/active
                (m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1  15

#metal-poly2 not blocked

#metal2
 sidewall (m2,m2c/m2,m3c/m2)/metal2 ~(m2,m2c/m2,m3c/m2)/metal2 ~(m2,m2c/m2,m3c/m2)/metal2
                (m2,m2c/m2,m3c/m2)/metal2  9.090
 areacap (m2,m3c/m2)/metal2  1.260

#metal2-sub blocked by ~space/a,~space/m1,poly2/polycap
 overlap (m2,m3c/m2)/metal2 ~space/w  1.260 ~space/a,~space/m1
 perimc (m2,m2c/m2,m3c/m2)/metal2 ~(m2,m2c/m2,m3c/m2)/metal2  5.100
 sideoverlap (m2,m2c/m2,m3c/m2)/metal2 ~(m2,m2c/m2,m3c/m2)/metal2 ~space/w  5.100
                ~space/a,~space/m1
 overlap (m2,m3c/m2)/metal2 (ndiff,anres,ndc/a)/active  1.260 ~space/m1
 sideoverlap (m2,m2c/m2,m3c/m2)/metal2 ~(m2,m2c/m2,m3c/m2)/metal2 (ndiff,anres,ndc/a)/active
                5.100 ~space/m1
 overlap (m2,m2c/m2,m3c/m2)/metal2 (pdiff,apres,pdc/a)/active  1.260 ~space/m1
 sideoverlap (m2,m2c/m2,m3c/m2)/metal2 ~(m2,m2c/m2,m3c/m2)/metal2 (pdiff,apres,pdc/a)/active
                5.100 ~space/m1

#metal2-polycap by ~space/m1,poly
```

```
#metal2-poly blocked by ~space/m1,poly2
 overlap (m2,m3c/m2)/metal2 (poly,pres,wcap,pc/a,wcap,pc/a,nfet,pfet)/active   1.620 ~space/m1
 sideoverlap (m2,m2c/m2,m3c/m2)/metal2 ~(m2,m2c/m2,m3c/m2)/metal2
                    (poly,pres,wcap,pc/a,wcap,pc/a,nfet,pfet)/active  6.300 ~space/m1
 sideoverlap (poly,pres,wcap,pc/a,wcap,pc/a,nfet,pfet)/active
                    ~(poly,pres,wcap,pc/a,wcap,pc/a,nfet,pfet)/active (m2,m2c/m2,m3c/m2)/metal2   6.300
                    ~space/m1

#metal2-poly2 blocked by ~space/m1

#M2->M1
 overlap (m2,m3c/m2)/metal2
                    (m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1   4.050
 sideoverlap (m2,m2c/m2,m3c/m2)/metal2 ~(m2,m2c/m2,m3c/m2)/metal2
                    (m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1  11.700
 sideoverlap (m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1
                    ~(m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1
                    (m2,m2c/m2,m3c/m2)/metal2  11.700

#metal3
 sidewall (m3,m3c/m3,pad)/metal3 ~(m3,m3c/m3,pad)/metal3 ~(m3,m3c/m3,pad)/metal3
                    (m3,m3c/m3,pad)/metal3  12.900
 areacap (m3,pad)/metal3   0.810

#metal3-sub blocked by ~space/a,~space/m1,~space/m2,poly2/polycap
 overlap (m3,pad)/metal3 ~space/w   0.810 ~space/a,~space/m1,~space/m2
 perimc (m3,m3c/m3,pad)/metal3 ~(m3,m3c/m3,pad)/metal3   3.300
 sideoverlap (m3,m3c/m3,pad)/metal3 ~(m3,m3c/m3,pad)/metal3 ~space/w   3.300
                    ~space/a,~space/m1,~space/m2

#metal3-*diff blocked by ~space/m1,~space/m2,poly2/polycap
 overlap (m3,pad)/metal3 (ndiff,anres,ndc/a)/active   0.810 ~space/m1,~space/m2
 sideoverlap (m3,m3c/m3,pad)/metal3 ~(m3,m3c/m3,pad)/metal3 (ndiff,anres,ndc/a)/active   3.300
                    ~space/m1,~space/m2
 overlap (m3,pad)/metal3 (pdiff,apres,pdc/a)/active   0.810 ~space/m1,~space/m2
 sideoverlap (m3,m3c/m3,pad)/metal3 ~(m3,m3c/m3,pad)/metal3 (pdiff,apres,pdc/a)/active   3.300
                    ~space/m1,~space/m2

#metal3-polycap by ~space/m1,~space/m2,poly

#metal3-poly blocked by ~space/m1,~space/m2,poly2
 overlap (m3,pad)/metal3 (poly,pres,wcap,pc/a,wcap,pc/a,nfet,pfet)/active   0.990 ~space/m1,~space/m2
 sideoverlap (m3,m3c/m3,pad)/metal3 ~(m3,m3c/m3,pad)/metal3
                    (poly,pres,wcap,pc/a,wcap,pc/a,nfet,pfet)/active   3.600 ~space/m1,~space/m2
 sideoverlap (poly,pres,wcap,pc/a,wcap,pc/a,nfet,pfet)/active
                    ~(poly,pres,wcap,pc/a,wcap,pc/a,nfet,pfet)/active (m3,m3c/m3,pad)/metal3   3.600
                    ~space/m1,~space/m2

#metal3-poly2 blocked by ~space/m1

#M3->M1

#metal3-metal1 blocked by ~space/m2
 overlap (m3,pad)/metal3 (m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1
                    1.440 ~space/m2
 sideoverlap (m3,m3c/m3,pad)/metal3 ~(m3,m3c/m3,pad)/metal3
                    (m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1   5.400
                    ~space/m2
 sideoverlap (m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1
                    ~(m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1
                    (m3,m3c/m3,pad)/metal3   5.400 ~space/m2

#M3->M2
 overlap (m3,pad)/metal3 (m2,m2c/m2,m3c/m2)/metal2   4.230
 sideoverlap (m3,m3c/m3,pad)/metal3 ~(m3,m3c/m3,pad)/metal3 (m2,m2c/m2,m3c/m2)/metal2  10.800
 sideoverlap (m2,m2c/m2,m3c/m2)/metal2 ~(m2,m2c/m2,m3c/m2)/metal2 (m3,m3c/m3,pad)/metal3
                    10.800

#metal4

#fets

 fet  pfet pdiff,pdc 2 pfet Vdd! nwell 15 308
 fet  pfet pdiff,pdc 1 pfet Vdd! nwell 15 308

 fet  nfet ndiff,ndc 2 nfet Gnd! pwell 15 324
 fet  nfet ndiff,ndc 1 nfet Gnd! pwell 15 324

 fetresis  pfet linear        15658
 fetresis  pfet saturation      15658
 fetresis  nfet linear        4267
 fetresis  nfet saturation      4267

end
```

Appendix B - Linear capacitor fringing capacitance

The first step to determine the fringing component for the capacitance of interest is to check the MOSIS General Process Specifications and the MOSIS Parametric Test Results relative to the technology used. In our case both the General Process Specifications and the Parametric Test Results don't report the value for the required fringing component. The Parametric Test Results report only the fringing component from *poly* to N^+, as shown in Fig. B1 but without specifying if this value is relative to the gate oxide thickness (90Å for this process), to the linear capacitor oxide thickness (150Å for this process) or, with less probability, to the field oxide thickness (3500Å for this process). It can also be observed that the value is subject to a strong dispersion and that its average value 164.53 aF/μm is much higher than the value 15/03.=50 aF/μm reported in the technology file for the fringing component from *poly* to *well.* In order to get a more reliable value for the required fringing component, an investigation based on the geometrical parameters given by the General Process Specifications (i.e. *poly* thickness T and oxide thickness D in the case of interest) must be carried out.

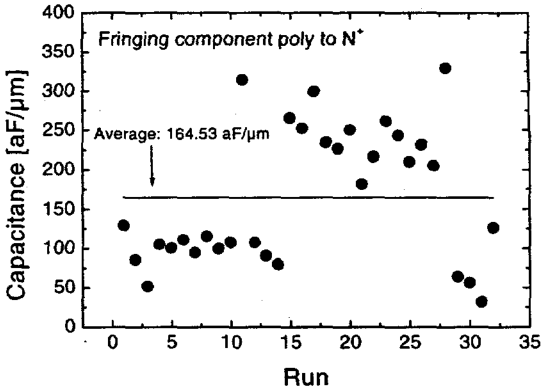


Fig. B1 - Fringing component of poly to N^+ from MOSIS Parametric Test Results.

The following equations for the evaluation of the fringing component from the two sides of a transmission line in the oxide can be found in the literature [6].

H. B. Bakoglu, *Circuits, Interconnections and Packaging for VLSI,* Addison-Wesley, 1990, p.138:

$$\varepsilon_{ox}\frac{2\pi}{\ln\left[1+\frac{2D}{T}\left(1+\sqrt{1+\frac{T}{D}}\right)\right]}-\frac{T}{2D} \tag{B1}$$

D. A. Pucknell , K. Eshraghian, *Basic VLSI Design,* Prentice Hall, 1988, p.91:

$$\varepsilon_{ox}\frac{2\pi}{\ln\left[1+\frac{2D}{T}\left(1+\sqrt{1+\frac{T}{D}}\right)\right]}-\frac{T}{4D} \tag{B2}$$

R. E. Collin, *Foundations for Microwave Engineering,* McGraw-Hill, 1992, p.150:

[6] For the dielectric constant of the oxide, the value $\varepsilon_{ox}=3.453133\cdot10^{-11}F/m=3.9\varepsilon_o$ is assumed.

$$\varepsilon_{ox}\left\{0.398\frac{T}{D}\left(1+\ln\frac{2D}{T}\right)+1.393D+0.667\ln\left[\frac{W}{D}+0.398\frac{T}{D}\left(1+\ln\frac{2D}{T}\right)+1.444\right]\right\} \quad \text{(B3)}$$

J. P. Uyemura, *Circuit Design for CMOS VLSI*, Kluwer Academic, 1992, p.242:

$$\varepsilon_{ox}\left[0.15\frac{W}{D}+2.8\left(\frac{T}{D}\right)^{0.222}\right] \quad \text{(B4)}$$

T. Sakurai, K. Tamaru, *Simple Formulas forTwo- and Three-Dimensional Capacitances*, IEEE Trans. Elec. Dev., Vol.30, No.2, Feb. 1983, p.183:

$$\varepsilon_{ox}\left[0.15\frac{W}{D}+2.8\left(\frac{T}{D}\right)^{0.222}+4.12\frac{D}{L}\left(\frac{T}{D}\right)^{0.728}\right] \quad \text{(B5)}$$

C. P. Yuan, T. N. Trick, *A Simple Formula for the Estimation of the Capacitance of Two-Dmensional Interconnects in VLSI Circuitss*, IEEE Elec. Dev. Lett., Vol.3, No.12, Dec. 1982, p.391:

$$\varepsilon_{ox}\left[0.15\frac{W}{D}+2.8\left(\frac{T}{D}\right)^{0.222}+4.12\frac{D}{L}\left(\frac{T}{D}\right)^{0.728}\right] \quad \text{(B6)}$$

In order to evaluate the accuracy of these equations, each one characterized by a limited range of application, a simulation has been carried out by using the recently purchased Maxwell 2D software from Ansoft. A comparison in the case of a W=1μm and W=10μm width line with a D=1000Å oxide tickness versus the line thickness T is shown in Fig.s B2. In the case of the Maxwell simulation, a term $\varepsilon_{ox}W/D$ (i.e. the area component) has been subtracted from the simulation result.

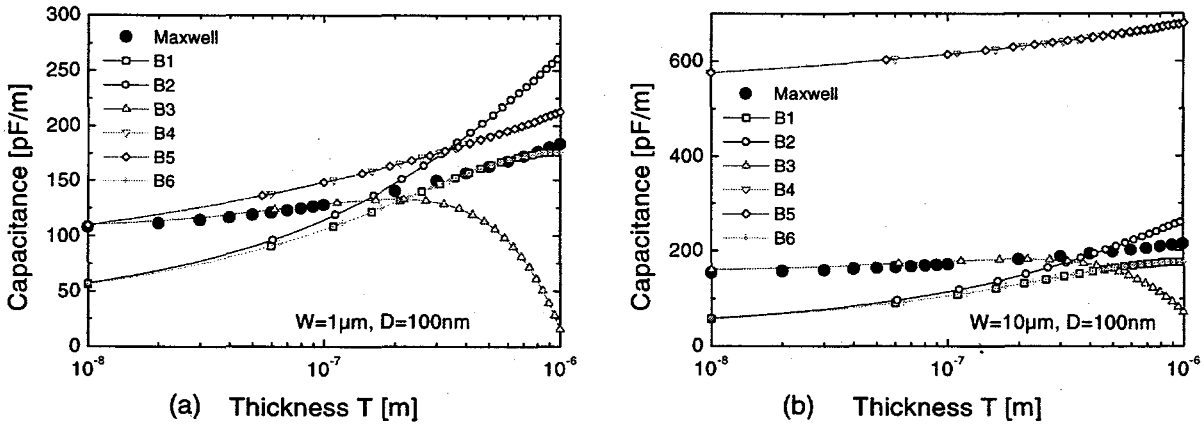


Fig. B2 - Comparison of the fringing component as results from the Eq.s B1 to B6 and from the Maxwell simulator at equal geometry: (a) W=1μm; (b) W=10μm.

From Fig. B2(a) (W=1µm) it can be observed that a good prediction can be expected only from the first (B1), the third (B3) and the sixth (B6) equation, depending in any case on the range of the thickness T. From Fig. B2(b) (W=10µm) the prediction is much worse and only the third (B3) equation seems to give a good prediction, limited to the first range of the thickness T.

An equation which can give a more accurate prediction in the range W=0.05µm÷100µm is proposed:

$$\varepsilon_{ox}\frac{W}{D}\left\{\frac{2.91}{0.83+\left(0.25\frac{D}{W}\right)^{0.25}}\left(\frac{D}{W}\right)^{0.91}+\left[\frac{3.3}{1+\left(3\frac{D}{W}\right)^{0.29}}+3\left(\frac{D}{W}\right)^{0.4}\right]\ln\left[1+\left(\frac{1}{2}\frac{D}{W}\right)^{1.05}\right]\ln\left[1+\left(\frac{T}{D}\right)^{0.72}\right]\right\} \quad \text{(B7)}$$

The accuracy of this equation in the evaluation of the fringing component can be observed from the comparisons shown in Fig. B3.

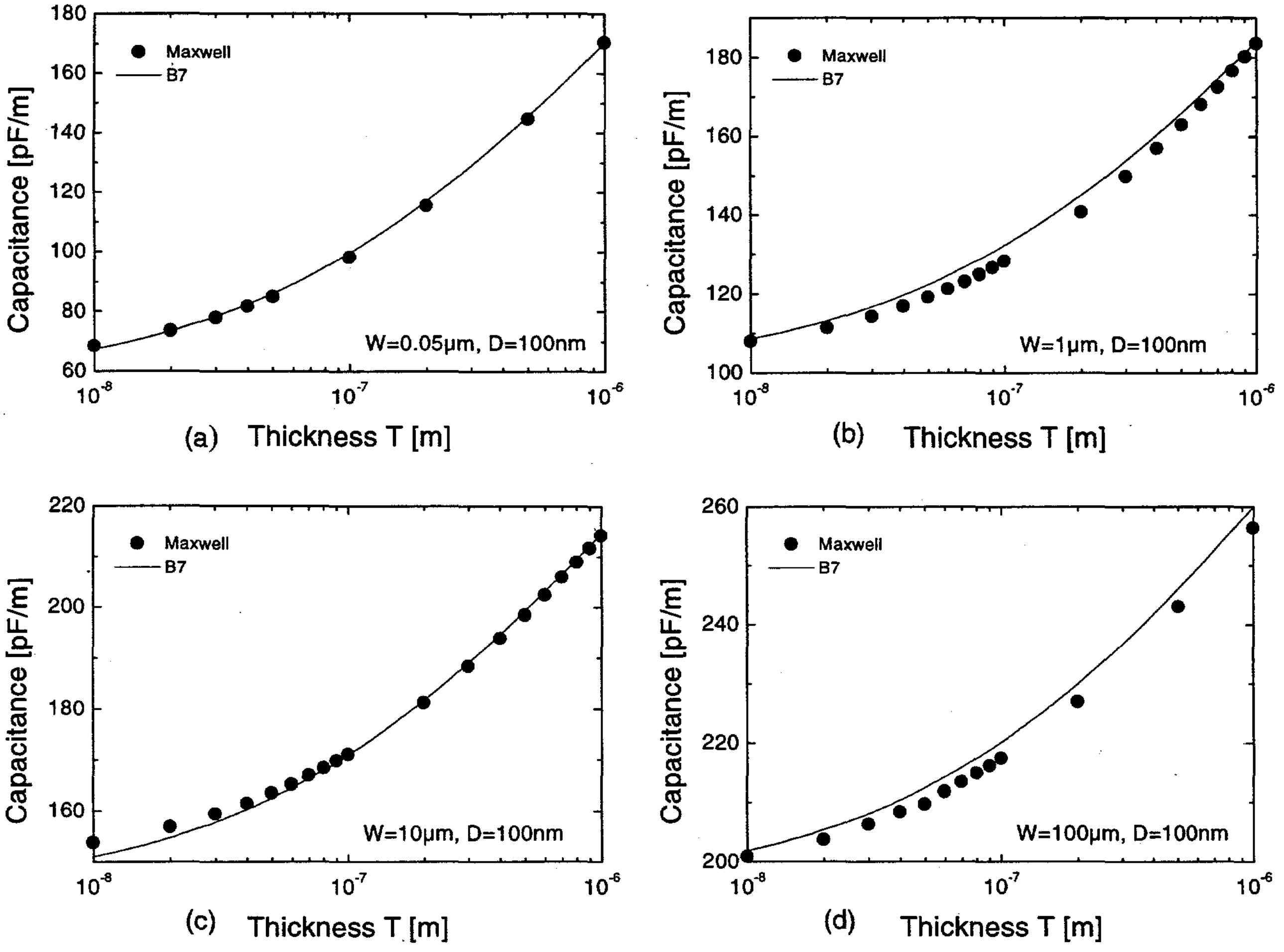


Fig. B3 - Comparison of the fringing component as results from the Eq. B7 and from the Maxwell simulator at equal geometry: (a) W=0.05µm; (b) W=1µm; (c) W=10µm; (d) W=100µm.

In our case, the *poly* thickness T and *oxide* thickness D for the linear capacitor can be determined from the General Process Specifications. Introducing these values (T=250nm, D=15nm) in the Eq.B7 and by multiplying for a factor 0.5 (fringing of a single side), the coefficient for the fringing capacitance can be determined and it depends on the "width" as shown in Fig.B4. It is worth noting that this coefficient increases as W increases, and the reason is that the fringing component includes the upper plane of the *poly*, which increases with the width W.

By considering that the typical width(s) of the linear capacitors lies between 10μm and 50μm, a value close to 40 aF/λ can be chosen.

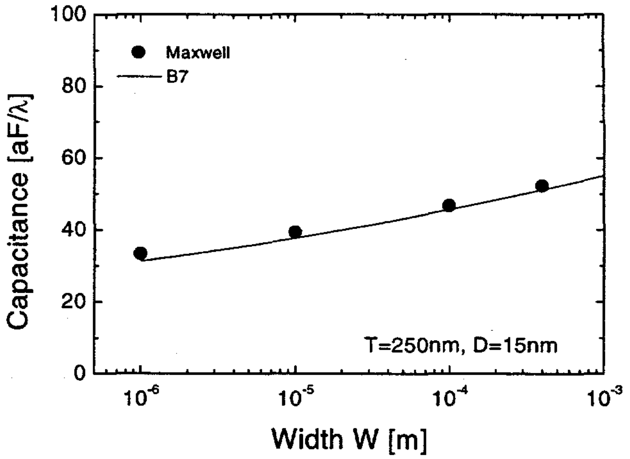


Fig. B4 - Fringing component of *poly* to *cwnsd* according to Eq.B7, expressed in aF/λ.

Appendix C - Introduction of the additional plane and layer

The following line must be introduced at the beginning of the *planes* section of the technology file:

```
substrate,ss
```

The following line must be introduced at the beginning of the *types* section of the technology file:

```
ss  ss
```

No line is added in the *styles* section of the technology file, if we don't want to see the new layer.

No line is added or modified in the *cifoutput* section of the technology file, if we don't want to see the new layer during the CIF extraction.

Some lines must be modified in the extract section:

```
planeorder well 0
planeorder implant 1
planeorder select 2
planeorder active 3
planeorder metal1 4
planeorder metal2 5
planeorder metal3 6
planeorder oxide 7
planeorder xp 8
planeorder comment 9
planeorder contact 10
planeorder via1 11
planeorder via2 12
```

becomes:

```
planeorder ss 0
planeorder well 1
planeorder implant 2
planeorder select 3
planeorder active 4
planeorder metal1 5
planeorder metal2 6
planeorder metal3 7
planeorder oxide 8
planeorder xp 9
planeorder comment 10
planeorder contact 11
planeorder via1 12
planeorder via2 13
```

Each *areacap* and *perimcap* line in the *extract* section must be replaced with a corresponding *overlap* and *sideoverlap* line as follows:

```
# REPLACED areacap (cwell)/well  8.460
 overlap (nwell,cwell)/well ss/ss  8.460

# REPLACED areacap (poly,pres,wcap,pc/a,wcap,pc/a)/active  7.920
 overlap (poly,pres,wcap,pc/a,wcap,pc/a)/active ss/ss  7.920  ~space/w

# REPLACED perimc (poly,pres,wcap,pc/a,wcap,pc/a)/active ~(poly,pres,wcap,pc/a,wcap,pc/a)/active  15
 sideoverlap (poly,pres,wcap,pc/a,wcap,pc/a)/active ~(poly,pres,wcap,pc/a,wcap,pc/a)/active ss/ss 15  ~space/w

# REPLACED areacap (m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1  2.700
 overlap (m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1 ss/ss  2.700  ~space/w,~space/a

# REPLACED perimc (m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1
   ~(m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1  9.300
sideoverlap (m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1
   ~(m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1 ss/ss  9.300
~space/w,~space/a

# REPLACED areacap (m2,m3c/m2)/metal2  1.260
 overlap (m2,m3c/m2)/metal2 ss/ss  1.260  ~space/w,~space/a

# REPLACED perimc (m2,m2c/m2,m3c/m2)/metal2 ~(m2,m2c/m2,m3c/m2)/metal2  5.100
 sideoverlap (m2,m2c/m2,m3c/m2)/metal2 ~(m2,m2c/m2,m3c/m2)/metal2 ss/ss  5.100  ~space/w,~space/a

# REPLACED areacap (m3,pad)/metal3  0.810
 overlap (m3,pad)/metal3 ss/ss  0.810  ~space/w,~space/a

# REPLACED perimc (m3,m3c/m3,pad)/metal3 ~(m3,m3c/m3,pad)/metal3  3.300
 sideoverlap (m3,m3c/m3,pad)/metal3 ~(m3,m3c/m3,pad)/metal3 ss/ss  3.300   ~space/w,~space/a
```

Finally, for the correction of the error of type 3:

```
# REPLACED sideoverlap (poly,pres,wcap,pc/a,wcap,pc/a)/active ~(poly,pres,wcap,pc/a,wcap,pc/a)/active ~space/w  15
sideoverlap (poly,pres,wcap,pc/a,wcap,pc/a)/active ~(poly,pres,wcap,pc/a,wcap,pc/a,cwnsd)/active ~space/w  15
# ADDED
sideoverlap (wcap)/active (cwnsd)/active ~space/w  40
```

Appendix D - Ext2spice Manual (by Lloyd Clonts)

NAME

ext2spice - convert from .ext format to .spice format

SYNOPSIS

ext2spice [-c cmin][-cs cdiv][-cdghmnrvBCLMNSV] extfile

DESCRIPTION

ext2spice reads a file in .ext format and creates a new file in .spice format. The output consists of a list of subcircuits, containing diodes, capacitors, resistors, and transistors. Each .ext file is transformed into a spice subcircuit.

Within an ext.defs file, the values of the different elements can be defined to automatically generate PSPICE or HSPICE.model statements in the SPICE file. In addition, an option is available to create a .cir file (if it does not exist) which calls the .spice file. Filescan be re-extracted without modifying the stimuluses which are in the .cir file. In addition, statements can be defined in the the ext.defs files which are "echoed" into the .cir file. The echo statements are very useful to predefine .include statements or setup parameters.

OPTIONS

-c cmin Do not output capacitors that are less than cmin femptoFarads. Default cmin is 1.0 FEMPTOFarads.

-cd Do not output any parastic capacitors.

-cs cdiv Scale all parasitic by value cdiv. This option is for processe with small lambda values where the tech file has its output values scaled. For these technology file modifications, area type parastics *should be scaled by a factor of the squareroot of cdiv and fringing parasitics should be scaled by a* factor of cdiv. This modification may be necessary because the magic extractor has a lower limit on parasitic value of 1 FF/lambda.

-d Generate distributed models for devices and create WELL diodes and the GLOBAL node Vsub for parasitics.

-D Generate WELL diodes and create a GLOBAL node Vsub for parasitics.

-g Create a separate .cir file to put the stimuluses, to define analysis parameters, and other statements (see echo in ext.defs).

-h Output devices in HSPICE format instead of PSPICE format.

-m Merge parallel/serial devices into an equivalent device. Applies for resistors, capacitors, and transistors.

-mm Transistors are defined with a multiplicity (M=). Averages are done on the widths, perimeters, and areas to create an equivalent device.

-mg Transistors are defined in terms of the geo= parameter. The MODEL statement for the device will define the area/perimeter of the drain and sources for the transistor.

-mn Output the NRG, NRD, and NRS spice parameter for each transistor.

-n Print all of the node names with their node numbers at the end of each subckt definition.

-C Output file in CaZM format.

-L Output transistor length as part of the MODEL name for MOS devices. For example, a p-transistor of length = 5 will be given the MODEL name CMOSP_5.

-M Print all of the node names with their node numbers into a separate .merge file.

-N Print the root node names with their node numbers into a separate .names file.

-S Output all node and device names as strings instead of NODES.

-V Make well errors global nodes and insert voltage sources to drive these wells. This feature corrects the floating well errors that occur when wells are not defined for certain devices. With this option, simulation is possible, but CIF may not correctly generate wells to prevent them from floating.

NODE CLEANUP OPERATIONS

Ext2spice checks to see if node is being used in the circuit. If the nodes is not used by device or not a part of a subcircuit terminal list, then all parasitics related to the nodes are deleted. Shorted parasitic capacitors are also deleted.

NODE NUMBERS AND NAMES

An additional feature is in the identification of node numbers. Most older SPICE programs prefer node numbers rather than node names. To aid in circuit verification and identification, ext2spice will number nodes based on a scheme similar to that used for numbering transistors. Namely, a node named 'Nx', where x is an integer, will be identified as node x. For example, a node labeled as 'N13' will be identified as node number 13. No special ending character is needed. If more than one identifier is used on a node, the lowest numbered one is used. All identifiers appear in the node listings.

For better clarity, ext2spice attempts to find the best node name to be matched with a node number for the circuit's node listing. The search criteria is the following going from lowest to highest priority:

1) Ext2spice names or names with last character as a "#".
2) Hierarchical names or names containing a "/".
3) Names in number format Nx.

To force nodes as a name instead of a number, a $NAME will cause a node to be called NAME in the circuit regardless of the –S option.

NODE MERGING

In older versions, any node name ending in the character '!' is assumed to be universal, so ext2spice electrically connected these nodes. For example, node 'Vdd!' in one cell is merged with the node 'Vdd!' in all other cells, whether or not they are physically connected. This option was removed to prevent simulation inaccuracy. Nodes should not be merged if they are not physically connected. Names ending in the characther '!' have no special interpretation in

terms of merging except for well errors (NWerror! and PWerror!) and the global substrate node. The substrate node (Vsub!) is the only true global node which should exist. The well error nodes should be corrected.

GLOBAL WELL ERRORS

The NWerror! and PWerror! signals are generated as globals to indicate that a well material is missing from the layout. With the –V option, voltage sources are added at the end of the spice file to prevent the wells from floating and to allow simulation. However, an ERROR could be present in the layout. CIF generation will add wells for transistor not going into the native substrate. However, no guarantees can be made that a well is connected to the correct bias if the well is not present for the device. The layout should be corrected by inserting a well under the devices with the errors. Ext2spice generates an error when NWerror! or PWerror! signals are found in the layout.

GENERAL DEVICE LABELLING

The user can control what identification number ext2spice gives to any particular device. The program looks to see if the gate argument is of the form "device character""name"^. If that is the case, then in the .spice file, the fet will be identified by "device character""name". For example, a transistor label'M12^' will be called M12 in the .spice file. A resistor labeld 'RC1^' will be called RC1. The same applies for bipolar transistors 'Qname^', diodes 'Dname^', and capactors 'Cname^'. A transistor whose identification number is not specified will be issued one by ext2spice. Ext2spice will create new names if conflicts occur.

TRANSISTOR LABELLING

The terminals of the MOS transistor are the source and drain. At least one terminal attribute must be 'S' or 'D' to be correctly identified. Both terminals may be labeled, but one is sufficient. In order for magic to place terminal attributes in the .ext file, the label 'S$' or 'D$' must be placed on the junction between the channel and the diffusion. When merging parallel transistors, it is sufficient to label only one of the transistors in order to specify the identification name and source-drain orientation of the merged transistor. Both Mx and S or D labels should be placed on the same transistor. S or D labels on transistors other than the one with Mx labels may be lost during the merging process.

For HSPICE users, the geo parameter can be extracted using the terminal attributes. The designations becomes 'Dx$' and 'Sx$'. If x=1,2, or 3, then geo=1 for shared drains, x=2 indicates shared sources, and x=3 or x=B means both shared drains and sources. Also, drains labelled 'DD$' indicates shared drains, sources labelled 'SS$' indicates shared sources, and labeling both the drain with 'DD$' and the source with 'SS$' will give in a geo=3 condition. For regular extraction, the geo designications are ignored.

RESISTOR DISTRIBUTED MODELLING

For improved accuracy of the resistor modelling, an option is available. A resistor becomes a subcircuit call which creates a number of individual resistive elements which equally divide the main resistor's value and include parasitics devices. The individual elements are defined using an equation define by distrib in the ext.defs file. For example, a resistor is nwell with a width of 10.0U and a length of 20.0U. The normal model is

```
R1 1 2 RNWELL_W10r0U 2.0
```

For a distribed model of 5 elements, the devices becomes

```
XR1 1 2 3 D_RNWELL_W10r0U SQ=2.0 A=200.0P P=60.0U
.
.

.SUBCKT D_RNWELL_W10r0U 1 2 3
XR1 1 4 3 E_RNWELL_W10r0U SQ='SQ/5' A='A/5' P='P/5'
XR2 4 5 3 E_RNWELL_W10r0U SQ='SQ/5' A='A/5' P='P/5'
XR3 5 6 3 E_RNWELL_W10r0U SQ='SQ/5' A='A/5' P='P/5'
XR4 6 7 3 E_RNWELL_W10r0U SQ='SQ/5' A='A/5' P='P/5'
XR5 7 2 3 E_RNWELL_W10r0U SQ='SQ/5' A='A/5' P='P/5'
.ENDS
.SUBCKT D_RNWELL_W10r0U 1 2 3
R1 1 2 RNWELL_W10r0U SCALE=equation
D1 1 3 DNWELL AREA='A/2' PJ='P/2'
D2 2 3 DNWELL AREA='A/2' PJ='P/2'
.ENDS
```

SEARCH PATHS

Ext2spice looks for three different default files like magic. The first file is ~cad/lib/ext.defs, and it is the system default file that must exist. This file defines the different devices available and setups parameters for model values. Users can add or redefine all settings and include additional information using one or both of two custom files (~/ext.defs and ./ext.defs). The local ext.defs has the highest priority in terms of value defining.

The .ext search path uses the same search procedure as magic. The ~cad/lib/magic/sys.magic, ~/.magic, and ./magic files are read in sequences and the path is determined via the path and addpath statements.

EXTRACTION DEFAULT FILE COMMANDS

The following are commands recognized in the ext.defs files.

SET

The set command (set nodename number) causes the node with the label nodename to have the node-number "number" in the spice file. For example:

```
set GND! 0
set input 7
```

would set the node with the label 'GND!' to 0, and the node with the label 'input' to 7.

DEF

The def command

```
def extname spicename spicechar flavor
```

specifies:

a extraction name (extname),
a spice model name (spicename),
a spice device designator (spicechar), and
a flavor for a fet (flavor).

For example:

```
def pfet PMOS M p
def nfet NMOS M n
```

would cause the transistors identified as 'pfet' in the .ext file to have model name 'PMOS' is the .spice file and a device definition of M"name". Likewise, 'nfet' transistors get the 'NMOS' model name. The flavor is an internal switch to indicate how a devices is to be output and what parameters are needed.

DISTRIB
The distrib command is modelling statements in the form

```
distrib extname scale deviation method parfet equation
```

specifies:

the extraction fet type (extname),
a device value (scale),
a deviation parameter (deviation),
the method of calculating a devices value (method),
the associated parasitic device (parfet), and
an equation is for distributed modelling.

The parfet and equation are optional for devices without parastics. For each type of device, the parameters are interpreted differently.

CAPACITOR
The scale becomes the number of FF/SQ.

RESISTOR
For method=0, the device's ohms/SQ is scale.
For method=1, the device's ohms/SQ is scale * exp(deviation/W) where W is the resistors width. The model name will include the width. At present, the distrib statements are used to define just capacitor and resistor model statements. The equation defines how the unit element reponds as a function its voltages, currents, or passed parameters.

PARAM
The param command (param name value) specifies defaults for generating voltages sources to fix erronous nodes in subcircuits. For example:

```
param VNWELL 5V
```

causes the a voltage source to be generated for the NWerror! node in a subcircuit.

ECHO
The echo command allows statements to be directly written to the .cir file.

```
echo statement
```

All statements should be in the form *message or SPICE will try to interpret them.

RESISTANCE CLASS COMMANDS
nrclass, prclass, plrclass, nwclass, and pwclass tell ext2spice which resistance classes (as specified by the magic technology file) correspond to layers which contain information for device. They are called re-

sistance classes because magic thinks it is extracting areas and perimeters to calculate resistance. Instead, ext2spice uses it as part of the spice transistor and capacitor description.

plrclase is for the bottom plate for the poly-poly capacitors, nrclass is nmos drain and source areas/perimeters, and prclass is pmos device. nwclass is for nwell diodes areas/perimeters and pwclass is for pwell diodes. These commands have the form:

```
nrclass k
prclass k
nwclass k
pwclass k
plrclass k
```

where k is an integer specifying the first or second or third resistance class. The defaults are prclass 2; nrclass 3;
pwclass 7; nwclass 8; plrclass 9.

SPECIAL NAMES

In cmos technology file, there are special nodes, defined in the magic technology file, which represent the bulk (substrate) or well connections for a device. PWerror!, NWerror!, and Vsub! are commonly used names. When this nodes are found, ext2spice will generate a global voltage source at the end of the .spice file to correct the problem. The best scenio is to always define the wells for transistors.

RESISTOR EXTRACTION

Because resistors are extracted like FETs, the size of the resistor's terminals affect the value extracted. The material for a resistor's terminals must be the same width as the resistor material or the FETs value will not be extracted correctly. The magic file called resistors contains the correct layout of the various types of resistors. However, a resistor's width is always printed on the following line as a comment.

WARNINGS

Since spice will be modeling the gate capacitance and the source and drain area and perimeter capacitances, it would be redundant to include them as lumped capacitances . Therefore, it is best to set the ndc, ndiff, ntrans, pdc, pdiff, and ptrans area and perimeter capacitances to 0 in the magic technology file.

SEE ALSO

magic(1) spice(1) ext(5)

AUTHOR

Core ext2spice originally done by Andy Burstein. Additions and modifications by Danny F. Newport for MOSIS npn transistors, poly and well resistors, and poly-poly capacitors extractions and other capabilities. Other modifications by Lloyd Clonts for enhancing performance, removing unused nodes, generating automatic .model statements for resistors and capacitors.

BUGS

Diffusion areas that overlap between cells might not be measured correctly.

Appendix E - Ext.defs support file for Ext2spice (by Lloyd Clonts)

```
|****************************************************************************
|
| Ext2spice defaults file ~cad/lib/ext.defs
|
|****************************************************************************


|****************************************************************************
|
| Modified 12/03/96 by Lloyd Clonts
|
|                    Groupped the different devices togethor and started to document
|     the changes.  It was out of control.
|
|****************************************************************************

|****************************************************************************
|****************************************************************************
|
| First letter of FET name must be in the following format:
|
| p = PMOS device
| n = NMOS device
| r = Resistor
| c = Capacitor
| l = inductor
|
| With this format, the ext2sim can be modified to have only one extraction
| format for all devices. The following actions will be taken for the
| given devices:
|
|                    p,n        = No changes
|                    r,l        = Short or a resistor
|                    c          = Change to a capacitor in the output
|
| LGC - 12/16/96
|
|****************************************************************************
|****************************************************************************

|****************************************************************************
|
| Class definitions
|
|****************************************************************************

                    prclass 1
                    nrclass 2
                    nwclass 3
                    pwclass 4
                    plrclass 5
```

```
|********************************************************************************
|********************************************************************************
| Base structures - Present most CMOS processes
|********************************************************************************
|********************************************************************************

                def nfet   CMOSN  M n
                def dnwell DNWELL D n
                def dndiff DNDIFF D n
                def pfet   CMOSP  M p
                def dpwell DPWELL D p
                def dpdiff DNDIFF D p

| Bipolar Types

                def qnpn NPN Q q

| Float capacitor

                def enfet  EFETN  M n
                def epfet  EFETP  M p

|********************************************************************************
| Triquent GaAs Structures
|********************************************************************************

                def ngaas M1SSac B b
                def dgaas DGAAS B b
                def egaas EGAAS B b

                def gaasdio GAASDIODE D d

                def resdiff  RDEPL_M R r
                def resdiffd RDEPL_D R r
                def resdiffe RDEPL_E R r

                def ngaasres NGAASRES R r
                def nicrom   NICROM R r
                def lgaas LGAAS L l

                distrib resdiff    10.0 0.0 0
                distrib resdiffd  625.0 0.0 0
                distrib resdiffe 1000.0 0.0 0
                distrib ngaasres  125.0 0.0 0
                distrib nicrom     50.0 0.0 0

|********************************************************************************
|********************************************************************************
| Generic Definitions - Older extraction format
|********************************************************************************
|********************************************************************************

                def ecap      CAP       C c
                def ecapp      CAPP      C c
```

```
        def pcap      PSCAP      C c p
        def pcapp     PSCAPP     C c p
        def ncap      NSCAP      C c n
        def ncapp     NSCAPP     C c n
        def hcap      HPCAP        C c
        def hcapp     HPCAPP       C c

        distrib ecap      0.475  0.0 0 ecapp
        distrib ecapp     0.088  0.0 0
        distrib pcap      0.475  0.0 0 pcapp
        distrib pcapp     0.930  0.0 0
        distrib ncap      0.475  0.0 0 ncapp
        distrib ncapp     0.930  0.0 0

        def ndres         RNDIFF R r
        def pdres         RPDIFF R r
        def polyres       RPOLY  R r
        def nwres         RNWELL R r
        def pwres         RPWELL R r

        distrib polyres     90.0  0.000 0 ecapp    SQ
        distrib ndres       25.0  0.000 0 dndiff   SQ
        distrib pdres       77.5  0.000 0 dpdiff   SQ
        distrib nwres     2392.0  1.289 1 dnwell   SQ*(1+0.02*(v(1,3)+v(2,3))+0.02*ABS(v(1,2)))
        distrib pwres     1921.0  1.347 1 dpwell   SQ*(1+0.02*(v(1,3)+v(2,3))+0.02*ABS(v(1,2)))

|*****************************************************************************
|*****************************************************************************
| 2.0 ORBIT micron definitions
|*****************************************************************************
|*****************************************************************************

        def ndres200u     RNDIFF R r
        def pdres200u     RPDIFF R r
        def polyres200u    RPOLY  R r
        def nwres200u      RNWELL R r
        def pwres200u      RPWELL R r

        distrib polyres200u    22.5  0.000 0 ecapp200u  SQ
        distrib ndres200u      25.0  0.000 0 dndiff    SQ
        distrib pdres200u      77.5  0.000 0 dpdiff    SQ
        distrib nwres200u    2392.0  1.289 1 dnwell    SQ*(1+0.02*(v(1,3)+v(2,3))+0.02*ABS(v(1,2)))
        distrib pwres200u    1921.0  1.347 1 dpwell    SQ*(1+0.02*(v(1,3)+v(2,3))+0.02*ABS(v(1,2)))

        def ecap200u   CAP2r0    C c
        def ecapp200u  CAP2r0P    C c
        def pcap200u   PSCAP2r0   C c p
        def pcapp200u  PSCAP2r0P  C c p
        def ncap200u   NSCAP2r0   C c n
        def ncapp200u  NSCAP2r0P  C c n

        def pfcap200u   PFCAP2r0   C c p
        def nfcap200u   NFCAP2r0   C c n

        distrib ecap200u   0.475  0.0 0 ecapp200u
        distrib ecapp200u  0.057  0.0 0
        distrib pcap200u   0.475  0.0 0 pcapp200u
        distrib pcapp200u  0.930  0.0 0
        distrib ncap200u   0.475  0.0 0 ncapp200u
        distrib ncapp200u  0.930  0.0 0

        distrib nfcap200u   0.475  0.0 0
        distrib pfcap200u   0.475  0.0 0

|*******************************************************************************
|*******************************************************************************
| 1.2 ORBIT micron
|
| The NWELL measurements were done by Dr. Kennedy 1994 for a ORBIT nwell process.
|
|*******************************************************************************
|*******************************************************************************

        def ecap120u   CAP1r2     C c
        def ecapp120u  CAP1r2P    C c
        def pcap120u   PSCAP1r2   C c
        def pcapp120u  PSCAP1r2P  C c
        def ncap120u   NSCAP1r2   C c
        def ncapp120u  NSCAP1r2P  C c

        distrib ecap120u   0.500  0.0 0 ecapp120u
        distrib ecapp120u  0.057  0.0 0
        distrib pcap120u   0.500  0.0 0 pcapp120u
        distrib pcapp120u  0.930  0.0 0
        distrib ncap120u   0.500  0.0 0 ncapp120u
        distrib ncapp120u  0.930  0.0 0

        def ndres120u       RNDIFF R r
        def pdres120u       RPDIFF R r
        def polyres120u     RPOLY  R r
        def nwres120u       RNWELL R r
        def pwres120u       RPWELL R r

        distrib polyres120u     22.4  0.000 0 ecapp120u  SQ
        distrib ndres120u       39.5  0.000 0 dndiff     SQ
        distrib pdres120u       77.5  0.000 0 dpdiff     SQ
        distrib nwres120u       953.0  7.160 1 dnwell    SQ*(1+0.02*(v(1,3)+v(2,3))+0.02*ABS(v(1,2)))
        distrib pwres120u       1080.6  6.968 1 dpwell    SQ*(1+0.02*(v(1,3)+v(2,3))+0.02*ABS(v(1,2)))

|*******************************************************************************
|*******************************************************************************
| 1.2 HP micron
|*******************************************************************************
|*******************************************************************************
        def ndreshp120u       RHPNDIFF120 R r
        distrib ndreshp120u   2.4  0.000 0 dndiff     SQ

        def pdreshp120u       RHPPDIFF120 R r
        distrib pdreshp120u   2.1  0.000 0 dpdiff     SQ
```

```
                    def polyreshp120u    RHPPOLY120  R r
                    distrib polyreshp120u  2.6  0.000 0 chppoly120u   SQ

                    def chppoly120u    CHPPOLY120  C c
                    distrib chppoly120u      0.57  0.000 0

                    def nwreshp120u     RHPNWELL120 R r
                    distrib nwreshp120u     821.0  0.000 0 dnwell  SQ

                    def hcap120u  HPCAP1r2  C c
                    distrib hcap120u   1.260  0.0 0 hcapp120u

# The parasitic capacitance is between the cwell and substrate.
#                   def hcapp120u  HPCAP1r2P  C c
#                   distrib hcapp120u  0.001  0.0 0

|***************************************************************************
|***************************************************************************
| Generic Definitions - New Format
|***************************************************************************
|***************************************************************************

                    def cap       CAP       C c
                    def capp       CAPP       C c
                    def cpfet      PSCAP      C c p
                    def cpfetp      PSCAPP    C c p
                    def cnfet      NSCAP      C c n
                    def cnfetp      NSCAPP    C c n
                    def cwhp       HPCAP        C c
                    def cwhpp       HPCAPP      C c

                    distrib cap       0.475  0.0 0 capp
                    distrib capp       0.088  0.0 0
                    distrib cpfet      0.475  0.0 0 cpfetp
                    distrib cpfetp     0.930  0.0 0
                    distrib cnfet      0.475  0.0 0 cnfetp
                    distrib cnfetp     0.930  0.0 0

                    def rndiff          RNDIFF R r
                    def rpdiff          RPDIFF R r
                    def rpoly           RPOLY  R r
                    def rnwell           RNWELL R r
                    def rpwell           RPWELL R r

                    distrib rpoly           90  0.000 0 capp       SQ
                    distrib rndiff        25.0  0.000 0 dndiff     SQ
                    distrib rpdiff        77.5  0.000 0 dpdiff     SQ
                    distrib rnwell      2392.0  1.289 1 dnwell     SQ*(1+0.02*(v(1,3)+v(2,3))+0.02*ABS(v(1,2)))
                    distrib rpwell      1921.0  1.347 1 dpwell     SQ*(1+0.02*(v(1,3)+v(2,3))+0.02*ABS(v(1,2)))

|***************************************************************************
|***************************************************************************
| Generic Definitions - New format
|***************************************************************************
|***************************************************************************
```

```
          def cap        CAP       C c
          def capp       CAPP      C c
          def cpfet      PSCAP     C c p
          def cpfetp     PSCAPP    C c p
          def cnfet      NSCAP     C c n
          def cnfetp     NSCAPP    C c n
          def hcap       HPCAP         C c
          def hcapp      HPCAPP        C c

          distrib cap       0.475  0.0 0 capp
          distrib capp      0.057  0.0 0
          distrib cpfet     0.475  0.0 0 cpfetp
          distrib cpfetp    0.930  0.0 0
          distrib cnfet     0.475  0.0 0 cnfetp
          distrib cnfetp    0.930  0.0 0

          def rndiff        RNDIFF R r
          def rpdiff        RPDIFF R r
          def rpoly         RPOLY  R r
          def rnwell        RNWELL R r
          def rpwell        RPWELL R r

          distrib rpoly        22.5  0.000 0 capp     SQ
          distrib rndiff       25.0  0.000 0 dndiff    SQ
          distrib rpdiff       77.5  0.000 0 dpdiff    SQ
          distrib rnwell     2392.0  1.289 1 dnwell    SQ*(1+0.02*(v(1,3)+v(2,3))+0.02*ABS(v(1,2)))
          distrib rpwell     1921.0  1.347 1 dpwell    SQ*(1+0.02*(v(1,3)+v(2,3))+0.02*ABS(v(1,2)))

|******************************************************************************
|******************************************************************************
| 2.0 ORBIT micron definitions - New Format
|******************************************************************************
|******************************************************************************

          def rndiff200u     RNDIFF R r
          def rpdiff200u     RPDIFF R r
          def rpoly200u      RPOLY  R r
          def rnwell200u     RNWELL R r
          def rpwell200u     RPWELL R r

          distrib rpoly200u      22.5  0.000 0 capp200u  SQ
          distrib rndiff200u     25.0  0.000 0 dndiff    SQ
          distrib rpdiff200u     77.5  0.000 0 dpdiff    SQ
          distrib rnwell200u   2392.0  1.289 1 dnwell    SQ*(1+0.02*(v(1,3)+v(2,3))+0.02*ABS(v(1,2)))
          distrib rpwell200u   1921.0  1.347 1 dpwell    SQ*(1+0.02*(v(1,3)+v(2,3))+0.02*ABS(v(1,2)))

          def cap200u    CAP2r0    C c
          def capp200u   CAP2r0P   C c
          def cpfet200u  PSCAP2r0  C c p
          def cpfetp200u PSCAP2r0P C c p
          def cnfet200u  NSCAP2r0  C c n
          def cnfetp200u NSCAP2r0P C c n

          def pfcap200u  PFCAP2r0  C c p
```

```
                    def nfcap200u   NFCAP2r0   C c n

                    distrib cap200u      0.475  0.0 0 capp200u
                    distrib capp200u     0.057  0.0 0
                    distrib cpfet200u    0.475  0.0 0 cpfetp200u
                    distrib cpfetp200u   0.930  0.0 0
                    distrib cnfet200u    0.475  0.0 0 cnfetp200u
                    distrib cnfetp200u   0.930  0.0 0

                    distrib nfcap200u    0.475  0.0 0
                    distrib pfcap200u    0.475  0.0 0

|*******************************************************************************
|*******************************************************************************
| 1.2 ORBIT micron - New Format
|
| The NWELL measurements were done by Dr. Kennedy 1994 for a ORBIT nwell process.
|
|*******************************************************************************
|*******************************************************************************


                    def cap120u      CAP1r2      C c
                    def capp120u     CAP1r2P     C c
                    def cpfet120u    CAP1r2   C c p
                    def cpfetp120u   CAP1r2P  C c p
                    def cnfet120u    CAP1r2   C c n
                    def cnfetp120u   CAP1r2P  C c n
#                   def cpfet120u    PSCAP1r2   C c
#                   def cpfetp120u   PSCAP1r2P  C c
#                   def cnfet120u    NSCAP1r2   C c
#                   def cnfetp120u   NSCAP1r2P  C c

                    distrib cap120u      0.500  0.0 0 capp120u
                    distrib capp120u     0.0  0.0 0
                    distrib cpfet120u    0.0  0.0 0
                    distrib cpfetp120u   0.0  0.0 0
                    distrib cnfet120u    0.0  0.0 0
                    distrib cnfetp120u   0.0  0.0 0

                    def rndiff120u       RNDIFF R r
                    def rpdiff120u       RPDIFF R r
                    def rpoly120u        RPOLY  R r
                    def rnwell120u       RNWELL R r
                    def rpwell120u       RPWELL R r

                    distrib rpoly120u        22.4  0.000 0 capp120u  SQ
                    distrib rndiff120u       39.5  0.000 0 dndiff    SQ
                    distrib rpdiff120u       77.5  0.000 0 dpdiff    SQ
                    distrib rnwell120u      953.0  7.160 1 dnwell    SQ*(1+0.02*(v(1,3)+v(2,3))+0.02*ABS(v(1,2)))
                    distrib rpwell120u     1080.6  6.968 1 dpwell    SQ*(1+0.02*(v(1,3)+v(2,3))+0.02*ABS(v(1,2)))

|*******************************************************************************
|*******************************************************************************
| 1.2 HP micron - New Format
|*******************************************************************************
```

```
|**************************************************************************************
                def cwellhp120u                 HPCAP120   C c
                def cwellhp120up        HPCAP120P  C c
                def cpolyhp120up                CHPPOLY120  C c

                def rndiffhp120u        RHPNDIFF120 R r
                def rpdiffhp120u        RHPPDIFF120 R r
                def rpolyhp120u                 RHPPOLY120  R r
                def rnwellhp120u        RHPNWELL120 R r

                distrib cwellhp120u     1.260   0.0     0
                distrib cpolyhp120up    0.57    0.0     0

                distrib rndiffhp120u    2.4     0.000   0 dndiff  SQ
                distrib rpolyhp120u     2.1     0.000   0 cpolyhp120up  SQ
                distrib rpdiffhp120u  2.6       0.000   0 dpdiff  SQ
                distrib rnwellhp120u    821.0   0.000   0 dnwell        SQ

|**************************************************************************************
|**************************************************************************************
| 0.8 HP micron - New format
|**************************************************************************************
|**************************************************************************************

                def cwellhp80u          HPCAP80   C c
                def cwellhp80up         HPCAP80P  C c
                def cpolyhp80up         CHPPOLY80  C c

                def rndiffhp80u         RHPNDIFF80 R r
                def rpdiffhp80u         RHPPDIFF80 R r
                def rpolyhp80u          RHPPOLY80  R r
                def rnwellhp80u         RHPNWELL80 R r

                distrib cwellhp80u      1.260   0.0     0
                distrib cpolyhp80up     0.57    0.0     0

                distrib rndiffhp80u     2.4     0.000   0 dndiff  SQ
                distrib rpolyhp80u      2.1     0.000   0 cpolyhp80up   SQ
                distrib rpdiffhp80u     2.6     0.000   0 dpdiff  SQ
                distrib rnwellhp80u     821.0   0.000   0 dnwell        SQ

|**************************************************************************************
|**************************************************************************************
| 0.5 HP micron - format
|**************************************************************************************
|**************************************************************************************
                def cwellhp50u          HPCAP50   C c
                def cwellhp50up         HPCAP50P  C c
                def cpolyhp50up         CHPPOLY50  C c

                def rndiffhp50u         RHPNDIFF50 R r
                def rpdiffhp50u         RHPPDIFF50 R r
                def rpolyhp50u          RHPPOLY50  R r
                def rpolysbhp50u RHPPOLYSB50  R r
                def rnwellhp50u         RHPNWELL50 R r
                distrib cwellhp50u          3.56     0.0      0
                distrib cpolyhp50up         0.088    0.0      0

                distrib rndiffhp50u         2.4      0.000    0 dndiff  SQ
                distrib rpolyhp50u          2.1      0.000    0 cpolyhp50up    SQ
                distrib rpolysbhp50u        90.0     0.000    0 cpolyhp50up    SQ
                distrib rpdiffhp50u         2.6      0.000    0 dpdiff  SQ
                distrib rnwellhp50u         821.0    0.000    0 dnwell         SQ

|***************************************************************************
|***************************************************************************
| 0.35 HP micron - format
|***************************************************************************
|***************************************************************************
                def cwellhp35u              HPCAP35    C c
                def cwellhp35up             HPCAP35P   C c
                def cpolyhp35up             CHPPOLY35  C c

                def rndiffhp35u             RHPNDIFF35 R r
                def rpdiffhp35u             RHPPDIFF35 R r
                def rpolyhp35u              RHPPOLY35  R r
                def rpolysbhp35u RHPPOLYSB35  R r
                def rnwellhp35u             RHPNWELL35 R r

                distrib cwellhp35u          3.56     0.0      0
                distrib cpolyhp35up         0.57     0.0      0

                distrib rndiffhp35u         2.4      0.000    0 dndiff  SQ
                distrib rpolyhp35u          2.1      0.000    0 cpolyhp35up    SQ
                distrib rpolysbhp35u        90.0     0.000    0 cpolyhp35up    SQ
                distrib rpdiffhp35u         2.6      0.000    0 dpdiff  SQ
                distrib rnwellhp35u         821.0    0.000    0 dnwell         SQ

|***************************************************************************
|***************************************************************************
| IMEMS - format
|***************************************************************************
|***************************************************************************
                def cbaseimems              CBASE      C c
                def cemitimems              CEMIT      C c

                def rndiffimems             RNDIFFIMEMS R r
                def rpdiffimems             RPDIFFIMEMS R r
                def rtflmimems              RTFLMIMEMS  R r
                def rnwellimems             RNWELLIMEMS R r
                def lpnp          LPNP      Q q
                def spnp          SPNP      Q q
                def vnpn          VNPN      Q q

                distrib rndiffimems         2.4 0 0
                distrib rpdiffimems         2.4 0 0
                distrib rtflmimems          2.4 0 0
                distrib rnwellimems         500 0 0
```

```
|****************************************************************************************
|****************************************************************************************
|****************************************************************************************
|****************************************************************************************
| Defaults for NWerror!, PWerror!, and CAPbody!
|****************************************************************************************
|****************************************************************************************
|****************************************************************************************

param VNWELL 5V
param NWELL DNWELL
param PWELL DPWELL
param VPWELL 0V
param VCWELL 0V

|****************************************************************************************
| Info into .cir files
|****************************************************************************************

echo ***********************************************************
echo *
echo *    Author(s):
echo *   Description:
echo *
echo ***********************************************************
echo *
echo ***********************************************************
echo ***********************************************************
echo * Include List and General setups
echo *
echo *  %.l2  = Level2
echo *  %.l3  = Level3
echo *  %.l13 = BSIM
echo *
echo ***********************************************************
echo * HP 0.5U SPICE models
echo ***********************************************************
echo *
echo *.include /usr/local/cad/lib/spice/level3/hp50_n56s.l3
echo *.include /usr/local/cad/lib/spice/level3/hp50_n61y.l3
echo *.include /usr/local/cad/lib/spice/level3/hp50_n62e.l3
echo *.include /usr/local/cad/lib/spice/level3/hp50_n64o.l3
echo *.include /usr/local/cad/lib/spice/level3/hp50_n65a.l3
echo *.include /usr/local/cad/lib/spice/level3/hp50_n6ay.l3
echo *.include /usr/local/cad/lib/spice/level3/hp50_n6bg.l3
echo *.include /usr/local/cad/lib/spice/level3/hp50_n71s.l3
echo *.include /usr/local/cad/lib/spice/level3/hp50_n72a.l3
echo *.include /usr/local/cad/lib/spice/level3/hp50_n73d.l3
echo *
echo *.include /usr/local/cad/lib/spice/level13/hp50_n56s.l13
echo *.include /usr/local/cad/lib/spice/level13/hp50_n61y.l13
echo *.include /usr/local/cad/lib/spice/level13/hp50_n62e.l13
echo *.include /usr/local/cad/lib/spice/level13/hp50_n64o.l13
echo *.include /usr/local/cad/lib/spice/level13/hp50_n65a.l13
echo *.include /usr/local/cad/lib/spice/level13/hp50_n6ay.l13
echo *.include /usr/local/cad/lib/spice/level13/hp50_n6bg.l13
echo *.include /usr/local/cad/lib/spice/level13/hp50_n71s.l13
echo *.include /usr/local/cad/lib/spice/level13/hp50_n72a.l13
echo *.include /usr/local/cad/lib/spice/level13/hp50_n73d.l13
echo *
echo **************************************************
echo * ORBIT 1.2U SPICE models
echo **************************************************
echo *
echo *.include /usr/local/cad/lib/spice/level2/n1nom1u2.l2
echo *
echo *.include /usr/local/cad/lib/spice/level3/n1fst1u2.l3
echo *.include /usr/local/cad/lib/spice/level3/n1nom1u2.l3
echo *.include /usr/local/cad/lib/spice/level3/n1slo1u2.l3
echo *
echo *.include /usr/local/cad/lib/spice/level13/oxxxx1u2.l13
echo *
echo **************************************************
echo * MOSIS/ORBIT 2.0U DIODE model parameters
echo **************************************************
echo *
echo *.include /usr/local/cad/lib/spice/level1/diocd2u0.l1
echo *
echo **************************************************
echo * ORBIT 2.0U SPICE models
echo **************************************************
echo *
echo *.include /usr/local/cad/lib/spice/level13/m96ln2u0.l13
echo *.include /usr/local/cad/lib/spice/level13/n07sn2u0.l13
echo *.include /usr/local/cad/lib/spice/level13/n13in2u0.l13
echo *.include /usr/local/cad/lib/spice/level13/n21hn2u0.l13
echo *.include /usr/local/cad/lib/spice/level13/n2cqn2u0.l13
echo *.include /usr/local/cad/lib/spice/level13/n02sn2u0.l13
echo *.include /usr/local/cad/lib/spice/level13/n09en2u0.l13
echo *.include /usr/local/cad/lib/spice/level13/n15sn2u0.l13
echo *.include /usr/local/cad/lib/spice/level13/n23qn2u0.l13
echo *.include /usr/local/cad/lib/spice/level13/n32an2u0.l13
echo *.include /usr/local/cad/lib/spice/level13/n04zn2u0.l13
echo *.include /usr/local/cad/lib/spice/level13/n0bpn2u0.l13
echo *.include /usr/local/cad/lib/spice/level13/n18mn2u0.l13
echo *.include /usr/local/cad/lib/spice/level13/n29tn2u0.l13
echo *.include /usr/local/cad/lib/spice/level13/n05in2u0.l13
echo *.include /usr/local/cad/lib/spice/level13/n11bn2u0.l13
echo *.include /usr/local/cad/lib/spice/level13/n1avn2u0.l13
echo *.include /usr/local/cad/lib/spice/level13/n2ben2u0.l13
echo *
echo **************************************************
echo * HSPICE options
echo **************************************************
echo *
echo *.options POST
echo *.options METHOD=GEAR
echo *
echo **************************************************
echo *
echo *    Used for distributed modelling or when well
echo *    diodes are generated.
echo *
echo ***************************************************
echo *
echo *Vsub Vsub 0 DC 0V  $NWell Process
echo *Vsub Vsub 0 DC 5V  $PWell Process
echo *
```

Appendix F - Ext2spice.corr code

```
# Replaces all "=" with " = " in order to get values.
sed '/=/s// = /g' $1|

# Replaces all "_L" with " " in order to get the value of NL.
sed '/_L/s// /' |

# Eliminates capacitors with value 0.0F.
sed '/0.0F/d' |

nawk ' BEGIN {LAMBDA=0.3e-6}
        {RSQ=90}
        {RSQPOLY=3.5}
        {RDW=0.0e-6}
        {RACAP=8.5e-5}
        {RPCAP=1.22e-10}
        {WMIN=1.5e-6}
        {RSQM1=0.07}
        {RSQM2=0.07}
        {RSQM3=0.05}
        {VC1=0.001}
        {VC2=0.0005}
        {TC1=0}
        {TC2=0}
        {NRSQ=2.5}
        {PRSQ=2}
        {NDC=4}
        {PDC=4}
        {PC=7}
        {CLT=5}
        {CMOSN="CMOSN"}
        {CMOSP="CMOSP"}
# Sets NL as parameter and calculates R, W, Weff, Cpar
    {if ($5=="D_RPOLY") {print $1,$2,$3,$4,$5,$6,$7$8$9,$10$11$12,("NL=0"); print
                "*",$1"="2*PC+2*RSQPOLY*2*LAMBDA/sqrt($12*1e-12/$9)+(RSQ*($9-0*0.45)/(1-
                RDW/sqrt($12*1e-12/$9))),"(W="sqrt($12*1e-12/$9)", Weff="(sqrt($12*1e-12/$9)-RDW)",
                Cpar="(RACAP*$12*1e-12+RPCAP*2*sqrt($12*1e-12*$9))")"}
     else
     if ($6=="D_RPOLY") {print $1,$3,$4,$5,$6,$7,$8$9$10,$11$12$13,("NL="$2); print
                "*",$1"="2*PC+2*RSQPOLY*2*LAMBDA/sqrt($13*1e-12/$10)+(RSQ*($10-$2*0.45)/(1-
                RDW/sqrt($13*1e-12/$10))),"(W="sqrt($13*1e-12/$10)", Weff="(sqrt($13*1e-12/$10)-RDW)",
                Cpar="(RACAP*$13*1e-12+RPCAP*2*sqrt($13*1e-12*$10))")"}
     else
     if ($2=="D_RPOLY") print $1,$2,$3,$4,$5,$6,$7$8$9,$10$11$12,("NL={NL}")
     else
     if ($2=="E_RPOLY") print $1,$2,$3,$4,$5,$6,$7$8$9,$10$11$12,("NL={NL}")
     else
     if ($5=="E_Rpoly") print $1,$2,$3,$4,$5,$6,$7$8$9,$10$11$12,("NL={NL/5}")
     else
     if ($4=="RPOLY" && $5=="{SQ}") print
                $1,$2,$3,("{2*PC+2*RSQPOLY*2*LAMBDAVSQRT(A*1pVSQ)+RSQ*(SQ-NL*0.45)V(1-
                RDWVSQRT(A*1pVSQ))}")
    else
    if ($4=="CAPP" && $5=="{A/2}") print $1,$2,$3,("{RACAP*A*1pV2+RPCAPV2*2*SQRT(A*1p*SQ)}")
    else
    if ($1==".MODEL" && $2=="RPOLY") print ".PARAM RSQ="RSQ,"RDW="RDW,"SQ=1 A=1p NL=0"
    else
    if ($1==".MODEL" && $2=="CAPP") print ".PARAM RACAP="RACAP,"RPCAP="RPCAP
    else

# Modifies NRS, NRD and NRG and adds .PARAM line for GRSQ
    if ($1~/^M/ && $6==CMOSN && $7=="W")
      {W=$9*1e-6
       L=$12*1e-6
       print $0
       getline
        {if ($1=="+" && $2=="NRG") print
                 $1,"RG",$3,(RSQPOLY*(1/$4+4*LAMBDA/L)/4+PC/2/(L/CLT/LAMBDA)),"RD",$6,(NRSQ*(
                 $7-3*LAMBDA/W)+NDC/(W/CLT/LAMBDA)),"RS",$9,(NRSQ*($10-
                 3*LAMBDA/W)+PDC/(W/CLT/LAMBDA))
         else print $0}}
    else
    if ($1~/^M/ && $6==CMOSN && $7=="M")
      {M=$9
       W=$12*1e-6
       L=$15*1e-6
       print $0
       getline
        {if ($1=="+" && $2=="NRG") print
                 $1,"RG",$3,(RSQPOLY*(1/$4/M+4*LAMBDA/L)/4+PC/2/(L/CLT/LAMBDA)),"RD",$6,(NRS
                 Q*($7*M*M-M*LAMBDA/W-
                 2*LAMBDA/W)/M+NDC*2/(W/CLT/LAMBDA)),"RS",$9,(NRSQ*($10*M*M-
                 M*LAMBDA/W-2*LAMBDA/W)/M+PDC*2/(W/CLT/LAMBDA))
         else print $0}}
    else
    if ($1~/^M/ && $6==CMOSP && $7=="W")
      {W=$9*1e-6
       L=$12*1e-6
       print $0
       getline
        {if ($1=="+" && $2=="NRG") print
                 $1,"RG",$3,(RSQPOLY*(1/$4+4*LAMBDA/L)/4+PC/2/(L/CLT/LAMBDA)),"RD",$6,(PRSQ*(
                 $7-3*LAMBDA/W)+NDC/(W/CLT/LAMBDA)),"RS",$9,(PRSQ*($10-
                 3*LAMBDA/W)+PDC/(W/CLT/LAMBDA))
         else print $0}}
    else
    if ($1~/^M/ && $6==CMOSP && $7=="M")
      {M=$9
       W=$12*1e-6
       L=$15*1e-6
       print $0
       getline
        {if ($1=="+" && $2=="NRG") print
                 $1,"RG",$3,(RSQPOLY*(1/$4/M+4*LAMBDA/L)/4+PC/2/(L/CLT/LAMBDA)),"RD",$6,(PRSQ
                 *($7*M*M-M*LAMBDA/W-
                 2*LAMBDA/W)/M+NDC*2/(W/CLT/LAMBDA)),"RS",$9,(PRSQ*($10*M*M-
                 M*LAMBDA/W-2*LAMBDA/W)/M+PDC*2/(W/CLT/LAMBDA))
       else print $0}}
    else
    if ($1=="*" && $2=="Model") print ".PARAM RSQPOLY="RSQPOLY,"PC="PC
    else

# Sets model for linear capacitors.
    if ($1==".MODEL" && $2=="LCAP") print $1,$2,"CAP (C=1
                VC1="VC1,"VC2="VC2,"TC1="TC1,"TC2="TC2,")"
    else

# Sets expression for R in the non distributed case.
    if ($5=="RPOLY") print $1,$3,$4,2*PC+2*RSQPOLY*2*LAMBDA/WMIN+(RSQ*($6-$2*0.45)/(1-
                RDW/WMIN))
    else

# Evaluates the resistance of Metal1 fets
if ($6=="RM1") {print "R"$1,$2,$4,(RSQM1*(($15+$21)/$9/$9+1))
         getline
         if ($1=="+") print "*"
         else print $0}
    else

# Evaluates the resistance of Metal2 fets
if ($6=="RM2") {print "R"$1,$2,$4,(RSQM2*(($15+$21)/$9/$9+1))
         getline
         if ($1=="+") print "*"
         else print $0}
    else

# Evaluates the resistance of Metal3 fets
if ($6=="RM3") {print "R"$1,$2,$4,(RSQM3*(($15+$21)/$9/$9+1))
         getline
         if ($1=="+") print "*"
         else print $0}

    else print $0}' |

# Separates linear capacitors LCAP
nawk ' BEGIN {K=0}
        {k=0}
        {flag=0}
    {if ($4=="LCAP")
     {K=K+1
      N1[K]=$2
      N2[K]=$3
      getline}
     else
     {while (flag<=0)
      {if (($2==N1[k] && $3==N2[k] && $1~/^C/) || ($2==N2[k] && $3==N1[k] && $1~/^C/))
      {print $1,$2,$3,"LCAP",$4
       flag=1}
       else
       if (k==K)
       {print $0
       flag=1}
```

```
        else
        {k=k+1}}}}'|

# Replaces all " = " with "=".
sed '/ = /s//=/g' |

# Delete extra * R line generated by extspice.
sed '/\* R.*$/d'
```

Appendix G - Technology File: Summary of Replacements and Additions

Planes section:

```
# ADDED
 substrate,ss
```

Types section:

```
# ADDED
 metal1  m1res,m1r
 metal2  m2res,m2r
 metal3  m3res,m3r
```

Styles section:

```
# ADDED
 m1res  20
 m1res  12
 m1res  13
 m2res  21
 m2res  12
 m2res  13
 m3res  22
 m3res  12
 m3res  13
```

Connect section:

```
# MODIFIED
 m1,m1res,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1
                      ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1
 m2,m2res,m2c/m2,m3c/m2,m3c/m2  m2c/m2,m3c/m2,m3c/m2
 m3,m3res,m3c/m3  m3c/m3
 poly,wcap,nfet,wcap,pfet,pc/a  poly,wcap,nfet,wcap,pfet,pc/a
```

Cifoutput section:

```
# MODIFIED
 layer CMF m1,m1res,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1
 labels m1,m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1
 layer CMS m2,m2res,m2c/m2,m3c/m2,m3c/m2
 labels m2,m2,m2c/m2,m3c/m2,m3c/m2
 layer CMT m3,m3res,m3c/m3
 labels m3,m3,m3c/m3
```

All Extract section:

```
extract
 style HP0.5um(hpcmos14tb)from:N81D
 cscale 1
 lambda 30
 step   100
 sidehalo 8
# MODIFIED ORDER
 planeorder ss 0
 planeorder well 1
```

```
  planeorder implant 2
  planeorder select 3
  planeorder active 4
  planeorder metal1 5
  planeorder metal2 6
  planeorder metal3 7
  planeorder oxide 8
  planeorder xp 9
  planeorder comment 10
  planeorder contact 11
  planeorder via1 12
  planeorder via2 13

 # REPLACED resist (ndiff,anres,ndc/a,nsd,nsc/a)/active 1900
  resist ((ndiff,ndc/a)/active,(pdiff,pdc/a)/active) 0
 # REPLACED resist (pdiff,apres,pdc/a,psd,psc/a)/active 2000
  resist (m1/metal1,m2/metal2,m3/metal3) 0
  resist (poly,pres,wcap,pc/a,wcap,pc/a,nfet,pfet)/active 1800
 # DELETED resist (m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1,m2c/m1)/metal1  70
 # DELETED resist (m2,m2c/m2,m3c/m2,m3c/m2)/metal2  70
 # DELETED resist (m3,m3c/m3,pad)/metal3  50

  contact ndc 4 2000
  contact pdc 4 2000
  contact pc 4 1700
  contact m2c 4 700
  contact m3c 4 360


 #wellcap
  overlap wcap capwell   207.900

 #nwell,cwell,pwell
 # REPLACED areacap (nwell,cwell)/well   8.460
 # REPLACED areacap (cwell)/well  8.460
  overlap (nwell,cwell)/well ss/ss  8.460


 #ndiff
  areacap (anres)/active  48.330
  overlap (anres)/active ~space/w  48.330
  perimc (anres)/active ~(anres)/active  67.800
  sideoverlap (anres)/active ~(anres)/active ~space/w  67.800
 # MODEL HANDLES THIS: areacap (ndiff,anres,ndc/a)/active  48.330
 # MODEL HANDLES THIS: overlap (ndiff,anres,ndc/a)/active ~space/w  48.330
 # MODEL HANDLES THIS: perimc (ndiff,anres,ndc/a)/active ~(ndiff,anres,ndc/a,nfet,pfet)/active  67.800
 # MODEL HANDLES THIS: sideoverlap (ndiff,anres,ndc/a)/active ~(ndiff,anres,ndc/a,nfet,pfet)/active ~space/w
                     67.800

 #pdiff
  areacap (apres)/active  84.600
  overlap (apres)/active ~space/w  84.600
  perimc (apres)/active ~(apres)/active  67.500
  sideoverlap (apres)/active ~(apres)/active ~space/w  67.500
 # MODEL HANDLES THIS: areacap (pdiff,apres,pdc/a)/active  84.600
```

```
# MODEL HANDLES THIS: overlap (pdiff,apres,pdc/a)/active ~space/w  84.600
# MODEL HANDLES THIS: perimc (pdiff,apres,pdc/a)/active ~(pdiff,apres,pdc/a,nfet,pfet)/active  67.500
# MODEL HANDLES THIS: sideoverlap (pdiff,apres,pdc/a)/active ~(pdiff,apres,pdc/a,nfet,pfet)/active ~space/w
                    67.500

#polycap

#poly
 sidewall (poly,pres,wcap,pc/a,wcap,pc/a)/active ~(poly,pres,wcap,pc/a,wcap,pc/a)/active
                    ~(poly,pres,wcap,pc/a,wcap,pc/a)/active (poly,pres,wcap,pc/a,wcap,pc/a)/active  2.136
# REPLACED areacap (poly,pres,wcap,pc/a,wcap,pc/a)/active  7.920
 overlap (poly,pres,wcap,pc/a,wcap,pc/a)/active ss/ss  7.920  ~space/w
 overlap (poly,pres,pc/a,pc/a)/active ~space/w  7.920
# REPLACED perimc (poly,pres,wcap,pc/a,wcap,pc/a)/active ~(poly,pres,wcap,pc/a,wcap,pc/a)/active  15
 sideoverlap (poly,pres,wcap,pc/a,wcap,pc/a)/active ~(poly,pres,wcap,pc/a,wcap,pc/a)/active ss/ss 15  ~space/w
# REPLACED sideoverlap (poly,pres,wcap,pc/a,wcap,pc/a)/active ~(poly,pres,wcap,pc/a,wcap,pc/a)/active
                    ~space/w  15
 sideoverlap (poly,pres,wcap,pc/a,wcap,pc/a)/active ~(poly,pres,wcap,pc/a,wcap,pc/a,cwnsd)/active ~space/w  15
# ADDED
 sideoverlap (wcap)/active (cwnsd)/active ~space/w  40
# MODEL HANDLES THIS: overlap (nfet)/active (ndiff,anres,ndc/a)/active 324.360
# MODEL HANDLES THIS: sideoverlap (nfet)/active ~(nfet)/active (ndiff,anres,ndc/a)/active  15
# MODEL HANDLES THIS: overlap (pfet)/active (pdiff,apres,pdc/a)/active 308.160
# MODEL HANDLES THIS: sideoverlap (pfet)/active ~(pfet)/active (pdiff,apres,pdc/a)/active  15

#poly2

#metal1
 sidewall (m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1
                    ~(m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1
                    ~(m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1
                    (m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1  6.690
# REPLACED areacap (m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1  2.700
 overlap (m1,nsc/m1,cwc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1 ss/ss  2.700  ~space/w,~space/a

#metal-sub blocked by ~space/a,~space/c
# REPLACED overlap (m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1 ~space/w
                    2.700 ~space/a
 overlap (m1,nsc/m1,cwc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1 ~space/w  2.700 ~space/a
# REPLACED perimc (m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1
                    ~(m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1  9.300
 sideoverlap (m1,nsc/m1,cwc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1
                    ~(m1,nsc/m1,cwc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1 ss/ss  9.300  ~space/w,~space/a
# REPLACED sideoverlap (m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1
                    ~(m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1 ~space/w  9.300
                    ~space/a
 sideoverlap (m1,nsc/m1,cwc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1
                    ~(m1,nsc/m1,cwc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1 ~space/w  9.300  ~space/a

#metal-diff blocked by ~space/c
# REPLACED overlap (m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1
                    (ndiff,anres,ndc/a)/active  2.790
 overlap (m1,nsc/m1,cwc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1 (ndiff,anres,ndc/a)/active  2.790
```

```
# REPLACED sideoverlap (m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1
                    ~(m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1
                    (ndiff,anres,ndc/a)/active  9.300
 sideoverlap (m1,nsc/m1,cwc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1
                    ~(m1,nsc/m1,cwc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1 (ndiff,anres,ndc/a)/active  9.300
# REPLACED overlap (m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1
                    (pdiff,apres,pdc/a)/active  2.790
 overlap (m1,nsc/m1,cwc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1 (pdiff,apres,pdc/a)/active  2.790
# REPLACED sideoverlap (m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1
                    ~(m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1
                    (pdiff,apres,pdc/a)/active  9.300
 sideoverlap (m1,nsc/m1,cwc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1
                    ~(m1,nsc/m1,cwc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1 (pdiff,apres,pdc/a)/active  9.300

#metal-polycap blocked by poly

#metal-poly blocked by poly2
# REPLACED overlap (m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1
                    (poly,pres,wcap,pc/a,wcap,pc/a,nfet,pfet)/active  5.400
 overlap (m1,nsc/m1,cwc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1
                    (poly,pres,wcap,pc/a,wcap,pc/a,nfet,pfet)/active  5.400  sideoverlap
                    (m1,nsc/m1,cwc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1
                    ~(m1,nsc/m1,cwc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1
                    (poly,pres,wcap,pc/a,wcap,pc/a,nfet,pfet)/active  15
 sideoverlap (poly,pres,wcap,pc/a,wcap,pc/a,nfet,pfet)/active ~(poly,pres,wcap,pc/a,wcap,pc/a,nfet,pfet)/active
                    (m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1  15

#metal-poly2 not blocked

#metal2
 sidewall (m2,m2c/m2,m3c/m2)/metal2 ~(m2,m2c/m2,m3c/m2)/metal2 ~(m2,m2c/m2,m3c/m2)/metal2
                    (m2,m2c/m2,m3c/m2)/metal2  9.090
# REPLACED areacap (m2,m3c/m2)/metal2  1.260
 overlap (m2,m3c/m2)/metal2 ss/ss  1.260  ~space/w,~space/a

#metal2-sub blocked by ~space/a,~space/m1,poly2/polycap
 overlap (m2,m3c/m2)/metal2 ~space/w  1.260 ~space/a,~space/m1
# REPLACED perimc (m2,m2c/m2,m3c/m2)/metal2 ~(m2,m2c/m2,m3c/m2)/metal2  5.100
 sideoverlap (m2,m2c/m2,m3c/m2)/metal2 ~(m2,m2c/m2,m3c/m2)/metal2 ss/ss  5.100  ~space/w,~space/a
 sideoverlap (m2,m2c/m2,m3c/m2)/metal2 ~(m2,m2c/m2,m3c/m2)/metal2 ~space/w  5.100 ~space/a,~space/m1
 overlap (m2,m3c/m2)/metal2 (ndiff,anres,ndc/a)/active  1.260 ~space/m1
 sideoverlap (m2,m2c/m2,m3c/m2)/metal2 ~(m2,m2c/m2,m3c/m2)/metal2 (ndiff,anres,ndc/a)/active  5.100
                    ~space/m1
 overlap (m2,m2c/m2,m3c/m2)/metal2 (pdiff,apres,pdc/a)/active  1.260 ~space/m1
 sideoverlap (m2,m2c/m2,m3c/m2)/metal2 ~(m2,m2c/m2,m3c/m2)/metal2 (pdiff,apres,pdc/a)/active  5.100
                    ~space/m1

#metal2-polycap by ~space/m1,poly

#metal2-poly blocked by ~space/m1,poly2
 overlap (m2,m3c/m2)/metal2 (poly,pres,wcap,pc/a,wcap,pc/a,nfet,pfet)/active  1.620 ~space/m1
 sideoverlap (m2,m2c/m2,m3c/m2)/metal2 ~(m2,m2c/m2,m3c/m2)/metal2
                    (poly,pres,wcap,pc/a,wcap,pc/a,nfet,pfet)/active  6.300 ~space/m1
 sideoverlap (poly,pres,wcap,pc/a,wcap,pc/a,nfet,pfet)/active ~(poly,pres,wcap,pc/a,wcap,pc/a,nfet,pfet)/active
                    (m2,m2c/m2,m3c/m2)/metal2  6.300 ~space/m1
```

```
#metal2-poly2 blocked by ~space/m1

#M2->M1
 overlap (m2,m3c/m2)/metal2 (m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1  4.050
 sideoverlap (m2,m2c/m2,m3c/m2)/metal2 ~(m2,m2c/m2,m3c/m2)/metal2
                 (m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1 11.700
 sideoverlap (m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1
                 ~(m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1
                 (m2,m2c/m2,m3c/m2)/metal2 11.700

#metal3
 sidewall (m3,m3c/m3,pad)/metal3 ~(m3,m3c/m3,pad)/metal3 ~(m3,m3c/m3,pad)/metal3 (m3,m3c/m3,pad)/metal3
                 12.900
# REPLACED areacap (m3,pad)/metal3  0.810
 overlap (m3,pad)/metal3 ss/ss 0.810  ~space/w,~space/a


#metal3-sub blocked by ~space/a,~space/m1,~space/m2,poly2/polycap
 overlap (m3,pad)/metal3 ~space/w  0.810 ~space/a,~space/m1,~space/m2
# REPLACED perimc (m3,m3c/m3,pad)/metal3 ~(m3,m3c/m3,pad)/metal3  3.300
 sideoverlap (m3,m3c/m3,pad)/metal3 ~(m3,m3c/m3,pad)/metal3 ss/ss 3.300  ~space/w,~space/a
 sideoverlap (m3,m3c/m3,pad)/metal3 ~(m3,m3c/m3,pad)/metal3 ~space/w  3.300 ~space/a,~space/m1,~space/m2

#metal3-*diff blocked by ~space/m1,~space/m2,poly2/polycap
 overlap (m3,pad)/metal3 (ndiff,anres,ndc/a)/active  0.810 ~space/m1,~space/m2
 sideoverlap (m3,m3c/m3,pad)/metal3 ~(m3,m3c/m3,pad)/metal3 (ndiff,anres,ndc/a)/active  3.300
                 ~space/m1,~space/m2
 overlap (m3,pad)/metal3 (pdiff,apres,pdc/a)/active  0.810 ~space/m1,~space/m2
 sideoverlap (m3,m3c/m3,pad)/metal3 ~(m3,m3c/m3,pad)/metal3 (pdiff,apres,pdc/a)/active  3.300
                 ~space/m1,~space/m2

#metal3-polycap by ~space/m1,~space/m2,poly

#metal3-poly blocked by ~space/m1,~space/m2,poly2
 overlap (m3,pad)/metal3 (poly,pres,wcap,pc/a,wcap,pc/a,nfet,pfet)/active  0.990 ~space/m1,~space/m2
 sideoverlap (m3,m3c/m3,pad)/metal3 ~(m3,m3c/m3,pad)/metal3 (poly,pres,wcap,pc/a,wcap,pc/a,nfet,pfet)/active
                 3.600 ~space/m1,~space/m2
 sideoverlap (poly,pres,wcap,pc/a,wcap,pc/a,nfet,pfet)/active ~(poly,pres,wcap,pc/a,wcap,pc/a,nfet,pfet)/active
                 (m3,m3c/m3,pad)/metal3  3.600 ~space/m1,~space/m2

#metal3-poly2 blocked by ~space/m1

#M3->M1

#metal3-metal1 blocked by ~space/m2
 overlap (m3,pad)/metal3 (m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1  1.440
                 ~space/m2
 sideoverlap (m3,m3c/m3,pad)/metal3 ~(m3,m3c/m3,pad)/metal3
                 (m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1  5.400 ~space/m2
 sideoverlap (m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1
                 ~(m1,ndc/m1,nsc/m1,cwc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1)/metal1
                 (m3,m3c/m3,pad)/metal3  5.400 ~space/m2

#M3->M2
```

```
 overlap (m3,pad)/metal3 (m2,m2c/m2,m3c/m2)/metal2  4.230
 sideoverlap (m3,m3c/m3,pad)/metal3 ~(m3,m3c/m3,pad)/metal3 (m2,m2c/m2,m3c/m2)/metal2  10.800
 sideoverlap (m2,m2c/m2,m3c/m2)/metal2 ~(m2,m2c/m2,m3c/m2)/metal2 (m3,m3c/m3,pad)/metal3  10.800

#metal4

#fets

 fet  pfet pdiff,pdc 2 pfet Vdd! nwell 15 308
 fet  pfet pdiff,pdc 1 pfet Vdd! nwell 15 308

# REPLACED fet  nfet ndiff,ndc 2 nfet Gnd! pwell 15 324
 fet  nfet ndiff,ndc 2 nfet $$G_Vsub! pwell,ss 15 324
# REPLACED fet  nfet ndiff,ndc 1 nfet Gnd! pwell 15 324
 fet  nfet ndiff,ndc 1 nfet $$G_Vsub! pwell,ss 15 324

 fetresis  pfet linear        15658
 fetresis  pfet saturation     15658
 fetresis  nfet linear        4267
 fetresis  nfet saturation     4267

# ADDED
 fet  pres poly 2 polyres $$G_Vsub! ss,nwell,pwell 0 0

# ADDED
 fet  wcap cwnsd 1 wcap $$G_Vsub! 0 0

# ADDED
 fet  m1res m1,ndc/m1,nsc/m1,pdc/m1,psc/m1,cwc/m1,pc/m1,m2c/m1 2 m1res $$G_Vsub! 0 0
# ADDED
 fet  m2res m2,m2c/m2,m3c/m2,m3c/m2 2 m2res $$G_Vsub! 0 0
# ADDED
 fet  m3res m3,m3c/m3 2 m3res $$G_Vsub! 0 0end

end
```