

## Table of Contents

Revision History 2

Design Rule Specifications 3

Nburied rules 4

Nwell rules 5

Pwell rules 6

Oxide Rules 7

Thick Oxide Rules 8

N+ Implant rules 9

P+ Implant rules 10

POLY Rules 11

Contact rules 12

Metal rules 13

VIA rules 14

Capacitor Metal 15

Passivation rules 16

Latch-up rules 17

Antenna Rules 18

Density rules 22

Connectivity Definition 23

Component LVS Definitions 24

Resistors 25

Capacitors 29

Inductor 31

Bipolars 32

Nmos 34

HV Nmos 35

RF Nmos 36

Pmos and HV Pmos 37

RF Pmos 38

## ...contents...

<u>Diodes</u>	<u>39</u>
<u>Junction Varactors</u>	<u>40</u>
<u>LVS Diva Compare rules</u>	<u>41</u>
<u>LVS Assura Compare rules</u>	<u>42</u>
<u>Library CDF Definition</u>	<u>43</u>
<u>Library MOS Definitions</u>	<u>44</u>
<u>Global MOS Parameters</u>	<u>45</u>
<u>Callbacks</u>	<u>48</u>
<u>MOS Callback Procedures</u>	<u>48</u>
<u>CDF Macro Definitions</u>	<u>49</u>
<u>mosParamsA - MOS Parameters (Part A)</u>	<u>49</u>
<u>mosParamsTap - MOS Tap Parameters</u>	<u>54</u>
<u>mosParamsB - MOS Parameters (Part B)</u>	<u>55</u>
<u>mosSimInfo - MOS Simulation Information (common)</u>	<u>61</u>
<u>mosSimInfoRF</u>	<u>63</u>
<u>mos3n - 3 terminal NMOS Transistor CDF</u>	<u>65</u>
<u>mos3p - 3 terminal PMOS Transistor CDF</u>	<u>66</u>
<u>mos4 - 4 terminal MOS Transistor CDF</u>	<u>67</u>
<u>mos4RF - 4 terminal RF MOS Transistor CDF</u>	<u>68</u>
<u>CDF Definitions</u>	<u>69</u>
<u>MOS Transistor</u>	<u>69</u>
<u>nmos3</u>	<u>69</u>
<u>nmos</u>	<u>71</u>
<u>nmos3hv</u>	<u>73</u>
<u>nmoshv</u>	<u>75</u>
<u>nmosrf</u>	<u>77</u>
<u>pmos3</u>	<u>79</u>
<u>pmos</u>	<u>81</u>
<u>pmos3hv</u>	<u>83</u>

**...contents...**

<u>pmoshv</u>	<u>85</u>
<u>pmosrf</u>	<u>87</u>
<u>PCells</u>	<u>89</u>
<u>PCell SKILL Procedures</u>	<u>89</u>
<u>MOS Abutment</u>	<u>89</u>
<u>Metal Coverage</u>	<u>90</u>
<u>PCell Macros</u>	<u>91</u>
<u>mos (macro)</u>	<u>91</u>
<u>mos_nplus (macro)</u>	<u>105</u>
<u>mos_pplus (macro)</u>	<u>105</u>
<u>mos_rf (macro)</u>	<u>108</u>
<u>mos_volt (macro)</u>	<u>109</u>
<u>mos_volt_tap (macro)</u>	<u>110</u>
<u>mos_nw (macro)</u>	<u>110</u>
<u>mos_pw (macro)</u>	<u>110</u>
<u>mos_nw_tap (macro)</u>	<u>111</u>
<u>mos_nbl_tap (macro)</u>	<u>111</u>
<u>mos_pw_tap (macro)</u>	<u>111</u>
<u>mos_cap_tap (macro)</u>	<u>111</u>
<u>mosltap (macro)</u>	<u>112</u>
<u>mosTap (macro)</u>	<u>115</u>
<u>mosTap_pplus (macro)</u>	<u>122</u>
<u>mosTap_nplus (macro)</u>	<u>123</u>
<u>PCell Devices</u>	<u>124</u>
<u>nmos3</u>	<u>124</u>
<u>pmos3</u>	<u>124</u>
<u>nmos3hv</u>	<u>124</u>
<u>pmos3hv</u>	<u>124</u>
<u>nmos</u>	<u>125</u>

## ...contents...

<u>pmos</u>	<u>125</u>
<u>nmoshv</u>	<u>126</u>
<u>pmoshv</u>	<u>126</u>
<u>nmosrf</u>	<u>127</u>
<u>pmosrf</u>	<u>127</u>
<u>Library RESISTOR Definitions</u>	<u>128</u>
<u>Global Resistor Parameters</u>	<u>129</u>
<u>Poly Resistor</u>	<u>130</u>
<u>Diffusion Resistor</u>	<u>131</u>
<u>NWell Resistor</u>	<u>132</u>
<u>Callbacks</u>	<u>133</u>
<u>Resistor Callback Procedures</u>	<u>133</u>
<u>Resistor Spectre netlist procedure</u>	<u>134</u>
<u>CDF Macro Definitions</u>	<u>135</u>
<u>resParamsA - Resistor CDF Parameters (Part A)</u>	<u>135</u>
<u>resParamsB_uniSeg - Resistor CDF Parameters (Part B)</u>	<u>137</u>
<u>resParamsB_multiSeg - Resistor CDF Parameters (Part B)</u>	<u>138</u>
<u>resParamsC - Resistor CDF Parameters (Part C)</u>	<u>140</u>
<u>resParamsTap - Resistor CDF Parameters (Tap)</u>	<u>141</u>
<u>resParamsD - Resistor CDF Parameters (Part D)</u>	<u>142</u>
<u>resParamsE - Resistor CDF Parameters (Part E)</u>	<u>146</u>
<u>resParamsF - Resistor CDF Parameters (Part F)</u>	<u>148</u>
<u>resSimInfo2 - 2 terminal Resistor Simulation Information</u>	<u>149</u>
<u>resSimInfo3 - 3 terminal Resistor Simulation Information</u>	<u>150</u>
<u>res2_uniSeg - 2 terminal Resistor CDF</u>	<u>151</u>
<u>res3_uniSeg - 3 terminal Resistor CDF</u>	<u>151</u>
<u>res2_multiSeg - 2 terminal Resistor CDF</u>	<u>151</u>
<u>res3_multiSeg - 3 terminal Resistor CDF</u>	<u>151</u>
<u>Resistors Schematic Components</u>	<u>152</u>

## ...contents...

<u>resPrim</u>	<u>152</u>
<u>resPrim2</u>	<u>153</u>
<u>resPrim3</u>	<u>154</u>
<u>Resistors</u>	<u>155</u>
<u>polyres</u>	<u>155</u>
<u>polyhres</u>	<u>157</u>
<u>nplusres</u>	<u>159</u>
<u>pplusres</u>	<u>161</u>
<u>nwellres</u>	<u>163</u>
<u>mxres</u>	<u>165</u>
<u>PCell Macros</u>	<u>167</u>
<u>res_uniSeg (macro)</u>	<u>167</u>
<u>res_multiSeg (macro)</u>	<u>167</u>
<u>mres_multiSeg (macro)</u>	<u>167</u>
<u>res_poly (macro)</u>	<u>168</u>
<u>res_poly_nplus (macro)</u>	<u>171</u>
<u>res_poly_pplus (macro)</u>	<u>171</u>
<u>res_poly_nw (macro)</u>	<u>172</u>
<u>res_poly_id (macro)</u>	<u>172</u>
<u>res_poly_sb (macro)</u>	<u>173</u>
<u>res_diff (macro)</u>	<u>174</u>
<u>res_diff_nplus (macro)</u>	<u>177</u>
<u>res_diff_pplus (macro)</u>	<u>177</u>
<u>res_diff_nw (macro)</u>	<u>178</u>
<u>res_diff_id (macro)</u>	<u>178</u>
<u>res_nwell (macro)</u>	<u>179</u>
<u>res_nwell_id (macro)</u>	<u>184</u>
<u>res_connect (macro)</u>	<u>185</u>
<u>res_m (macro)</u>	<u>186</u>

## ...contents...

<u>res_m1 (macro)</u>	<u>187</u>
<u>res_m2 (macro)</u>	<u>188</u>
<u>res_m3 (macro)</u>	<u>189</u>
<u>res_m4 (macro)</u>	<u>190</u>
<u>res_m5 (macro)</u>	<u>191</u>
<u>res_m6 (macro)</u>	<u>192</u>
<u>res_m1_id (macro)</u>	<u>193</u>
<u>res_m2_id (macro)</u>	<u>193</u>
<u>res_m3_id (macro)</u>	<u>194</u>
<u>res_m4_id (macro)</u>	<u>194</u>
<u>res_m5_id (macro)</u>	<u>195</u>
<u>res_m6_id (macro)</u>	<u>195</u>
<u>m1res_connect (macro)</u>	<u>196</u>
<u>m2res_connect (macro)</u>	<u>197</u>
<u>m3res_connect (macro)</u>	<u>198</u>
<u>m4res_connect (macro)</u>	<u>199</u>
<u>m5res_connect (macro)</u>	<u>200</u>
<u>m6res_connect (macro)</u>	<u>201</u>
<u>PCell Devices</u>	<u>202</u>
<u>Poly Resistor</u>	<u>202</u>
<u>High Poly Resistor</u>	<u>202</u>
<u>N+ Diff Resistor</u>	<u>202</u>
<u>P+ Diff Resistor</u>	<u>202</u>
<u>NWell Resistor</u>	<u>202</u>
<u>Metal Resistor</u>	<u>202</u>
<u>Library CAP Definitions</u>	<u>203</u>
<u>CDF Definitions</u>	<u>204</u>
<u>mimcap</u>	<u>204</u>
<u>CDF Macro Definitions</u>	<u>205</u>

## ...contents...

<u>Capacitor</u>	205
<u>Callbacks</u>	218
<u>mimcap</u>	218
<u>Pcell</u>	219
<u>mimcap</u>	219
<u>OLD CDF/PCELL/CALLBACK Definitions</u>	220
<u>Junction Varactors</u>	221
<u>Junction Varactor Pcells</u>	222
<u>xjvar_w40 pcell</u>	222
<u>xjvar_nf36 pcell</u>	228
<u>Junction Varactor CDF</u>	234
<u>xjvar_w40 cdf</u>	234
<u>xjvar_nf36 cdf</u>	236
<u>Junction Varactor Callback</u>	238
<u>Diode Macro</u>	239
<u>ndio</u>	242
<u>pdio</u>	243
<u>inductor</u>	244
<u>vpnp</u>	247
<u>npn</u>	250
<u>pnp</u>	253
<u>moscap macro</u>	256
<u>nmoscap</u>	262
<u>nmoscap3</u>	263
<u>pmoscap</u>	264
<u>pmoscap3</u>	265
<u>Component Callback Definitions</u>	266
<u>Common Callback Procedures</u>	266
<u>diode</u>	267

## ...contents

<u>inductor</u>	<u>268</u>
<u>bipolar</u>	<u>269</u>
<u>nmoscap / pmoscap</u>	<u>270</u>
<u>Component PCell Code</u>	<u>271</u>
<u>Common Pcell Procedures</u>	<u>271</u>
<u>ndio</u>	<u>272</u>
<u>pdio</u>	<u>273</u>
<u>inductor</u>	<u>274</u>
<u>npn</u>	<u>277</u>
<u>pnv</u>	<u>280</u>
<u>vpnp</u>	<u>283</u>
<u>moscap (macro)</u>	<u>287</u>
<u>nmoscap (uses nmos pcell)</u>	<u>288</u>
<u>nmoscap3 (uses nmos pcell)</u>	<u>288</u>
<u>pmoscap (uses pmos pcell)</u>	<u>289</u>
<u>pmoscap3 (uses pmos pcell)</u>	<u>289</u>
<u>Basic Test Structures</u>	<u>290</u>
<u>Taps</u>	<u>291</u>
<u>Create Layouts</u>	<u>292</u>
<u>Appendix A</u>	<u>A1</u>
<u>Appendix B</u>	<u>B1</u>



## Cadence Design Systems

### GPDK 180 nm Mixed Signal Process Spec

### REV 3.3

#### DISCLAIMER

The information contained herein is provided by Cadence on an "AS IS" basis without any warranty, and Cadence has no obligation to support or otherwise maintain the information. Cadence disclaims any representation that the information does not infringe any intellectual property rights or proprietary rights of any third parties. There are no other warranties given by Cadence, whether express, implied or statutory, including, without limitation, implied warranties of merchantability and fitness for a particular purpose.

#### STATEMENT OF USE

This information contains confidential and proprietary information of Cadence. No part of this information may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any human or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise, without the prior written permission of Cadence. This information was prepared for informational purpose and is for use by Cadence customers only. Cadence reserves the right to make changes in the information at any time and without notice.

#### Generator Information

Sample runset for 180 nm  
technology

Default Grid: 0.005  
Valid Angle: 45  
Flag Acute: true  
Flag Self-intersecting: true

## Revision History

### DRC Revision History

---

#### VERSION v3.3

---

##### Changes:

1. gpd180 OA22 library built natively with IC6.1.3.1 release code
2. gpd090 CDB library built natively with IC5.10.41\_USR5.90.69 release code
3. Added missing 64bit context files (CCR 568109)
4. Renamed LEFDefaultRouteSpec to LEFDefaultRouteSpec\_gpd180 (CCR 594263)
5. Techfile updates made in preparation for IC6.1.3 release
6. Removed CDF extraneous simulation MOS parameters (CCR 595042)
7. Created new QRC database with 3d field solver information
8. Updated stream maps to include fill layer purpose
9. Updated DRC and Extract decks to merge drawing and fill layer purposes

---

#### VERSION v3.2

---

##### Changes:

1. gpd180 OA22 library built natively with IC6.1.1 preFCS code
2. Added parasitic AD/AS/PD/PS calculations to the Assura extract rules
3. Added ignore of "simM" to the CPH lam file to solve back annotation problem
4. Updated Circuit prospector libinit settings to match new format
5. Corrected names in the resistor model file for ADS
6. Modified libinit to prevent overwrite of simulation model setup information
7. Modified Assura RF LVS files for VPCM inductor extraction
8. Modified Assura RF LVS files to save additional recognition layers
9. Modified Assura RF LVS files to add dummy device filtering
10. Modified Assura RF LVS files to avoid double counting of contact resistances
11. Added switch to Assura RF LVS files control amount of saved extracted layers
12. Modified SCparameters.cds file for RF kit
13. Updates made to the display.drf file for layout productivity
14. Updates made to the techfile layer display sections for layout productivity

---

#### VERSION v3.1

---

##### Changes:

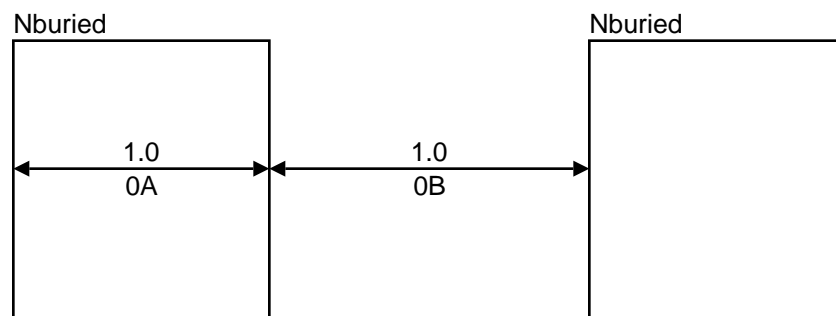
1. Gpd180 OA22 library built natively with IC6.1.0 FCS code
2. Removed DBU settings in the library cdsenv file
3. Updated the CPH Lam file with VXL ignore check parameters
4. Added a PDK version print statement to the library libinit file
5. Added Circuit Prospector default settings in PDK init files

...

## **Design Rule Specifications**

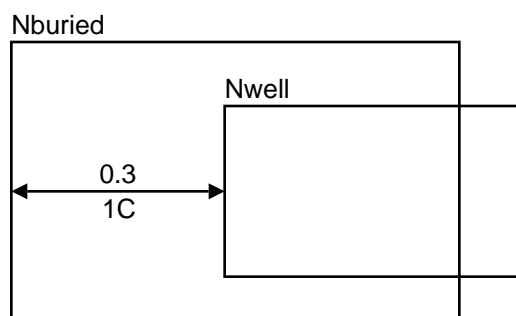
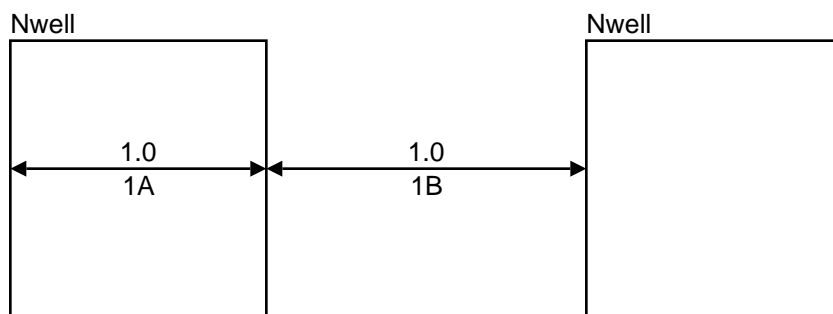
## Nburied rules

Rule No.	Description	Rule (um)
0A	Minimum width of an Nburied.	1.0
0B	Minimum space between two Nburied.	1.0



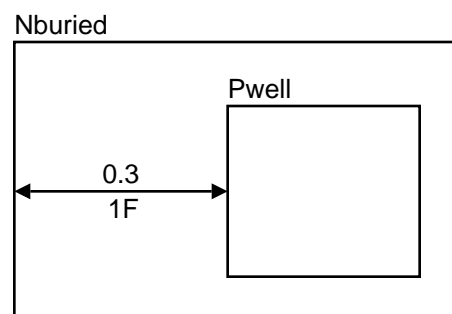
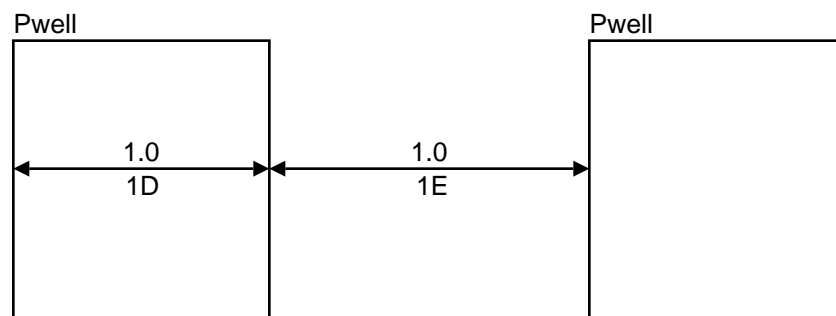
## Nwell rules

Rule No.	Description	Rule (um)
1A	Minimum width of an Nwell.	1.0
1B	Minimum space between two Nwell.	1.0
1C	Minimum Nburied overlap of Nwell	0.3



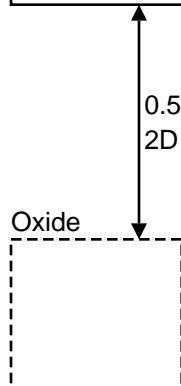
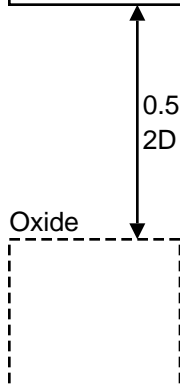
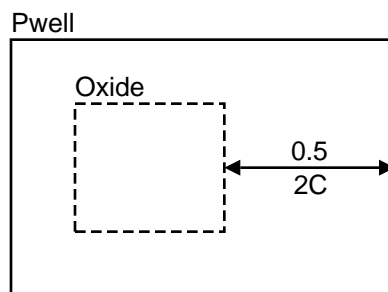
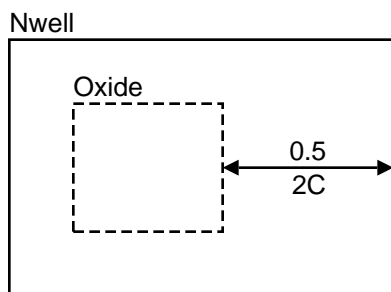
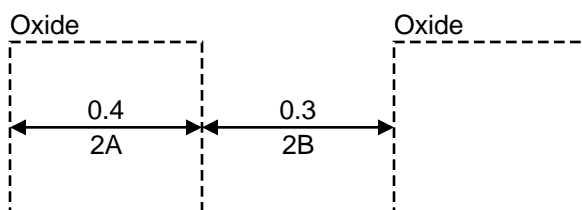
## Pwell rules

Rule No.	Description	Rule (um)
1D	Minimum width of an Pwell.	1.0
1E	Minimum space between two Pwell.	1.0
1F	Minimum Nburied overlap of Pwell	0.3



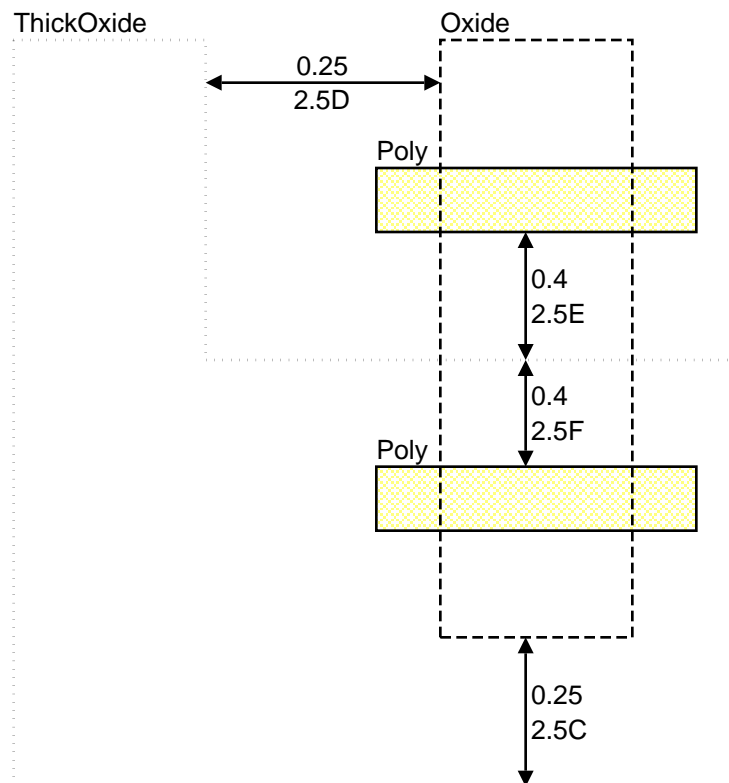
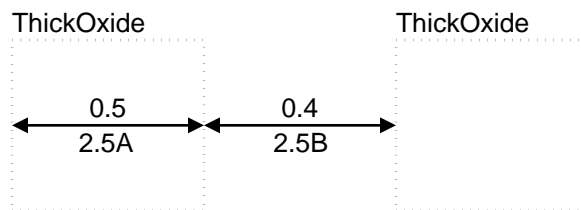
## Oxide Rules

Rule No.	Description	Rule (um)
2A	Minimum width of an Oxide.	0.4
2B	Minimum space between two Oxides.	0.3
2C	Minimum Nwell overlap of Oxide.	0.5
2C	Minimum Pwell overlap of Oxide.	0.5
2D	Minimum spacing between Nwell and Oxide.	0.5
2D	Minimum spacing between Pwell and Oxide.	0.5



## Thick Oxide Rules

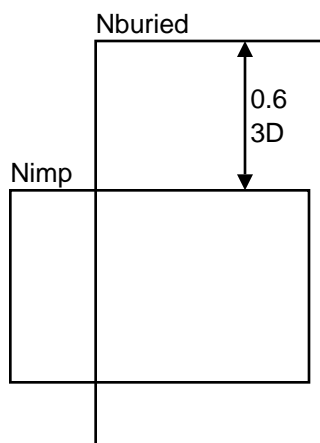
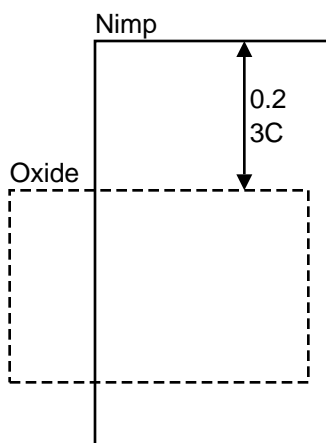
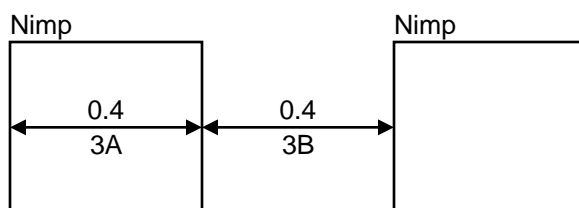
Rule No.	Description	Rule (um)
2.5A	Minimum width of an Oxide.	0.5
2.5B	Minimum space between two Oxides.	0.4
2.5C	Thick Oxide to Oxide enclosure.	0.25
2.5D	ThickOxide to Oxide spacing.	0.25
2.5E	ThickOxide to Poly spacing.	0.4
2.5F	ThickOxide to Poly enclosure.	0.4





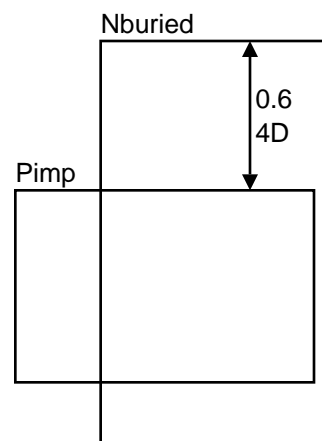
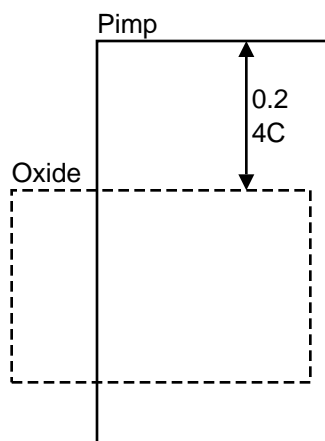
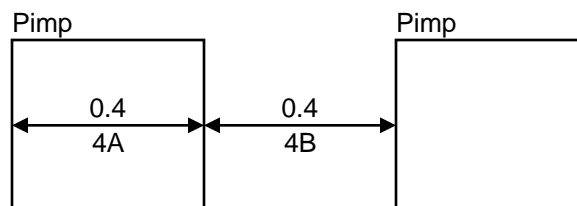
## N+ Implant rules

Rule No.	Description	Rule (um)
3A	Minimum width of an Nimp.	0.4
3B	Minimum space between two Nimp.	0.4
3C	Minimum Nimp overlap of Oxide.	0.2
3D	Minimum Nburied overlap of Nimp.	0.6



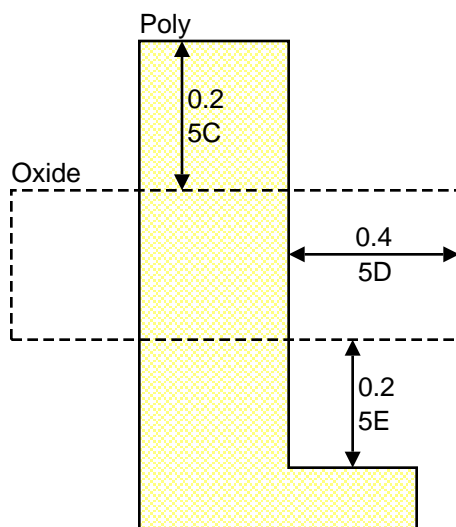
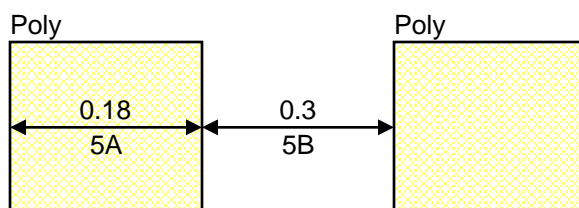
## P+ Implant rules

Rule No.	Description	Rule (um)
4A	Minimum width of an Pimp.	0.4
4B	Minimum space between two Pimp.	0.4
4C	Minimum Pimp overlap of Oxide.	0.2
4D	Minimum Nburied overlap of Pimp.	0.6



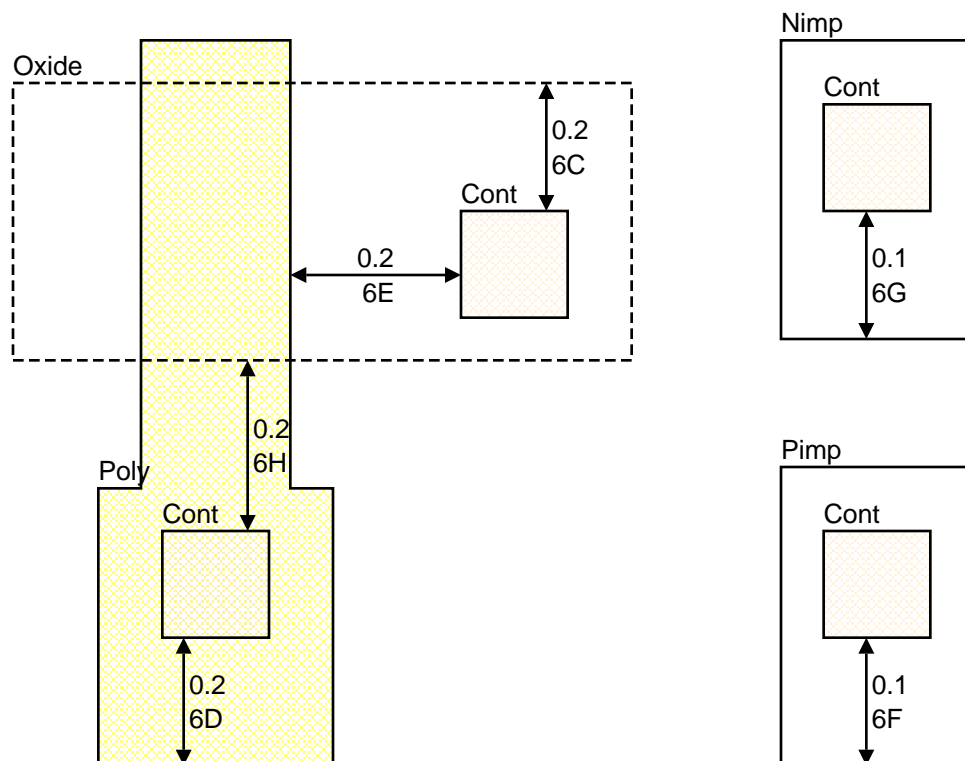
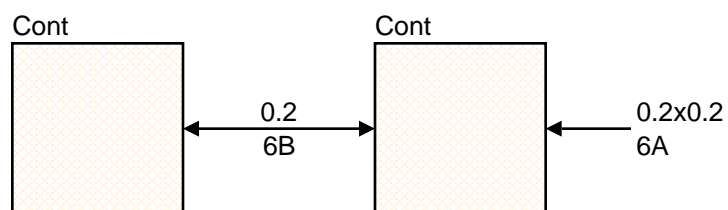
## POLY Rules

Rule No.	Description	Rule (um)
5A	Minimum width of an Poly.	0.18
5B	Minimum space between two Poly.	0.3
5C	Minimum Poly extension over Oxide.	0.2
5D	Minimum Oxide extension over Poly.	0.4
5E	Minimum Poly space to Oxide.	0.2



## Contact rules

Rule No.	Description	Rule (um)
6A	Absolute width and length of Contact.	0.2x0.2
6B	Minimum space between two Contact.	0.2
6C	Minimum Oxide overlap of Contact.	0.2
6D	Minimum Poly overlap of Contact.	0.2
6E	Minimum Poly space to Contact.	0.2
6F	Minimum Pimp overlap of Contact.	0.1
6G	Minimum Nimp overlap of Contact.	0.1
6H	Minimum Contact space to Oxide.	0.2



## Metal rules

(Metalx, x = 1, 2, 3, 4, 5, 6)

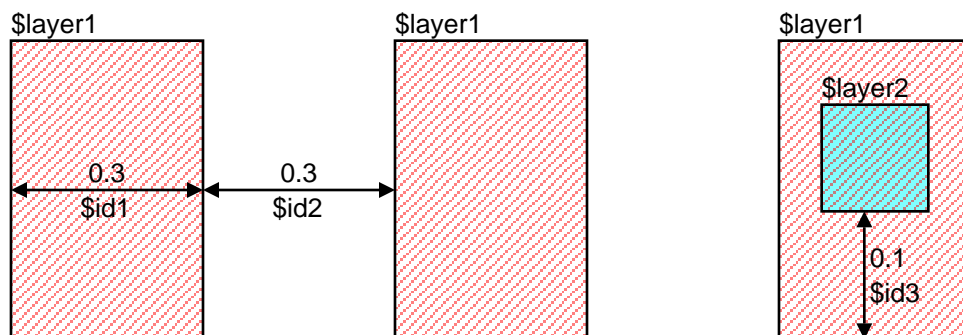
(Viax, x = 1, 2, 3, 4, 5)

Rule No.	Description	Rule (um)
7A, 9A, 11A, 15A, 17A, 19A	Minimum width of an Metalx.	0.3
7B, 9B, 11B, 15B, 17B, 19B	Minimum space between two Metalx.	0.3
7C	Minimum Metal1 overlap of Contact.	0.1
9C, 11C, 15C, 17C, 19C	Minimum Metalx overlap of Viax.	0.1

macro

Macro Table

\$layer1	\$layer2	\$id1	\$id2	\$id3
Metal1	Cont	7A	7B	7C
Metal2	Via1	9A	9B	9C
Metal3	Via2NoCapInd	11A	11B	11C
Metal4	Via3	15A	15B	15C
Metal5	Via4	17A	17B	17C
Metal6	Via5	19A	19B	19C



## VIA rules

(Metalx, x = 1, 2, 3, 4, 5, 6)

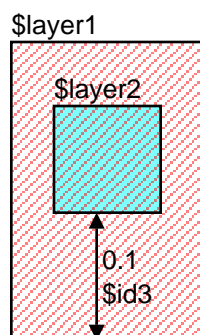
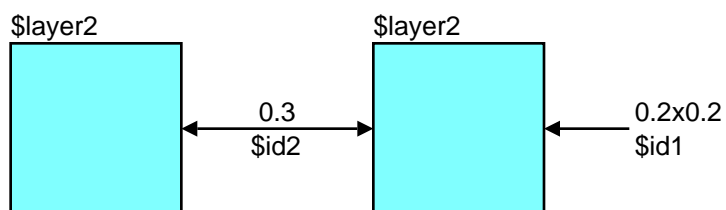
(Viax, x = 1, 2, 3, 4, 5)

Rule No.	Description	Rule (um)
8,10,14,16,18A	Absolute width and Length of Viax.	0.2x0.2
8,10,14,16,18B	Minimum space between two Viax.	0.3
8,10,14,16,18C	Minimum Metalx overlap of Viax.	0.1

macro

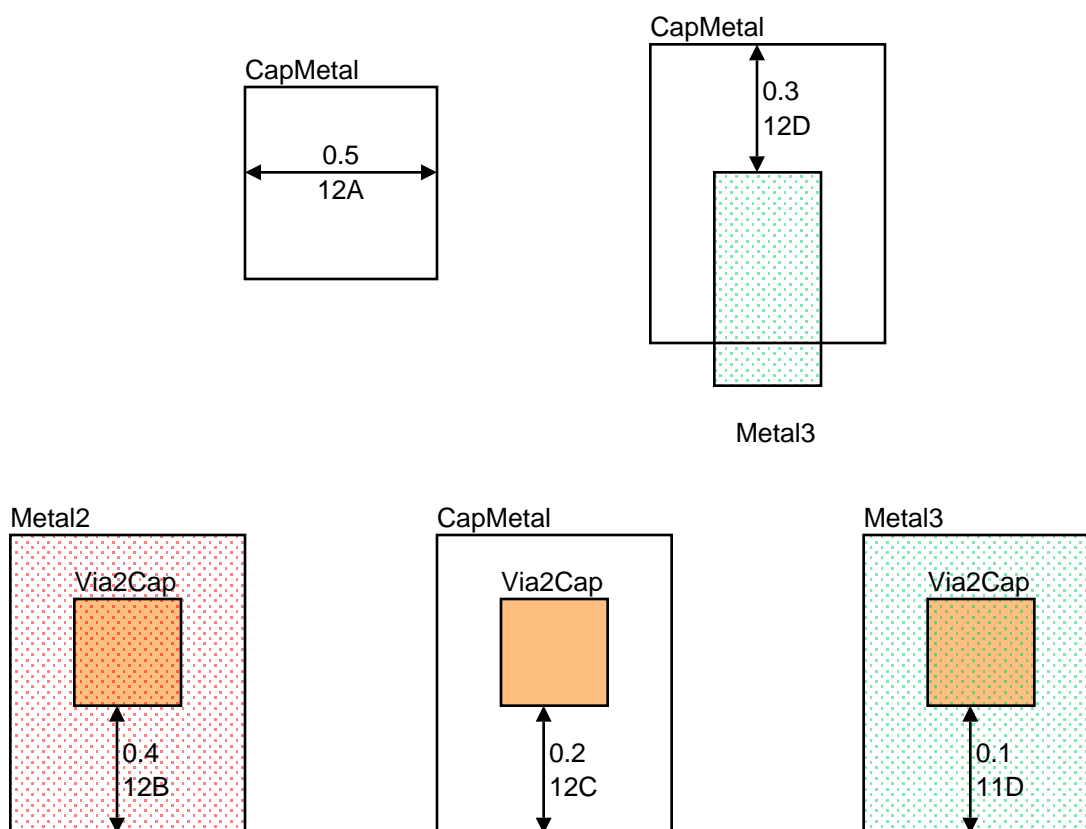
Macro Table

\$layer1	\$layer2	\$id1	\$id2	\$id3
Metal1	Via1	8A	8B	8C
Metal2	Via2	10A	10B	10C
Metal3	Via3	14A	14B	14C
Metal4	Via4	16A	16B	16C
Metal5	Via5	18A	18B	18C



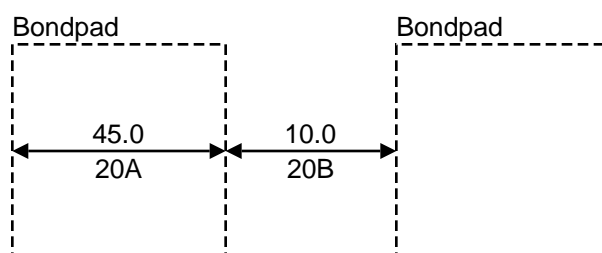
# Capacitor Metal

Rule No.	Description	Rule (um)
12A	Minimum width of CapMetal.	0.5
12B	Minimum Metal2 overlap of Via2 on CapMetal.	0.4
12C	Minimum CapMetal overlap of Via2.	0.2
11D	Minimum Metal3 overlap of Via2 on CapMetal.	0.1
12D	Minimum CapMetal overlap of Metal3.	0.3



## Passivation rules

Rule No.	Description	Rule (um)
20A	Minimum width of Bondpad.	45.0
20B	Minimum space of two Bondpad.	10.0
20C	Metal1 overlap of Bondpad.	3.0
20D	Metal2 overlap of Bondpad.	3.0
20E	Metal3 overlap of Bondpad.	3.0
20F	Metal4 overlap of Bondpad.	3.0
20G	Metal5 overlap of Bondpad.	3.0
20H	Metal6 overlap of Bondpad.	3.0

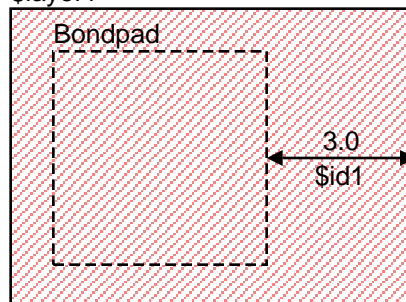


macro

Macro Table

\$layer1	\$id1
Metal1	20C
Metal2	20D
Metal3	20E
Metal4	20F
Metal5	20G
Metal6	20H

\$layer1

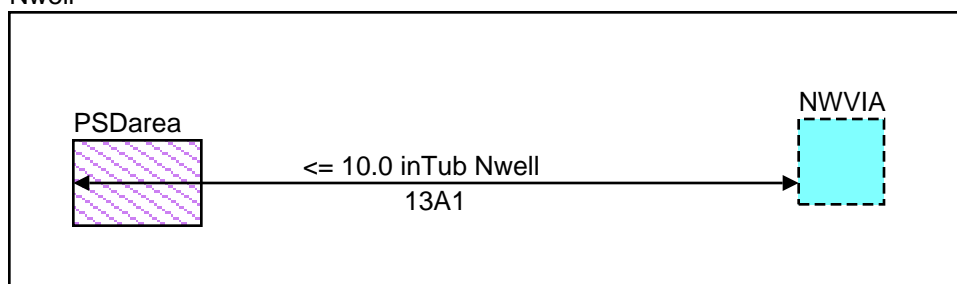




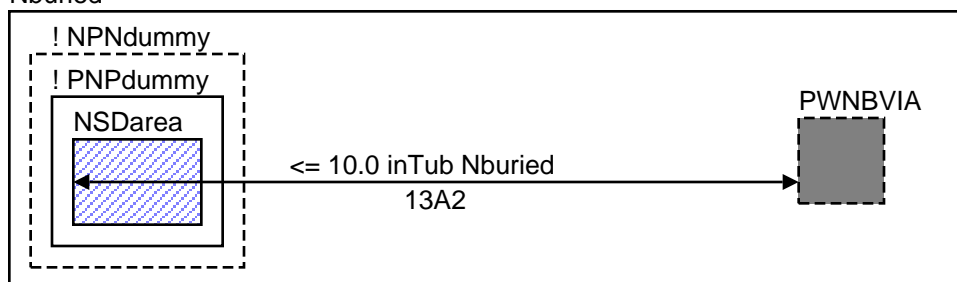
## Latch-up rules

Rule No.	Description	Rule (um)
13A1	The maximum distance from any point inside a source/drain OD area to the nearest Nwell pick-up in the same NW.	10.0
13A2	The maximum distance from any point inside a source/drain OD area to the nearest Nburied pick-up in the same Nburied.	10.0
13A3	The maximum distance from any point inside a source/drain OD area to the nearest Substrate pick-up.	10.0

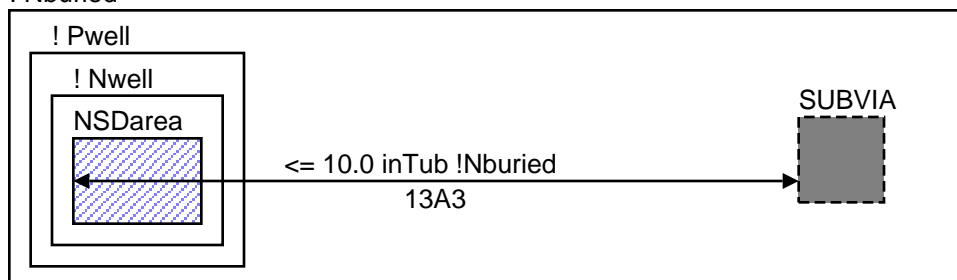
Nwell



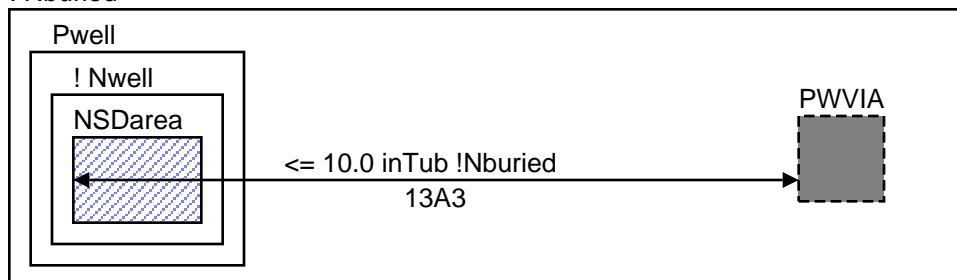
Nburied



! Nburied



! Nburied



## Antenna Rules

Rule No.	Description	Ratio ( )
Antenna	Maximum Field Poly area to the active Poly gate area connected.	100.0
Antenna	Maximum Metal1 area to the active Poly gate area connected.	200.0
Antenna	Maximum Metal2 area to the active Poly gate area connected.	200.0
Antenna	Maximum Metal3 area to the active Poly gate area connected.	200.0
Antenna	Maximum Metal4 area to the active Poly gate area connected.	200.0
Antenna	Maximum Metal5 area to the active Poly gate area connected.	200.0
Antenna	Maximum Metal6 area to the active Poly gate area connected.	200.0

switch CHECK\_ANTENNA

Antenna

Ratio <= 100.0

FIELD\_POLY1

FIELD\_POLY1\_CONNECT

Poly

ALL GATES

Antenna

Ratio <= 200.0

Metal6

Via5

Metal5

Via4

Metal4

Via3

Metal3

Via2NoCapInd

Metal2

Via1

Metal1

Cont

Poly

ALL GATES

Antenna

Ratio <= 200.0

Metal1

Cont

Poly

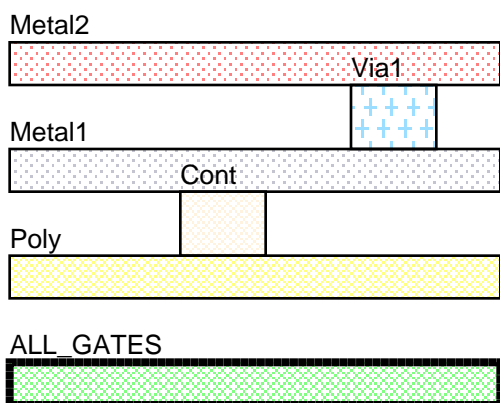
ALL GATES

## Antenna Rules (cont.)

switch CHECK\_ANTENNA

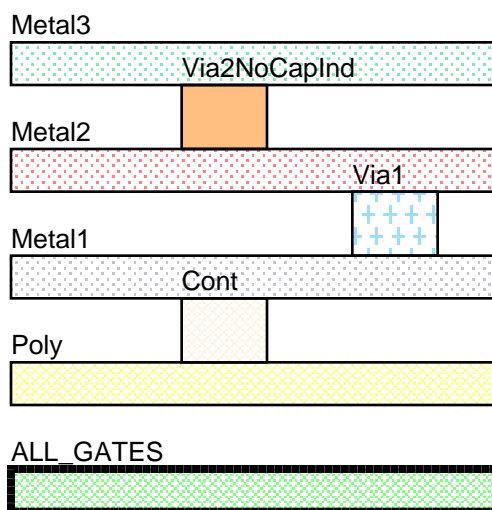
Antenna

Ratio <= 200.0



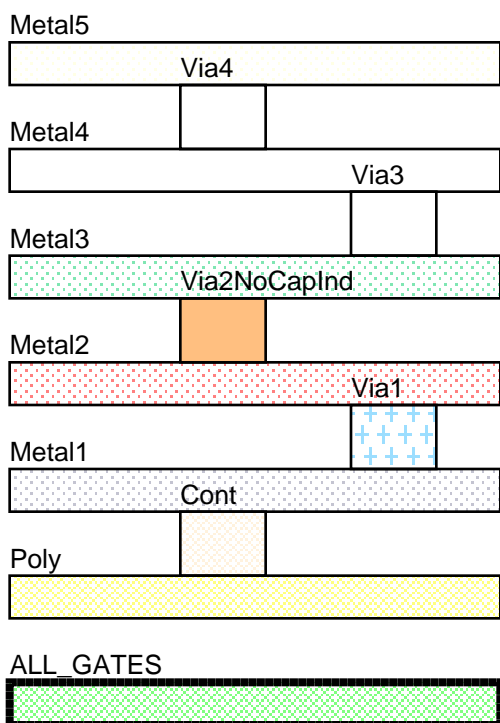
Antenna

Ratio <= 200.0



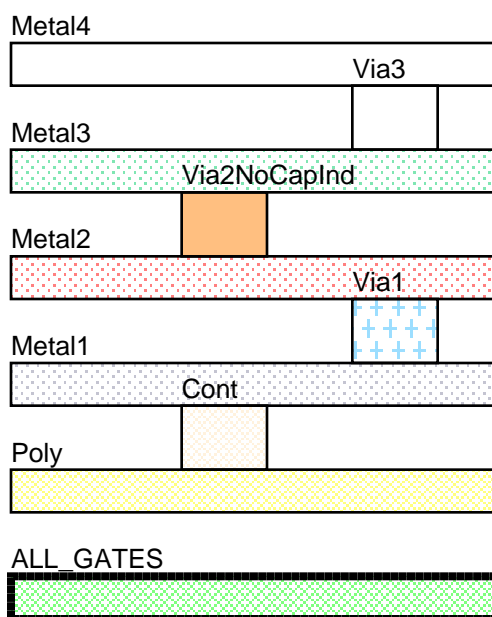
Antenna

Ratio <= 200.0



Antenna

Ratio <= 200.0



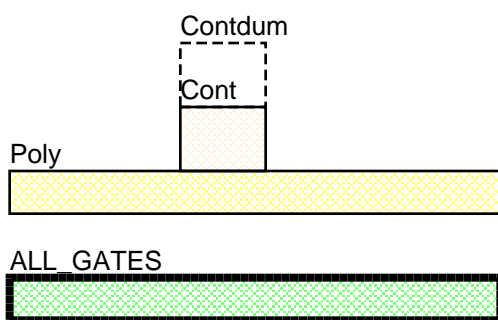
## Antenna Rules (cont.)

Rule No.	Description	Ratio ( )
Antenna	Maximum Contact area to the active Poly gate area connected.	10.0
Antenna	Maximum Via1 area to the active Poly gate area connected.	20.0
Antenna	Maximum Via2 area to the active Poly gate area connected.	20.0
Antenna	Maximum Via3 area to the active Poly gate area connected.	20.0
Antenna	Maximum Via4 area to the active Poly gate area connected.	20.0
Antenna	Maximum Via5 area to the active Poly gate area connected.	20.0

switch CHECK\_ANTENNA

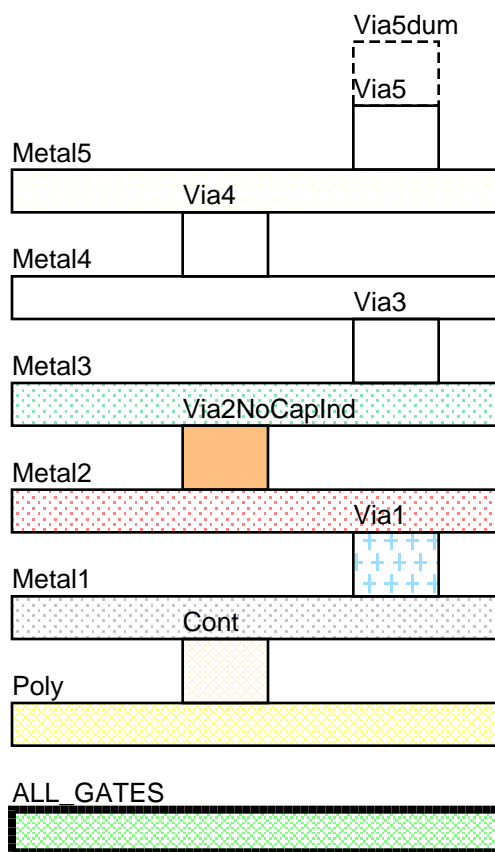
Antenna

Ratio <= 10.0



Antenna

Ratio <= 20.0

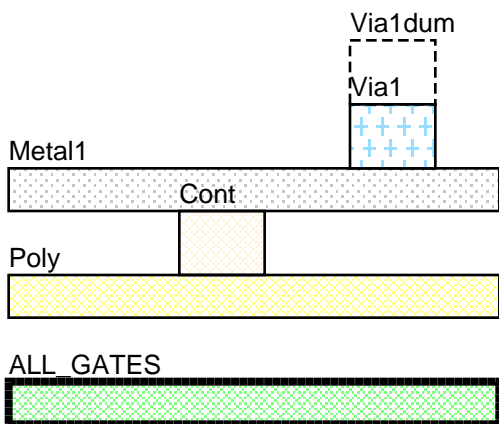


## Antenna Rules (cont.)

switch CHECK\_ANTENNA

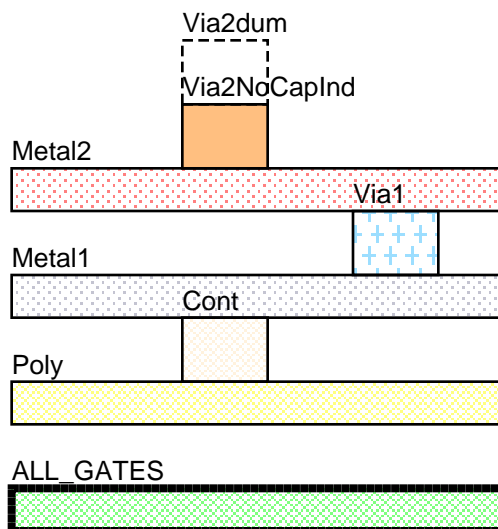
Antenna

Ratio <= 20.0



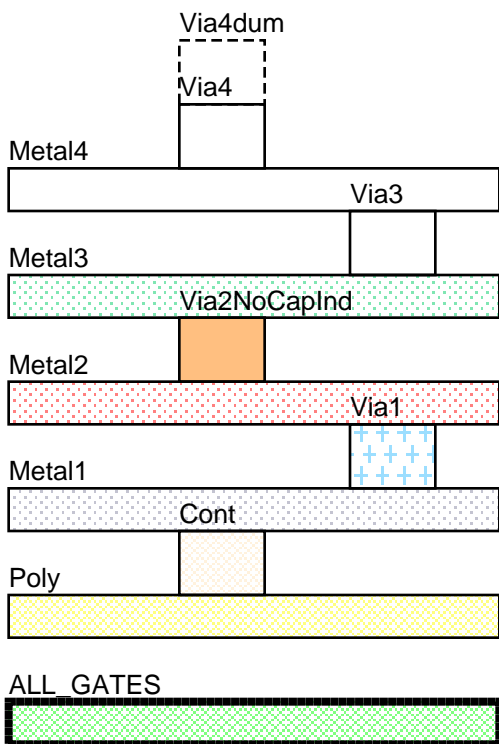
Antenna

Ratio <= 20.0



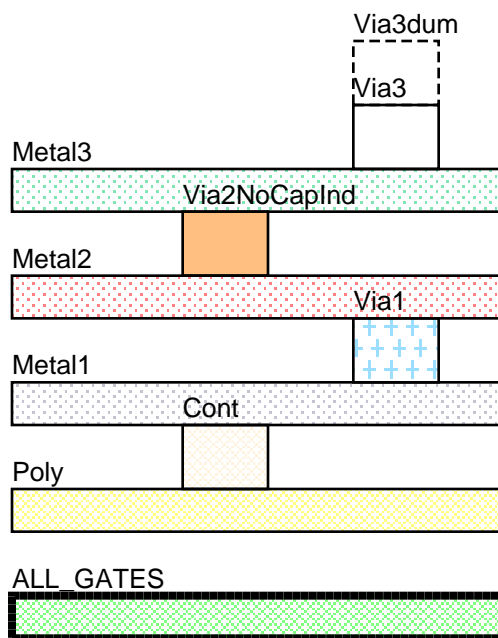
Antenna

Ratio <= 20.0



Antenna

Ratio <= 20.0



## Density rules

Rule No.	Description	Ratio ( )
Density	Minimum Poly density across the full chip.	0.15
Density	Minimum Metal1 density across the full chip.	0.25
Density	Minimum Metal2 density across the full chip.	0.25
Density	Minimum Metal3 density across the full chip.	0.25
Density	Minimum Metal4 density across the full chip.	0.25
Density	Minimum Metal5 density across the full chip.	0.25
Density	Minimum Metal6 density across the full chip.	0.25

switch CHECK\_DENSITY

Density

Ratio  $\geq 0.15$

Poly



Density

Ratio  $\geq 0.25$

Metal1



Density

Ratio  $\geq 0.25$

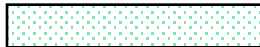
Metal2



Density

Ratio  $\geq 0.25$

Metal3



Density

Ratio  $\geq 0.25$

Metal4



Density

Ratio  $\geq 0.25$

Metal5



Density

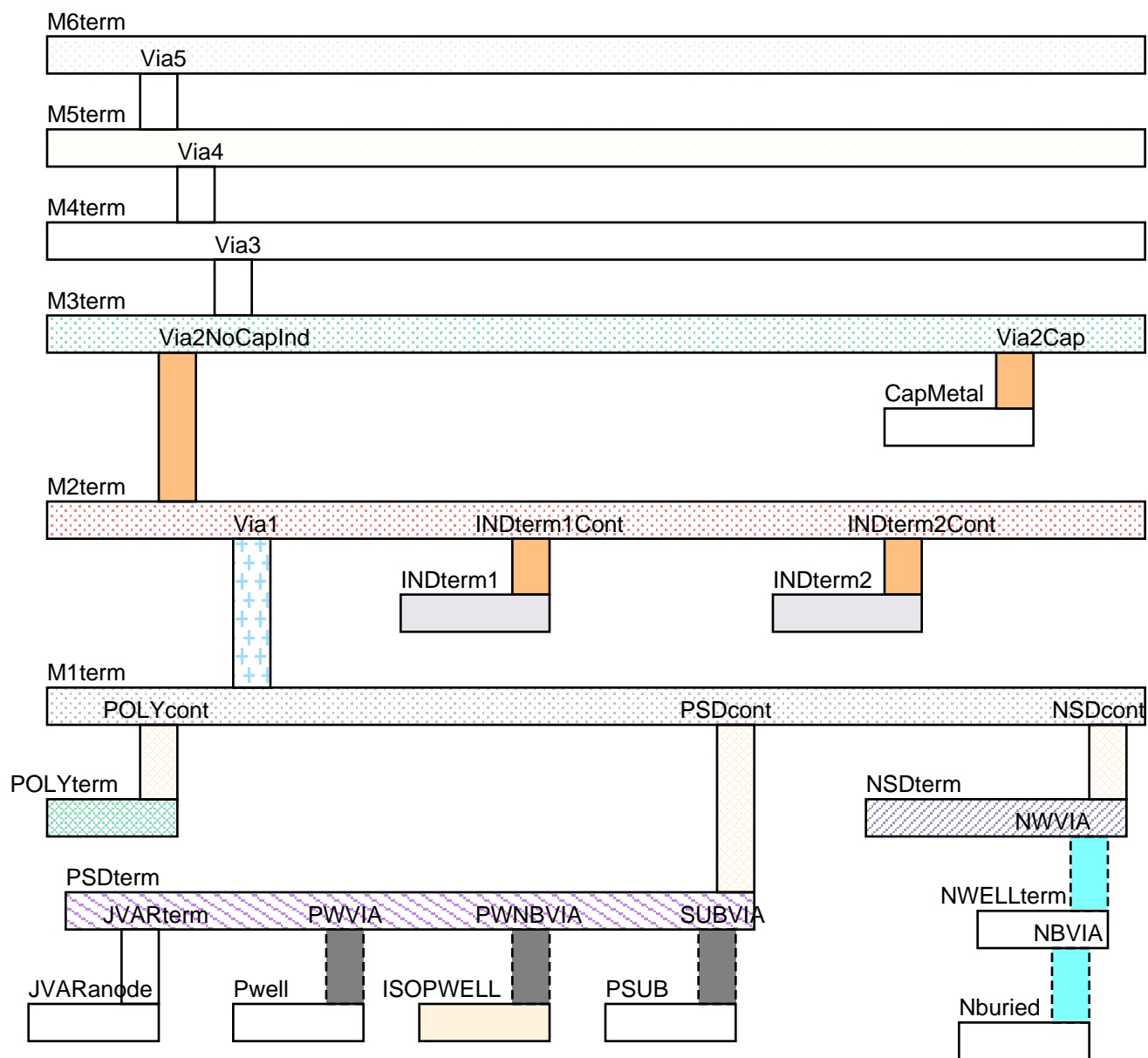
Ratio  $\geq 0.25$

Metal6



# Connectivity Definition

## Connectivity



## **Component LVS Definitions**



# Resistors

## RESISTOR

name polyres      cellview "polyres ivpcell gpd180"  
 type polyres      model polyres ivpcell gpd180

### Device Terminals

<b>term1</b>	PLUS
<b>term2</b>	MINUS

POLYterm



POLYterm

### Device Parameters

<b>w</b>	width*1e-6
<b>l</b>	length*1e-6
<b>sl</b>	length*1e-6
<b>effL</b>	length*1e-6
<b>r</b>	\$poly_rho*length/width

## RESISTOR

name polyhres      cellview "polyhres ivpcell gpd180"  
 type polyhres      model polyhres ivpcell gpd180

### Device Terminals

<b>term1</b>	PLUS
<b>term2</b>	MINUS

POLYterm



POLYterm

### Device Parameters

<b>w</b>	width*1e-6
<b>l</b>	length*1e-6
<b>sl</b>	length*1e-6
<b>effL</b>	length*1e-6
<b>r</b>	\$polyh_rho*length/width

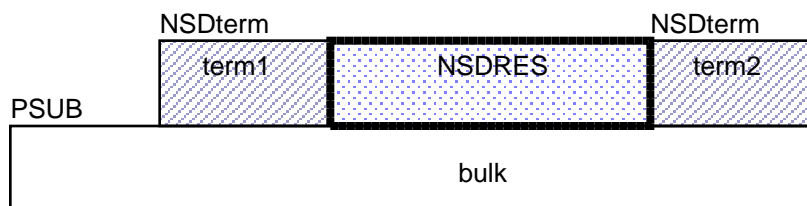
## Resistors (cont.)

### RESISTOR

name nplusres cellview "nplusres ivpcell gpd180"  
 type nplusres model nplusres ivpcell gpd180

#### Device Terminals

<b>term1</b>	PLUS
<b>term2</b>	MINUS
<b>bulk</b>	B



#### Device Parameters

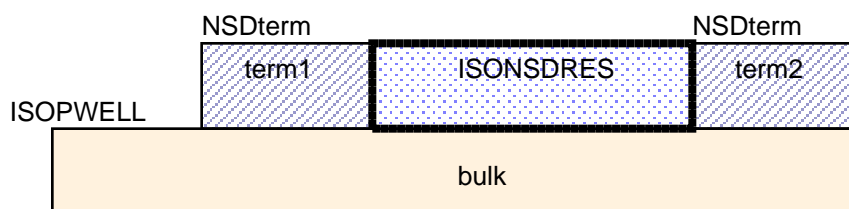
<b>w</b>	width*1e-6
<b>l</b>	length*1e-6
<b>sl</b>	length*1e-6
<b>effL</b>	length*1e-6
<b>r</b>	\$nplus_rho*length/width

### RESISTOR

name nplusres cellview "nplusres ivpcell gpd180"  
 type nplusres model nplusres ivpcell gpd180

#### Device Terminals

<b>term1</b>	PLUS
<b>term2</b>	MINUS
<b>bulk</b>	B



#### Device Parameters

<b>w</b>	width*1e-6
<b>l</b>	length*1e-6
<b>sl</b>	length*1e-6
<b>effL</b>	length*1e-6
<b>r</b>	\$nplus_rho*length/width

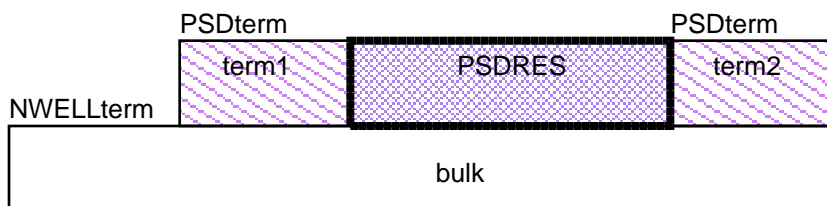
## Resistors (cont.)

### RESISTOR

name pplusres cellview "pplusres ivpcell gpd180"  
 type pplusres model pplusres ivpcell gpd180

#### Device Terminals

<b>term1</b>	PLUS
<b>term2</b>	MINUS
<b>bulk</b>	B



#### Device Parameters

<b>w</b>	width*1e-6
<b>l</b>	length*1e-6
<b>sl</b>	length*1e-6
<b>effL</b>	length*1e-6
<b>r</b>	\$pplus_rho*length/width

### RESISTOR

name nwellres cellview "nwellres ivpcell gpd180"  
 type nwellres model nwellres ivpcell gpd180

#### Device Terminals

<b>term1</b>	PLUS
<b>term2</b>	MINUS



#### Device Parameters

<b>w</b>	width*1e-6
<b>l</b>	length*1e-6
<b>sl</b>	length*1e-6
<b>effL</b>	length*1e-6
<b>r</b>	\$nwell_rho*length/width

## Resistors (cont.)

macro

Macro Table

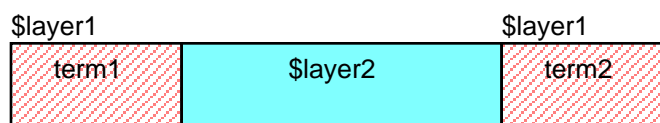
\$cell	\$rho	\$layer1	\$layer2
m1res	\$m1_rho	M1term	M1res
m2res	\$m2_rho	M2term	M2res
m3res	\$m3_rho	M3term	M3res
m4res	\$m4_rho	M4term	M4res
m5res	\$m5_rho	M5term	M5res
m6res	\$m6_rho	M6term	M6res

### RESISTOR

name \$cell                      cellview "\$cell ivpcell gpd180"  
 type \$cell                      model \$cell ivpcell gpd180

Device Terminals

<b>term1</b>	PLUS
<b>term2</b>	MINUS



Device Parameters

<b>w</b>	width*1e-6
<b>l</b>	length*1e-6
<b>sl</b>	length*1e-6
<b>effL</b>	length*1e-6
<b>r</b>	$\$rho * \text{length} / \text{width}$

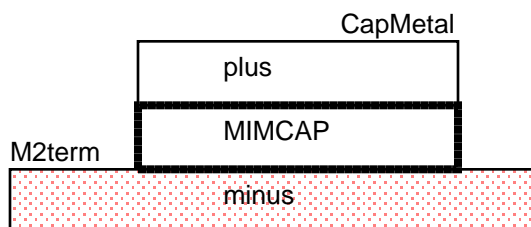
# Capacitors

## CAPACITOR

name mimcap      cellview "mimcap ivpcell gpd180"  
 type mimcap      model mimcap ivpcell gpd180

### Device Terminals

<b>plus</b>	PLUS
<b>minus</b>	MINUS



### Device Parameters

<b>w</b>	$((\text{perimeter} * 0.25) + 0.5 * \sqrt{(0.25 * \text{perimeter}^2) - (4 * \text{area})}) * 1e-6$
<b>l</b>	$((\text{perimeter} * 0.25) - 0.5 * \sqrt{(0.25 * \text{perimeter}^2) - (4 * \text{area})}) * 1e-6$
<b>c</b>	$\text{area} * 1e-15 + \text{perimeter} * 1e-16$
<b>area</b>	$\text{area} * 1e-12$
<b>perim</b>	$\text{perimeter} * 1e-6$

## PMOS

name pmoscap      cellview "pmoscap ivpcell gpd180"  
 type pmoscap      model pmoscap ivpcell gpd180

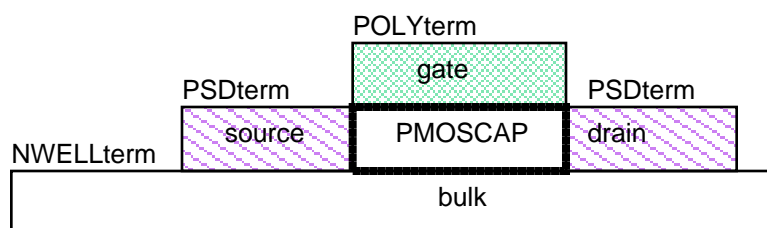
### Device Terminals

<b>source</b>	S
<b>gate</b>	G
<b>drain</b>	D
<b>bulk</b>	B

DESCRIPTION:  
 Pchannel device  
 configured as capacitor

### Device Parameters

<b>w</b>	$\text{width} * 1e-6$
<b>fw</b>	$\text{width} * 1e-6$
<b>simW</b>	$\text{width} * 1e-6$
<b>l</b>	$\text{length} * 1e-6$



## Capacitors (cont.)

### NMOS

name nmoscap cellview "nmoscap ivpcell gpd180"  
 type nmoscap model nmoscap ivpcell gpd180

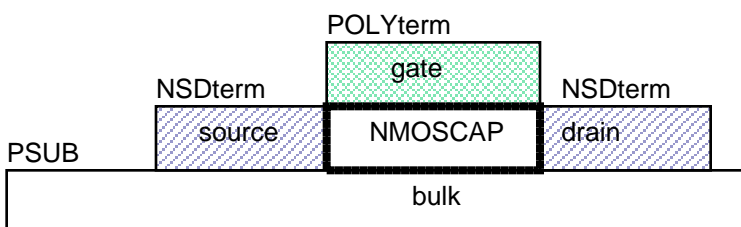
#### Device Terminals

<b>source</b>	S
<b>gate</b>	G
<b>drain</b>	D
<b>bulk</b>	B

DESCRIPTION:  
 Nchannel device  
 configured as capacitor

#### Device Parameters

<b>w</b>	width*1e-6
<b>fw</b>	width*1e-6
<b>simW</b>	width*1e-6
<b>l</b>	length*1e-6



### NMOS

name nmoscap cellview "nmoscap ivpcell gpd180"  
 type nmoscap model nmoscap ivpcell gpd180

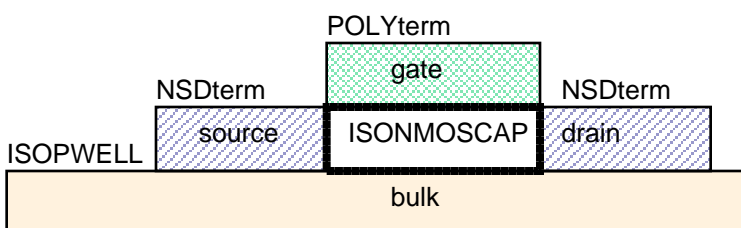
#### Device Terminals

<b>source</b>	S
<b>gate</b>	G
<b>drain</b>	D
<b>bulk</b>	B

DESCRIPTION:  
 Nchannel device  
 configured as capacitor

#### Device Parameters

<b>w</b>	width*1e-6
<b>fw</b>	width*1e-6
<b>simW</b>	width*1e-6
<b>l</b>	length*1e-6



# Inductor

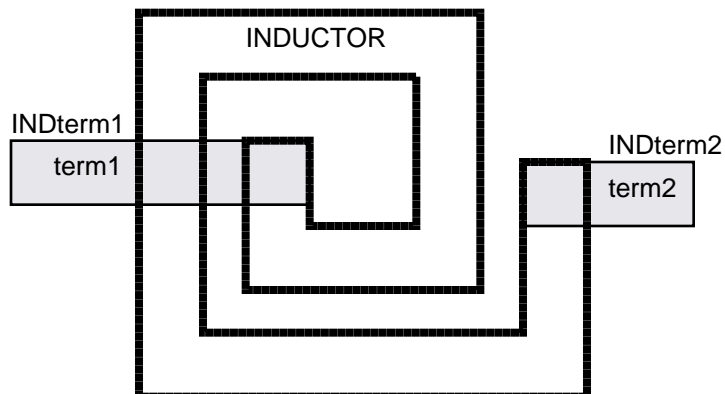
## BOX

name ind                      cellview "ind ivpcell gpd180"  
 type ind                      model ind ivpcell gpd180

### Device Terminals

<b>term1</b>	PLUS
<b>term2</b>	MINUS

DESCRIPTION:  
 Inductor. Modelled as Box for  
 Dracula



Bipolars

PNP

name vnpn                      cellview "vnpn ivpcell gpdk180"  
type vnpn                      model vnpn ivpcell gpdk180

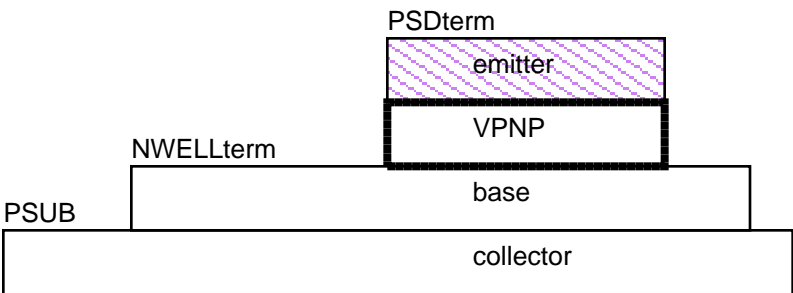
Device Terminals

collector	C
base	B
emitter	E

DESCRIPTION:  
Substrate PNP. collector  
connects...

Device Parameters

area	area
------	------



PNP

name pnp                      cellview "pnp ivpcell gpdk180"  
type pnp                      model pnp ivpcell gpdk180

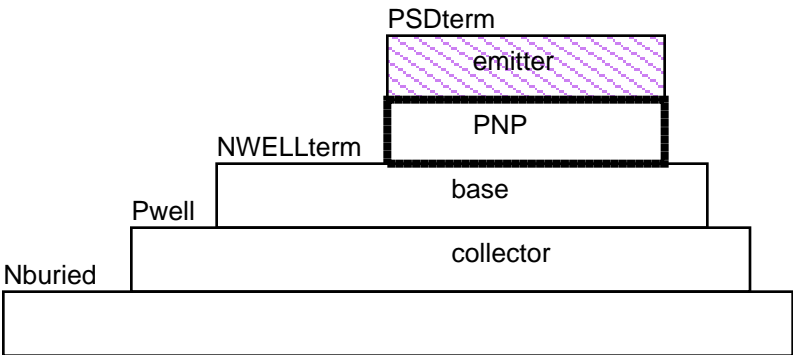
Device Terminals

collector	C
base	B
emitter	E

DESCRIPTION:  
Bipolar PNP.

Device Parameters

area	area
------	------





## Bipolars (cont.)

### NPN

name npn                      cellview "npn ivpcell gpd180"  
 type npn                      model npn ivpcell gpd180

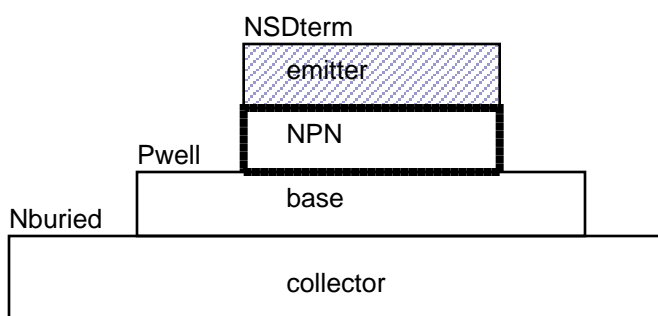
#### Device Terminals

<b>collector</b>	C
<b>base</b>	B
<b>emitter</b>	E

DESCRIPTION:  
 Bipolar NPN.

#### Device Parameters

<b>area</b>	area
-------------	------



# Nmos

## NMOS

name nmos cellview "nmos ivpcell gpd180"  
type nmos model nmos ivpcell gpd180

### Device Terminals

<b>source</b>	S
<b>gate</b>	G
<b>drain</b>	D
<b>bulk</b>	B

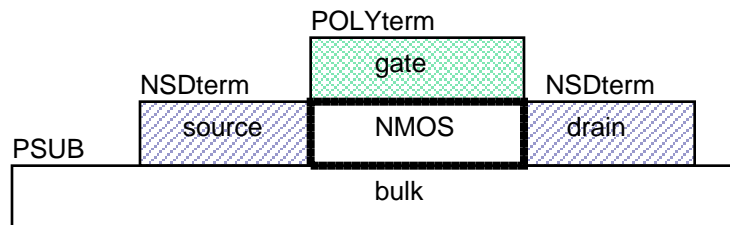
DESCRIPTION:  
Nchannel device

### Device Parameters

<b>w</b>	width*1e-6
<b>fw</b>	width*1e-6
<b>simW</b>	width*1e-6
<b>l</b>	length*1e-6

### assuraLVS Native Code

```
sd_area=measureParameter(area (NSDterm))
sd_perimeter=measureParameter(perimeter (NSDterm))
sdarea=calculateParameter(sd_area*1.0e-12)
sdperimeter=calculateParameter(sd_perimeter*1.0e-6)
attachParameter(sdarea ("as" "S")("ad" "D") NMOS shared)
attachParameter(sdperimeter ("ps" "S")("pd" "D") NMOS shared)
```



## NMOS

name nmos cellview "nmos ivpcell gpd180"  
type nmos model nmos ivpcell gpd180

### Device Terminals

<b>source</b>	S
<b>gate</b>	G
<b>drain</b>	D
<b>bulk</b>	B

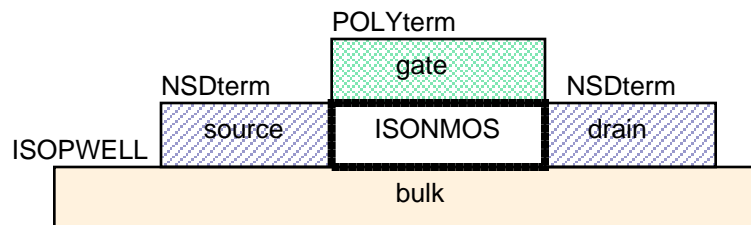
DESCRIPTION:  
Isolated Nchannel device

### Device Parameters

<b>w</b>	width*1e-6
<b>fw</b>	width*1e-6
<b>simW</b>	width*1e-6
<b>l</b>	length*1e-6

### assuraLVS Native Code

```
sd_area=measureParameter(area (NSDterm))
sd_perimeter=measureParameter(perimeter (NSDterm))
sdarea=calculateParameter(sd_area*1.0e-12)
sdperimeter=calculateParameter(sd_perimeter*1.0e-6)
attachParameter(sdarea ("as" "S")("ad" "D") ISONMOS shared)
attachParameter(sdperimeter ("ps" "S")("pd" "D") ISONMOS...)
```



# HV Nmos

## NMOS

name nmoshv cellview "nmoshv ivpcell gpdk180"  
type nmoshv model nmoshv ivpcell gpdk180

### Device Terminals

source	S
gate	G
drain	D
bulk	B

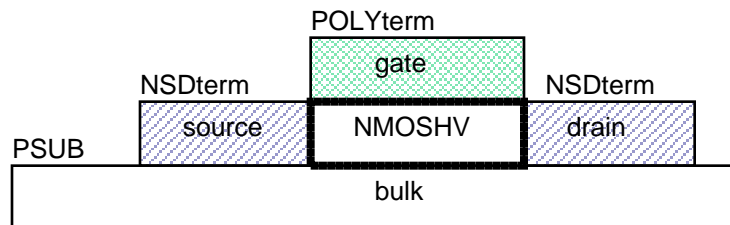
DESCRIPTION:  
Nchannel HV device

### Device Parameters

w	width*1e-6
fw	width*1e-6
simW	width*1e-6
l	length*1e-6

### assuraLVS Native Code

```
sd_area=measureParameter(area (NSDterm))
sd_perimeter=measureParameter(perimeter (NSDterm))
sdarea=calculateParameter(sd_area*1.0e-12)
sdperimeter=calculateParameter(sd_perimeter*1.0e-6)
attachParameter(sdarea ("as" "S")("ad" "D") NMOSHV shared)
attachParameter(sdperimeter ("ps" "S")("pd" "D") NMOSHV...
```



## NMOS

name nmoshv cellview "nmoshv ivpcell gpdk180"  
type nmoshv model nmoshv ivpcell gpdk180

### Device Terminals

source	S
gate	G
drain	D
bulk	B

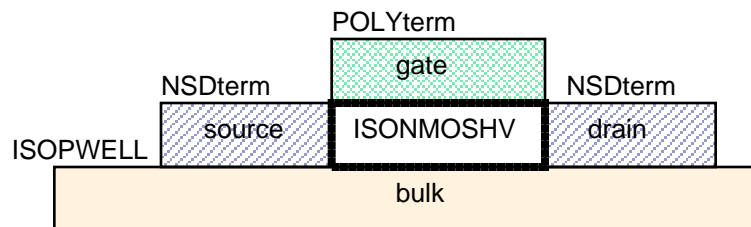
DESCRIPTION:  
Isolated Nchannel HV  
device

### Device Parameters

w	width*1e-6
fw	width*1e-6
simW	width*1e-6
l	length*1e-6

### assuraLVS Native Code

```
sd_area=measureParameter(area (NSDterm))
sd_perimeter=measureParameter(perimeter (NSDterm))
sdarea=calculateParameter(sd_area*1.0e-12)
sdperimeter=calculateParameter(sd_perimeter*1.0e-6)
attachParameter(sdarea ("as" "S")("ad" "D") ISONMOSHV shared)
attachParameter(sdperimeter ("ps" "S")("pd" "D") ISONMOSHV...
```



# RF Nmos

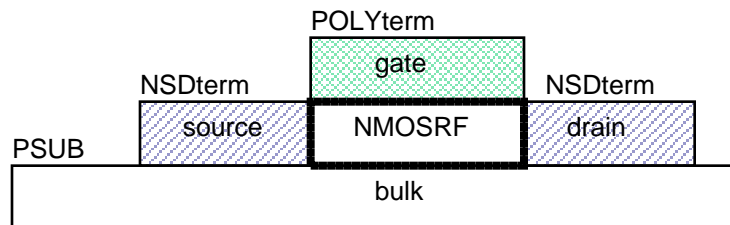
## NMOS

name nmosrf cellview "nmosrf ivpcell gpd180"  
type nmosrf model nmosrf ivpcell gpd180

### Device Terminals

<b>source</b>	S
<b>gate</b>	G
<b>drain</b>	D
<b>bulk</b>	B

DESCRIPTION:  
Nchannel RF device



### assuraLVS Native Code

### Device Parameters

<b>w</b>	width*1e-6
<b>fw</b>	width*1e-6
<b>simW</b>	width*1e-6
<b>l</b>	length*1e-6

```
sd_area=measureParameter(area (NSDterm))
sd_perimeter=measureParameter(perimeter (NSDterm))
sdarea=calculateParameter(sd_area*1.0e-12)
sdperimeter=calculateParameter(sd_perimeter*1.0e-6)
attachParameter(sdarea ("as" "S")("ad" "D") NMOSRF shared)
attachParameter(sdperimeter ("ps" "S")("pd" "D") NMOSRF shared)
```

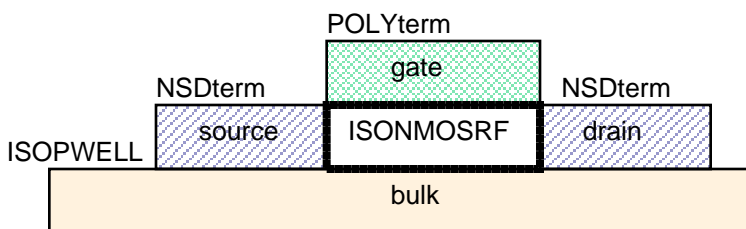
## NMOS

name nmosrf cellview "nmosrf ivpcell gpd180"  
type nmosrf model nmosrf ivpcell gpd180

### Device Terminals

<b>source</b>	S
<b>gate</b>	G
<b>drain</b>	D
<b>bulk</b>	B

DESCRIPTION:  
Isolated Nchannel RF  
device



### assuraLVS Native Code

### Device Parameters

<b>w</b>	width*1e-6
<b>fw</b>	width*1e-6
<b>simW</b>	width*1e-6
<b>l</b>	length*1e-6

```
sd_area=measureParameter(area (NSDterm))
sd_perimeter=measureParameter(perimeter (NSDterm))
sdarea=calculateParameter(sd_area*1.0e-12)
sdperimeter=calculateParameter(sd_perimeter*1.0e-6)
attachParameter(sdarea ("as" "S")("ad" "D") ISONMOSRF shared)
attachParameter(sdperimeter ("ps" "S")("pd" "D") ISONMOSRF shared)
```

# Pmos and HV Pmos

## PMOS

name pmos cellview "pmos ivpcell gpd180"  
type pmos model pmos ivpcell gpd180

### Device Terminals

source	S
gate	G
drain	D
bulk	B

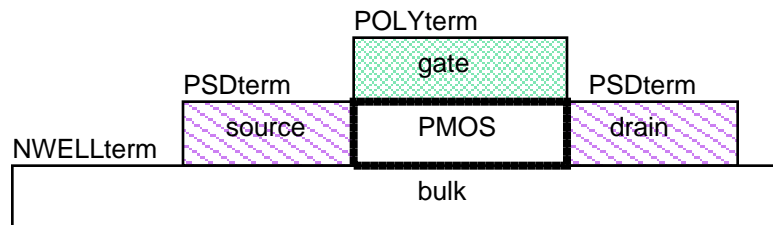
DESCRIPTION:  
Pchannel device

### Device Parameters

w	width*1e-6
fw	width*1e-6
simW	width*1e-6
l	length*1e-6

### assuraLVS Native Code

```
sd_area=measureParameter(area (PSDterm))
sd_perimeter=measureParameter(perimeter (PSDterm))
sdarea=calculateParameter(sd_area*1.0e-12)
sdperimeter=calculateParameter(sd_perimeter*1.0e-6)
attachParameter(sdarea ("as" "S")("ad" "D") PMOS shared)
attachParameter(sdperimeter ("ps" "S")("pd" "D") PMOS shared)
```



## PMOS

name pmoshv cellview "pmoshv ivpcell gpd180"  
type pmoshv model pmoshv ivpcell gpd180

### Device Terminals

source	S
gate	G
drain	D
bulk	B

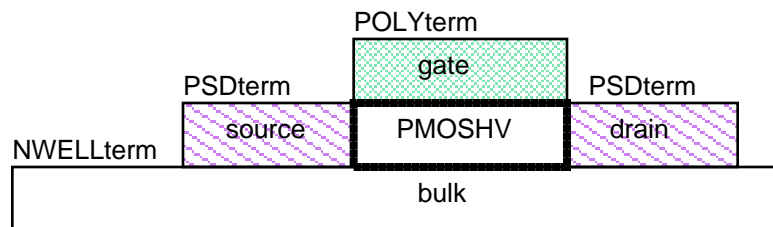
DESCRIPTION:  
Pchannel HV device

### Device Parameters

w	width*1e-6
fw	width*1e-6
simW	width*1e-6
l	length*1e-6

### assuraLVS Native Code

```
sd_area=measureParameter(area (PSDterm))
sd_perimeter=measureParameter(perimeter (PSDterm))
sdarea=calculateParameter(sd_area*1.0e-12)
sdperimeter=calculateParameter(sd_perimeter*1.0e-6)
attachParameter(sdarea ("as" "S")("ad" "D") PMOSHV shared)
attachParameter(sdperimeter ("ps" "S")("pd" "D") PMOSHV shared)
```



# RF Pmos

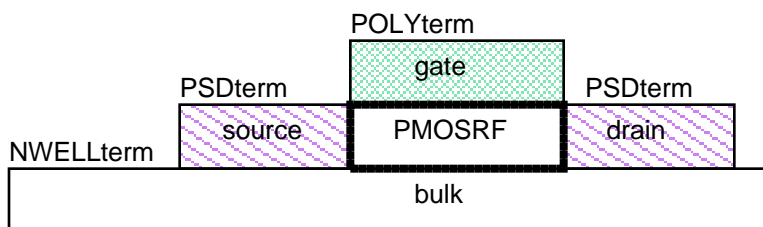
## PMOS

name pmosrf                      cellview "pmosrf ivpcell gpd180"  
 type pmosrf                    model pmosrf ivpcell gpd180

### Device Terminals

<b>source</b>	S
<b>gate</b>	G
<b>drain</b>	D
<b>bulk</b>	B

DESCRIPTION:  
 Pchannel RF device



### assuraLVS Native Code

### Device Parameters

<b>w</b>	width*1e-6
<b>fw</b>	width*1e-6
<b>simW</b>	width*1e-6
<b>l</b>	length*1e-6

```
sd_area=measureParameter(area (PSDterm))
sd_perimeter=measureParameter(perimeter (PSDterm))
sdarea=calculateParameter(sd_area*1.0e-12)
sdperimeter=calculateParameter(sd_perimeter*1.0e-6)
attachParameter(sdarea ("as" "S")("ad" "D") PMOSRF shared)
attachParameter(sdperimeter ("ps" "S")("pd" "D") PMOSRF shared)
```

Diodes

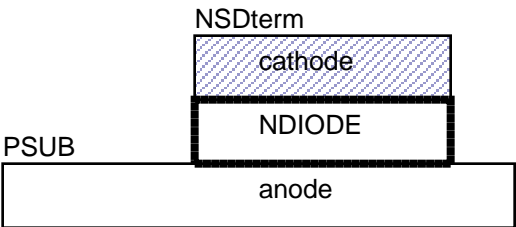
DIODE

name ndio            cellview "ndio ivpcell gpd180"  
type ndio            model ndio ivpcell gpd180

Device Terminals

<b>anode</b>	PLUS
<b>cathode</b>	MINUS

N type Diode



Device Parameters

<b>area</b>	area*1e-12
-------------	------------

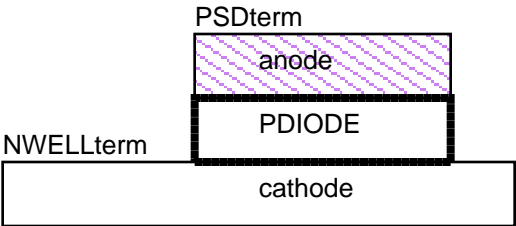
DIODE

name pdio            cellview "pdio ivpcell gpd180"  
type pdio            model pdio ivpcell gpd180

Device Terminals

<b>anode</b>	PLUS
<b>cathode</b>	MINUS

P type Diode



Device Parameters

<b>area</b>	area*1e-12
-------------	------------

# Junction Varactors

## DIODE

name xjvar\_w40 cellview "xjvar\_w40 ivpcell gpdk180"  
 type xjvar\_w40 model xjvar\_w40 ivpcell gpdk180

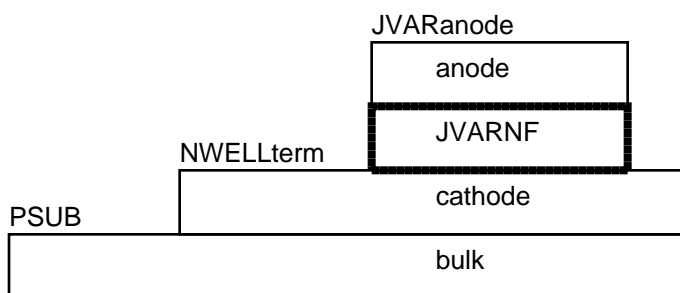
### Device Terminals

<b>anode</b>	ANODE
<b>cathode</b>	CATHODE
<b>bulk</b>	BULK

P type Diode

### Device Parameters

<b>nf</b>	(perimeter - 85.2) / 3.6
-----------	--------------------------



## DIODE

name xjvar\_nf36 cellview "xjvar\_nf36 ivpcell gpdk180"  
 type xjvar\_nf36 model xjvar\_nf36 ivpcell gpdk180

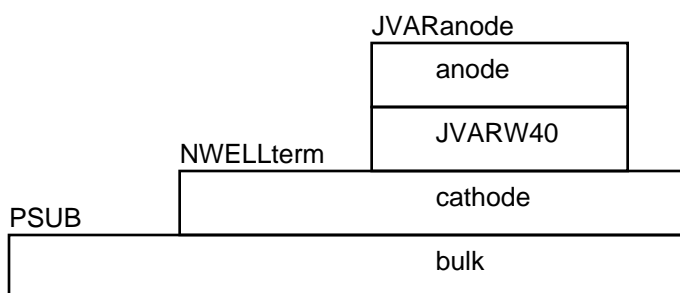
### Device Terminals

<b>anode</b>	ANODE
<b>cathode</b>	CATHODE
<b>bulk</b>	BULK

P type Diode

### Device Parameters

<b>w</b>	(perimeter - 134.8) / 2.0 * 1e-6
----------	----------------------------------





## LVS Diva Compare rules

diva LVS Compare Code divaLVS.rul

```

; /*****
;
;
; Title:    Diva LVS rules for GENERIC PDK Process
;
;
; *****/
;

; *****/
;
; ***** LVS RULES *****
;
; *****/

lvsRules(

printf("\n GENERIC PDK LVS Rules\n")

; *****/
;
; *** MOSFETS ***
;
; *****/

; *****/
;
; combine parallel MOS devices (mosfets with different l's are not combined)
; *****/
;

procedure(parallelMOS(m1 m2)
prog((mt)
  mt = ncons(nil)
  when( and(m1->l m2->l)
    unless(abs(m1->l - m2->l) < .005e-6)
      return("doNotCombine")
    )
  )
  when( and(m1->w m2->w)
    unless(m1->m
      m1->m = 1
    )
    unless(m2->m
      m2->m = 1
    )
    mt->w = m1->w * m1->m + m2->w * m2->m
    mt->m = 1
    mt->l = m1->l
  )
  return(mt)
)
)

; *****/
;

```

## LVS Assura Compare rules

assura LVS Compare Code assuraLVS.rul

```

;
;
;
; *****
; GENERIC PDK Process Assura LVS Rules v2.0 1/10/02
;
;
; *****/
;

avCompareRules(

    fileVer = "2.3"
    fileDate = "Aug 2005"
    libType = "GENERIC PDK Process"
    libVer = "version 2.3"
    refDoc1 = "Library Specification No. GPDK, Version 2.3"

    printf( " \n" )
    printf(
        "***** \n"
    )
    printf( " \n" )
    printf(
        " GENERIC PDK Assura Compare Rules file version %s\n"
        fileVer
    )
    printf( " Cadence Design Systems, Inc. \n" )
    printf( " PDK Technology Center, Melbourne, FL \n" )
    printf( " %s \n" fileDate )
    printf( " Use with %s %s\n" libType libVer )
    printf( " \n" )
    printf( " Reference Documents: \n" )
    printf( " %s \n" refDoc1 )
    printf( " \n" )
    printf( " NOTICE: \n" )
    printf(
        " Cadence Design Systems shall not be liable for the accuracy \n"
    )
    printf(
        " of this LVS rule file or its ability to capture errors. \n"
    )
    printf(
        " The user is responsible for thoroughly testing and \n"
    )
    printf(
        " implementing its features. \n"
    )
    printf( " \n" )
    printf(
        "***** \n"
    )
)...

```

## Library CDF Definition

CDF gpdk180 libcdf

CDF device params

<b>basePasswd</b>	"base"	private	Flex PDK private password
<b>masterPasswd</b>	"master"	private	Flex PDK private password

switch !IC61

CDF parameters

<b>name</b>	"IxComponentType"
<b>prompt</b>	"Virtuoso XL Component Types"
<b>defValue</b>	"(((\"cells\" \"gpdk180 nmos3\" \"gpdk180 nmos\")) (\"type\" \"nmos\")) (\"IxActiveLayer\" \"Oxide drawing\") (\"IxMOSDeviceType\" \"NMOSH\") (\"IxDeviceWidth\" \"w\") (\"IxMaxWidth\" 1e-05)) ((\"cells\" \"gpdk180 pmos3\" \"gpdk180 pmos\")) (\"type\" \"pmos\")) (\"IxActiveLayer\" \"Oxide drawing\") (\"IxMOSDeviceType\" \"PMOS\") (\"IxDeviceWidth\" \"w\") (\"IxMaxWidth\" 1e-05)))"
<b>type</b>	"string"
<b>display</b>	"nil"
<b>dontSave</b>	"nil"

CDF properties

<b>formInitProc</b>	""
<b>doneProc</b>	""
<b>buttonFieldWidth</b>	340
<b>fieldHeight</b>	35
<b>fieldWidth</b>	350
<b>promptWidth</b>	175
<b>instNameType</b>	"schematic"
<b>instDisplayMode</b>	"instName"
<b>netNameType</b>	"schematic"
<b>termSimType</b>	"DC"
<b>termDisplayMode</b>	"netName"
<b>paramSimType</b>	"DC"
<b>paramEvaluate</b>	"t nil nil nil nil"
<b>paramDisplayMode</b>	"parameter"

## **Library MOS Definitions**

## Global MOS Parameters

Global Parameters

<b>mos_msDiff</b>	{2B}	Minimum diffusion spacing	
<b>mos_mxGate</b>	{5C}	Minimum gate extension	
<b>mos_msGate</b>	{5B}	Minimum gate spacing	
<b>mos_meDiffGate</b>	{5D}	Minimum diffusion enclosure of gate	
<b>mos_msGateCont</b>	{6E}	Minimum gate to contact spacing	
<b>mos_mwPoly</b>	{5A}	Minimum poly width on field	
<b>mos_msDiffPoly</b>	{5E}	Minimum diffusion to poly spacing	
<b>mos_mwCont</b>	{6A}	Minimum & maximum contact width	
<b>mos_msCont</b>	{6B}	Minimum contact spacing	
<b>mos_masCont</b>	{6B}	Minimum contact array spacing	
<b>mos_meDiffCont</b>	{6C}	Minimum diffusion enclosure of contact	
<b>mos_meeDiffCont</b>	{6C}	Minimum diffusion end enclosure of contact	
<b>mos_mePolyCont</b>	{6D}	Minimum poly enclosure of contact	
<b>mos_meePolyCont</b>	{6D}	Minimum poly end enclosure of contact	
<b>mos_mwM1</b>	{7A}	Minimum metal 1 width	
<b>mos_msM1</b>	{7B}	Minimum metal 1 spacing	
<b>mos_meM1Cont</b>	{7C}	Minimum metal 1 enclosure of contact	
<b>mos_meeM1Cont</b>	{7C}	Minimum metal 1 end enclosure of contact	

## Global MOS Parameters (cont.)

### Global Parameters

<b>nmos_meImplDiff</b>	{4C}	Minimum implant enclosure of diffusion	
<b>nmos_meImplGate</b>	0.12	Minimum implant enclosure of gates	
<b>nmos_meImplGate</b>	0.12	Minimum implant end enclosure of gates	
<b>nmos_meImplPoly</b>	0.12	Minimum implant enclosure of field poly	
<b>nmos_meImplCont</b>	0.0	Minimum implant enclosure of contact	
<b>nmos_meVoltDiff</b>	{2.5C}	Minimum thick oxide enclosure of thin oxide diffusion	
<b>nmos_meVoltImpl</b>	nil	Minimum thick oxide enclosure of implant	
<b>nmos_meWellDiff</b>	{2C}	Minimum wellbody enclosure of diffusion	
<b>nmos_meWellVolt</b>	0.0	Minimum wellbody enclosure of thick diffusion	
<b>nmos_meVthDiff</b>	0.0	Minimum high-vt implant enclosure of diffusion	
<b>nmos_meVthGate</b>	0.0	Minimum high-vt implant enclosure of gates	
<b>nmos_meeVthGate</b>	0.0	Minimum implant end enclosure of gates	
<b>nmos_meNtnDiff</b>	0.0	Minimum NTN enclosure of thin oxide diffusion	
<b>nmos_meNtnImpl</b>	0.0	Minimum NTN enclosure of implant	
<b>nmos_meWellNtn</b>	0.0	Minimum wellbody enclosure of NTN	

### Global Parameters

<b>pmos_meImplDiff</b>	{4C}	Minimum implant enclosure of diffusion	
<b>pmos_meImplGate</b>	0.12	Minimum implant enclosure of gates	
<b>pmos_meImplGate</b>	0.12	Minimum implant end enclosure of gates	
<b>pmos_meImplPoly</b>	0.12	Minimum implant enclosure of field poly	
<b>pmos_meImplCont</b>	0.0	Minimum implant enclosure of contact	
<b>pmos_meVoltDiff</b>	{2.5C}	Minimum thick oxide enclosure of thin oxide diffusion	
<b>pmos_meVoltImpl</b>	nil	Minimum thick oxide enclosure of implant	
<b>pmos_meWellDiff</b>	{2C}	Minimum wellbody enclosure of diffusion	
<b>pmos_meWellVolt</b>	0.0	Minimum wellbody enclosure of thick diffusion	
<b>pmos_meVthDiff</b>	0.0	Minimum high-vt implant enclosure of diffusion	
<b>pmos_meVthGate</b>	0.0	Minimum high-vt implant enclosure of gates	
<b>pmos_meeVthGate</b>	0.0	Minimum implant end enclosure of gates	

## Global MOS Parameters (cont.)

### Global Parameters

<b>tap_nmos_msTapDiff</b>	0.0	Minimum tap diffusion to device diffusion spacing	
<b>tap_nmos_msTapImpl</b>	{3C}	Minimum tap diffusion to device implant spacing	
<b>tap_nmos_meTimpTap</b>	{4C}	Minimum tap implant enclosure of tap diffusion	
<b>tap_nmos_msTimpDiff</b>	0.0	Minimum tap implant to device diffusion spacing	
<b>tap_nmos_msTimpImpl</b>	0.0	Minimum tap implant to device implant spacing	
<b>tap_nmos_maTap</b>	0.16	Minimum tap diffusion area	
<b>tap_nmos_maTimp</b>	0.36	Minimum tap implant area	
<b>tap_nmos_meWellTap</b>	0.0	Minimum well enclosure of tap diffusion	
<b>tap_nmos_msTapVolt</b>	0.25	Minimum tap diffusion to thick oxide spacing	
<b>tap_nmos_mwltap</b>	0.50	Minimum width of integrated tap	
<b>tap_nmos_msItapGate</b>	0.50	Minimum integrated tap spacing to gate	
<b>tap_nmos_maltimp</b>	0.36	Minimum integrated tap implant area	
<b>tap_nmos_meltimpltap</b>	{4C}	Minimum integrated tap implant enclosure of tap	
<b>tap_nmos_meltimpCont</b>	0.10	Minimum integrated tap implant enclosure of contact	
<b>tap_nmos_msTapNtn</b>	0.0	Minimum tap diffusion to NTN spacing	

### Global Parameters

<b>tap_pmos_msTapDiff</b>	0.0	Minimum tap diffusion to device diffusion spacing	
<b>tap_pmos_msTapImpl</b>	{3C}	Minimum tap diffusion to device implant spacing	
<b>tap_pmos_meTimpTap</b>	{4C}	Minimum tap implant enclosure of tap diffusion	
<b>tap_pmos_msTimpDiff</b>	0.0	Minimum tap implant to device diffusion spacing	
<b>tap_pmos_msTimpImpl</b>	0.0	Minimum tap implant to device implant spacing	
<b>tap_pmos_maTap</b>	0.16	Minimum tap diffusion area	
<b>tap_pmos_maTimp</b>	0.36	Minimum tap implant area	
<b>tap_pmos_meWellTap</b>	{2C}	Minimum well enclosure of tap diffusion	
<b>tap_pmos_msTapVolt</b>	0.25	Minimum tap diffusion to thick oxide spacing	
<b>tap_pmos_mwltap</b>	0.50	Minimum width of integrated tap	
<b>tap_pmos_msItapGate</b>	0.50	Minimum integrated tap spacing to gate	
<b>tap_pmos_maltimp</b>	0.36	Minimum integrated tap implant area	
<b>tap_pmos_meltimpltap</b>	{4C}	Minimum integrated tap implant enclosure of tap	
<b>tap_pmos_meltimpCont</b>	0.10	Minimum integrated tap implant enclosure of contact	

### Global Parameters

<b>mos_perimGateEdge</b>	t	Use four sides to calculate ps, pd	
<b>mos_aggregateFingers</b>	nil	Aggregate fingers: affects simM, as,ad,ps,pd	

## Callbacks

### MOS Callback Procedures

#### CDF Callback

```

////////////////////////////////////
;
; Perforce:      $File$
;                $Revision$
;                $DateTime$
;                $Change$
;                $Author$
;
; Description:    MOS Transistor CDF Callback
;
; Author:         Amir KoucheKinia <amir@cadence.com>
; Created:        Nov  6 16:41 02
; Modified:       Nov  6 16:41 02 14419 amir
; Language:       Skill
; Package:        Primitive_mos
; MainFun:        gpdK180_mosCB
; Status:         Experimental (Do Not Distribute)
;
; (C) Copyright 2002, Cadence Design Systems, all rights reserved.
;
////////////////////////////////////

procedure(gpdK180_mosCB(param "s")                                ;_May 15 03 amir 5359

let(
(
cdfId cellName libName procName paramId
dpt grid scale sGrid doList
applyThresh mode min max value
fingers w fw threshold
)

;; set local variable to global cdFgData
cdfId = cdFgData

caseq( concat(cdfId~>type)
( ( cellData baseCellData userCellData )
cellName = cdfId~>id~>name
libName = cdfId~>id~>lib~>name
)
( instData
cellName = cdfId~>id~>master~>cellName
libName = cdfId~>id~>master~>libName
)
( t...

```



## CDF Macro Definitions

### mosParamsA - MOS Parameters (Part A)

CDF macro mosParamsA

CDF parameters

<b>name</b>	"model"
<b>prompt</b>	"Model Name"
<b>defValue</b>	gpdK180_mosValue( 'model ?id cdfId ?returnString t )
<b>type</b>	"string"
<b>display</b>	"gpdK180_mosDisplay('model)"
<b>editable</b>	"nil"
<b>parseAsCEL</b>	"yes"

CDF parameters

<b>name</b>	"m"
<b>prompt</b>	"Multiplier"
<b>defValue</b>	"1"
<b>type</b>	"string"
<b>display</b>	"t"
<b>editable</b>	"t"
<b>callback</b>	"gpdK180_mosCB( 'm )"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

CDF parameters

<b>name</b>	"simM"
<b>prompt</b>	"Simulation Multiplier"
<b>defValue</b>	gpdK180_mosValue( 'simM ?id cdfId ?returnString t )
<b>type</b>	"string"
<b>display</b>	"nil"
<b>editable</b>	"nil"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF macro mosParamsA

## CDF parameters

<b>name</b>	"l"
<b>prompt</b>	"Length"
<b>units</b>	"lengthMetric"
<b>defValue</b>	gpdK180_mosValue( 'l' ?id cdfld ?returnString t )
<b>type</b>	"string"
<b>display</b>	"t"
<b>callback</b>	"gpdK180_mosCB( 'l' )"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"w"
<b>prompt</b>	"Total Width"
<b>units</b>	"lengthMetric"
<b>defValue</b>	gpdK180_mosValue( 'w' ?id cdfld ?returnString t )
<b>type</b>	"string"
<b>display</b>	"t"
<b>callback</b>	"gpdK180_mosCB( 'w' )"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"fw"
<b>prompt</b>	"Finger Width"
<b>units</b>	"lengthMetric"
<b>defValue</b>	gpdK180_mosValue( 'fw' ?id cdfld ?returnString t )
<b>type</b>	"string"
<b>display</b>	"t"
<b>editable</b>	"t"
<b>callback</b>	"gpdK180_mosCB( 'fw' )"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF macro mosParamsA

## CDF parameters

<b>name</b>	"simW"
<b>prompt</b>	"Simulation Width"
<b>units</b>	"lengthMetric"
<b>defValue</b>	gpdK180_mosValue( 'simW ?id cdfld ?returnString t )
<b>type</b>	"string"
<b>display</b>	"nil"
<b>editable</b>	"nil"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"fingers"
<b>prompt</b>	"Fingers"
<b>defValue</b>	gpdK180_mosValue( 'fingers ?id cdfld ?returnString t )
<b>type</b>	"string"
<b>display</b>	"t"
<b>editable</b>	"t"
<b>callback</b>	"gpdK180_mosCB( 'fingers )"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"threshold"
<b>prompt</b>	"Threshold"
<b>units</b>	"lengthMetric"
<b>defValue</b>	gpdK180_mosValue( 'threshold ?id cdfld ?returnString t )
<b>type</b>	"string"
<b>display</b>	"t"
<b>callback</b>	"gpdK180_mosCB( 'threshold )"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"applyThresh"
<b>prompt</b>	"Apply Threshold"
<b>defValue</b>	nil
<b>type</b>	"boolean"
<b>display</b>	"t"
<b>callback</b>	"gpdK180_mosCB( 'applyThresh )"

## CDF macro mosParamsA

## CDF parameters

<b>name</b>	"connectGates"
<b>prompt</b>	"Gate Connection"
<b>defValue</b>	"None"
<b>choices</b>	'("None" "Top" "Bottom" "Both" "Alternate")
<b>type</b>	"radio"
<b>display</b>	"t"

## CDF parameters

<b>name</b>	"connectSD"
<b>prompt</b>	"S/D Connection"
<b>defValue</b>	"None"
<b>choices</b>	'("None" "Source" "Drain" "Both")
<b>type</b>	"radio"
<b>display</b>	"gpd180_mosDisplay('connectSD)"

## CDF parameters

<b>name</b>	"sdMtlWidth"
<b>prompt</b>	"S/D Metal Width"
<b>units</b>	"lengthMetric"
<b>defValue</b>	gpd180_mosValue( 'sdMtlWidth ?id cdfld ?returnString t )
<b>type</b>	"string"
<b>display</b>	"t"
<b>editable</b>	"t"
<b>callback</b>	"gpd180_mosCB( 'sdMtlWidth)"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF macro mosParamsA

## CDF parameters

<b>name</b>	"mtlCvg"
<b>prompt</b>	"Metal Coverage"
<b>defValue</b>	""
<b>type</b>	"string"
<b>display</b>	"nil"
<b>callback</b>	"gpdK180_mosCB( 'mtlCvg )"

## CDF parameters

<b>name</b>	"switchSD"
<b>prompt</b>	"Switch S/D"
<b>defValue</b>	nil
<b>type</b>	"boolean"
<b>display</b>	"gpdK180_mosDisplay('switchSD)"
<b>callback</b>	"gpdK180_mosCB( 'switchSD )" "

## CDF parameters

<b>name</b>	"leftAbut"
<b>prompt</b>	"Left Abutment"
<b>defValue</b>	0
<b>type</b>	"int"
<b>display</b>	"nil"
<b>editable</b>	"t"
<b>callback</b>	"gpdK180_mosCB( 'leftAbut)"

## CDF parameters

<b>name</b>	"rightAbut"
<b>prompt</b>	"Right Abutment"
<b>defValue</b>	0
<b>type</b>	"int"
<b>display</b>	"nil"
<b>editable</b>	"t"
<b>callback</b>	"gpdK180_mosCB( 'rightAbut)"

## mosParamsTap - MOS Tap Parameters

CDF macro mosParamsTap

CDF parameters

<b>name</b>	"tap"
<b>prompt</b>	"Bodytie Type"
<b>defValue</b>	"None"
<b>choices</b>	'( "None" "Integrated" "Detached" )
<b>type</b>	"cyclic"
<b>display</b>	"t"
<b>callback</b>	"gpdtk180_mosCB( 'tap' )"

CDF parameters

<b>name</b>	"leftTap"
<b>prompt</b>	"Left Tap"
<b>defValue</b>	t
<b>type</b>	"boolean"
<b>display</b>	"gpdtk180_mosDisplay( 'leftTap ')"

CDF parameters

<b>name</b>	"rightTap"
<b>prompt</b>	"Right Tap"
<b>defValue</b>	nil
<b>type</b>	"boolean"
<b>display</b>	"gpdtk180_mosDisplay( 'rightTap ')"

CDF parameters

<b>name</b>	"bottomTap"
<b>prompt</b>	"Bottom Tap"
<b>defValue</b>	nil
<b>type</b>	"boolean"
<b>display</b>	"gpdtk180_mosDisplay( 'bottomTap ')"

CDF parameters

<b>name</b>	"topTap"
<b>prompt</b>	"Top Tap"
<b>defValue</b>	nil
<b>type</b>	"boolean"
<b>display</b>	"gpdtk180_mosDisplay( 'topTap ')"

CDF parameters

<b>name</b>	"tapExtension"
<b>prompt</b>	"Tap Extension"
<b>defValue</b>	""
<b>type</b>	"string"
<b>display</b>	"gpdtk180_mosDisplay( 'tapExtension ')"
<b>callback</b>	"gpdtk180_mosCB( 'tapExtension ')"

CDF parameters

<b>name</b>	"tapCntRows"
<b>prompt</b>	"Tap Contact Rows"
<b>defValue</b>	1
<b>type</b>	"int"
<b>display</b>	"gpdtk180_mosDisplay( 'tapCntRows ')"

## mosParamsB - MOS Parameters (Part B)

CDF macro mosParamsB

CDF parameters

<b>name</b>	"showSimParams"
<b>prompt</b>	"Show Sim Parameters"
<b>defValue</b>	nil
<b>type</b>	"boolean"
<b>display</b>	"gpd180_mosDisplay('showSimParams)"

CDF parameters

<b>name</b>	"editAreaPerim"
<b>prompt</b>	"Edit Area & Perim"
<b>defValue</b>	nil
<b>type</b>	"boolean"
<b>display</b>	"gpd180_mosDisplay('editAreaPerim)"
<b>callback</b>	"gpd180_mosCB( 'editAreaPerim )" "

CDF parameters

<b>name</b>	"ad"
<b>prompt</b>	"Drain diffusion area"
<b>defValue</b>	gpd180_mosValue( 'ad ?id cdfId ?returnString t )
<b>type</b>	"string"
<b>display</b>	"gpd180_mosDisplay('ad)"
<b>callback</b>	"gpd180_mosCB( 'ad )" "
<b>editable</b>	"cdfgData->editAreaPerim->value"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

CDF parameters

<b>name</b>	"as"
<b>prompt</b>	"Source diffusion area"
<b>defValue</b>	gpd180_mosValue( 'as ?id cdfId ?returnString t )
<b>type</b>	"string"
<b>display</b>	"gpd180_mosDisplay('as)"
<b>callback</b>	"gpd180_mosCB( 'as )" "
<b>editable</b>	"cdfgData->editAreaPerim->value"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF macro mosParamsB

## CDF parameters

<b>name</b>	"pd"
<b>prompt</b>	"Drain diffusion periphery"
<b>units</b>	"lengthMetric"
<b>defValue</b>	gpd180_mosValue( 'pd ?id cdfld ?returnString t )
<b>type</b>	"string"
<b>display</b>	"gpd180_mosDisplay('pd)"
<b>callback</b>	"gpd180_mosCB( 'pd )" "
<b>editable</b>	"cdfgData->editAreaPerim->value"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"ps"
<b>prompt</b>	"Source diffusion periphery"
<b>units</b>	"lengthMetric"
<b>defValue</b>	gpd180_mosValue( 'ps ?id cdfld ?returnString t )
<b>type</b>	"string"
<b>display</b>	"gpd180_mosDisplay('ps)"
<b>callback</b>	"gpd180_mosCB( 'ps )" "
<b>editable</b>	"cdfgData->editAreaPerim->value"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"nrd"
<b>prompt</b>	"Drain diffusion res squares"
<b>defValue</b>	""
<b>type</b>	"string"
<b>display</b>	"gpd180_mosDisplay( 'nrd )" "
<b>editable</b>	"t"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"nrs"
<b>prompt</b>	"Source diffusion res squares"
<b>defValue</b>	""
<b>type</b>	"string"
<b>display</b>	"gpd180_mosDisplay( 'nrs )" "
<b>editable</b>	"t"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"



## CDF macro mosParamsB

## CDF parameters

<b>name</b>	"ld"
<b>prompt</b>	"Drain diffusion length"
<b>units</b>	"lengthMetric"
<b>defValue</b>	""
<b>type</b>	"string"
<b>display</b>	"gpd180_mosDisplay( 'ld )"
<b>editable</b>	"t"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"ls"
<b>prompt</b>	"Source diffusion length"
<b>units</b>	"lengthMetric"
<b>defValue</b>	""
<b>type</b>	"string"
<b>display</b>	"gpd180_mosDisplay( 'ls )"
<b>editable</b>	"t"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"off"
<b>prompt</b>	"Device initially off"
<b>type</b>	"boolean"
<b>display</b>	"gpd180_mosDisplay( 'off )"

## CDF macro mosParamsB

## CDF parameters

<b>name</b>	"Vds"
<b>prompt</b>	"Drain source initial voltage"
<b>units</b>	"voltage"
<b>defValue</b>	""
<b>type</b>	"string"
<b>display</b>	"gpd180_mosDisplay( 'Vds )"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"Vgs"
<b>prompt</b>	"Gate source initial voltage"
<b>units</b>	"voltage"
<b>defValue</b>	""
<b>type</b>	"string"
<b>display</b>	"gpd180_mosDisplay( 'Vgs )"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"Vbs"
<b>prompt</b>	"Bulk source initial voltage"
<b>units</b>	"voltage"
<b>defValue</b>	""
<b>type</b>	"string"
<b>display</b>	"gpd180_mosDisplay( 'Vbs )"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"trise"
<b>prompt</b>	"Temp rise from ambient"
<b>defValue</b>	""
<b>type</b>	"string"
<b>display</b>	"gpd180_mosDisplay( 'trise )"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF macro mosParamsB

## CDF parameters

<b>name</b>	"rdc"
<b>prompt</b>	"Additional drain resistance"
<b>units</b>	"current"
<b>defValue</b>	""
<b>type</b>	"string"
<b>display</b>	"gpd180_mosDisplay( 'rdc )"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"rsc"
<b>prompt</b>	"Additional source resistance"
<b>units</b>	"current"
<b>defValue</b>	""
<b>type</b>	"string"
<b>display</b>	"gpd180_mosDisplay( 'rsc )"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"dtemp"
<b>prompt</b>	"Temperature difference"
<b>defValue</b>	""
<b>type</b>	"string"
<b>display</b>	"gpd180_mosDisplay( 'dtemp )"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"geo"
<b>prompt</b>	"Source/drain selector"
<b>defValue</b>	""
<b>type</b>	"string"
<b>display</b>	"gpd180_mosDisplay( 'geo )"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF macro mosParamsB

## CDF parameters

<b>name</b>	"pasUpdateParamList"
<b>prompt</b>	"PAS Update Param List"
<b>defValue</b>	"I"
<b>type</b>	"string"
<b>display</b>	"nil"
<b>editable</b>	"nil"

## switch !IC61

## CDF parameters

<b>name</b>	"IxIgnoredParamsForCAS"
<b>prompt</b>	"IxIgnoredParamsForCAS"
<b>defValue</b>	"connectGates connectSD mtICvg leftAbut rightAbut showSimParams switchSD ad as pd ps"
<b>type</b>	"string"
<b>display</b>	"nil"
<b>editable</b>	"nil"

## CDF properties

<b>formInitProc</b>	"PasCdfFormInit"
<b>doneProc</b>	"PasCdfDone"
<b>buttonFieldWidth</b>	340
<b>fieldHeight</b>	35
<b>fieldWidth</b>	500
<b>promptWidth</b>	175
<b>instNameType</b>	"schematic"
<b>instDisplayMode</b>	"instName"
<b>netNameType</b>	"schematic"
<b>termSimType</b>	"DC"
<b>termDisplayMode</b>	"voltage"
<b>paramSimType</b>	"DC"
<b>paramEvaluate</b>	"nil nil nil t nil"
<b>paramDisplayMode</b>	"parameter"
<b>modelLabelSet</b>	"vto kp gamma"
<b>opPointLabelSet</b>	"id vgs vds gm"
<b>paramLabelSet</b>	"-model w l m"

## mosSimInfo - MOS Simulation Information (common)

CDF macro mosSimInfo

CDF siminfo

<b>simulator</b>	ads
<b>netlistProcedure</b>	ADSSimCompPrim
<b>otherParameters</b>	(model)
<b>instParameters</b>	(Width Length As Ad Ps Pd Nrd Nrs Id Is _M Trise)
<b>componentName</b>	nil
<b>termMapping</b>	(nil D ":P1" G ":P2" S ":P3" B ":P4")
<b>propMapping</b>	(nil Width simW Length l As as Ad ad Ps ps Pd pd Nrd nrd Nrs nrs _M simM Trise trise Region region)
<b>typeMapping</b>	nil
<b>uselib</b>	nil

CDF siminfo

<b>simulator</b>	ams
<b>propMapping</b>	( nil m simM w simW )
<b>namePrefix</b>	""
<b>isPrimitive</b>	t
<b>instParameters</b>	(w l as ad ps pd nrd nrs Id ls m trise model)
<b>otherParameters</b>	(model)

CDF siminfo

<b>simulator</b>	auCdl
<b>netlistProcedure</b>	ansCdlCompPrim
<b>instParameters</b>	(M L W)
<b>propMapping</b>	(nil M m L I W w)
<b>namePrefix</b>	"M"

CDF siminfo

<b>simulator</b>	auLvs
<b>propMapping</b>	nil
<b>netlistProcedure</b>	ansLvsCompPrim
<b>instParameters</b>	(m l w)
<b>permuteRule</b>	"(p D S)"
<b>namePrefix</b>	"M"

## CDF macro mosSimInfo

CDF siminfo

<b>simulator</b>	spectre
<b>propMapping</b>	( nil m simM w simW )
<b>namePrefix</b>	""
<b>otherParameters</b>	(model)
<b>instParameters</b>	(w l as ad ps pd nrd nrs ld ls m trise)
<b>termMapping</b>	(nil D \:d G \:g S \:s B \:b)

## mosSimInfoRF

CDF macro mosSimInfoRF

CDF siminfo

<b>simulator</b>	ads
<b>netlistProcedure</b>	ADSSimCompPrim
<b>otherParameters</b>	(model)
<b>instParameters</b>	(Width Length fingers simM)
<b>componentName</b>	nil
<b>termMapping</b>	(nil D ":P1" G ":P2" S ":P3" B ":P4")
<b>propMapping</b>	(nil Width simW Length l)
<b>typeMapping</b>	nil
<b>uselib</b>	nil

CDF siminfo

<b>simulator</b>	ams
<b>propMapping</b>	( nil w simW )
<b>namePrefix</b>	""
<b>isPrimitive</b>	t
<b>instParameters</b>	(w l fingers simM)
<b>otherParameters</b>	(model)

CDF siminfo

<b>simulator</b>	auCdl
<b>netlistProcedure</b>	ansCdlCompPrim
<b>instParameters</b>	(M L W)
<b>propMapping</b>	(nil M m L I W w)
<b>namePrefix</b>	"M"

CDF siminfo

<b>simulator</b>	auLvs
<b>propMapping</b>	nil
<b>netlistProcedure</b>	ansLvsCompPrim
<b>instParameters</b>	(m l w)
<b>permuteRule</b>	"(p D S)"
<b>namePrefix</b>	"M"

## CDF macro mosSimInfoRF

CDF siminfo

<b>simulator</b>	spectre
<b>propMapping</b>	( nil w simW )
<b>namePrefix</b>	""
<b>otherParameters</b>	(model)
<b>instParameters</b>	(w l fingers simM)
<b>termMapping</b>	(nil D \:d G \:g S \:s B \:b)



# mos3n - 3 terminal NMOS Transistor CDF

CDF macro mos3n

include macro mosParamsA

include macro mosParamsB

include macro mosSimInfo

CDF siminfo

<b>simulator</b>	ads
<b>termOrder</b>	(D G S bulkn)

CDF siminfo

<b>simulator</b>	ams
<b>termOrder</b>	(D G S bulkn)
<b>extraTerminals</b>	((nil name "bulkn" direction "inputOutput" netExpr "[@bulkn:%:gnd!]" ))

CDF siminfo

<b>simulator</b>	spectre
<b>termOrder</b>	(D G S bulkn)

CDF siminfo

<b>simulator</b>	auCdl
<b>termOrder</b>	(D G S bulkn)

CDF siminfo

<b>simulator</b>	auLvs
<b>termOrder</b>	(D G S bulkn)
<b>deviceTerminals</b>	"D G S B"

## mos3p - 3 terminal PMOS Transistor CDF

CDF macro mos3p

```
include macro mosParamsA
```

```
include macro mosParamsB
```

```
include macro mosSimInfo
```

CDF siminfo

<b>simulator</b>	ads
<b>termOrder</b>	(D G S bulkp)

CDF siminfo

<b>simulator</b>	ams
<b>termOrder</b>	(D G S bulkp)
<b>extraTerminals</b>	((nil name "bulkp" direction "inputOutput" netExpr "[@bulkp:%:vdd!]" ))

CDF siminfo

<b>simulator</b>	spectre
<b>termOrder</b>	(D G S bulkp)

CDF siminfo

<b>simulator</b>	auCdl
<b>termOrder</b>	(D G S bulkp)

CDF siminfo

<b>simulator</b>	auLvs
<b>termOrder</b>	(D G S bulkp)
<b>deviceTerminals</b>	"D G S B"

## mos4 - 4 terminal MOS Transistor CDF

CDF macro mos4

include macro mosParamsA

include macro mosParamsTap

include macro mosParamsB

include macro mosSimInfo

CDF siminfo

<b>simulator</b>	ads
<b>termOrder</b>	(D G S B)

CDF siminfo

<b>simulator</b>	ams
<b>termOrder</b>	(D G S B)

CDF siminfo

<b>simulator</b>	spectre
<b>termOrder</b>	(D G S B)

CDF siminfo

<b>simulator</b>	auCdl
<b>termOrder</b>	(D G S B)

CDF siminfo

<b>simulator</b>	auLvs
<b>termOrder</b>	(D G S B)

mos4RF - 4 terminal RF MOS Transistor CDF

CDF macro mos4RF

```
include macro mosParamsA
include macro mosParamsTap
include macro mosParamsB
include macro mosSimInfoRF
```

CDF siminfo

<b>simulator</b>	ads
<b>termOrder</b>	(D G S B)

CDF siminfo

<b>simulator</b>	ams
<b>termOrder</b>	(D G S B)

CDF siminfo

<b>simulator</b>	spectre
<b>termOrder</b>	(D G S B)

CDF siminfo

<b>simulator</b>	auCdl
<b>termOrder</b>	(D G S B)

CDF siminfo

<b>simulator</b>	auLvs
<b>termOrder</b>	(D G S B)

**CDF Definitions****MOS Transistor****nmos3**

CDF gpd180 nmos3

CDF device params

<b>grid</b>	\$grid	private	MFG Grid
<b>scale</b>	1.0e-6	private	Dimension scale factor
<b>model</b>	"nmos1"	private	Device Model Name
<b>category</b>	"mos"	private	Library Manager Category
<b>perimGateEdge</b>	\$mos_perimGateEdge	private	Use four sides to calculate ps, pd
<b>aggregateFingers</b>	\$mos_aggregateFingers	private	Aggregate fingers: affects simM, as,ad,ps,pd

CDF device params

<b>dwGate</b>	2.0	private	Default channel width
<b>mwGate</b>	{2A}	private	Minimum channel Width
<b>xwGate</b>	50.00	private	Maximum channel Width
<b>mlGate</b>	{5A}	private	Minimum channel Length
<b>msDiff</b>	\$mos_msDiff	private	Minimum diffusion spacing
<b>mxGate</b>	\$mos_mxGate	private	Minimum gate extension
<b>msGate</b>	\$mos_msGate	private	Minimum gate spacing
<b>meDiffGate</b>	\$mos_meDiffGate	private	Minimum diffusion enclosure of gate
<b>msGateCont</b>	\$mos_msGateCont	private	Minimum gate to contact spacing
<b>mwPoly</b>	\$mos_mwPoly	private	Minimum poly width on field
<b>msDiffPoly</b>	\$mos_msDiffPoly	private	Minimum diffusion to poly spacing
<b>mwCont</b>	\$mos_mwCont	private	Minimum contact width
<b>msCont</b>	\$mos_msCont	private	Minimum contact spacing
<b>masCont</b>	\$mos_masCont	private	Minimum contact array spacing
<b>meDiffCont</b>	\$mos_meDiffCont	private	Minimum diffusion enclosure of contact
<b>meeDiffCont</b>	\$mos_meeDiffCont	private	Minimum diffusion end enclosure of contact
<b>mePolyCont</b>	\$mos_mePolyCont	private	Minimum poly enclosure of contact
<b>meePolyCont</b>	\$mos_meePolyCont	private	Minimum poly end enclosure of contact
<b>mwM1</b>	\$mos_mwM1	private	Minimum metal 1 width
<b>msM1</b>	\$mos_msM1	private	Minimum metal 1 spacing
<b>meM1Cont</b>	\$mos_meM1Cont	private	Minimum metal 1 enclosure of contact
<b>meeM1Cont</b>	\$mos_meeM1Cont	private	Minimum metal 1 end enclosure of contact

CDF device params

<b>meImplDiff</b>	\$nmos_meImplDiff	private	Minimum implant enclosure of diffusion
<b>meImplGate</b>	\$nmos_meImplGate	private	Minimum implant enclosure of gate
<b>meImplGate</b>	\$nmos_meImplGate	private	Minimum implant end enclosure of gate
<b>meImplPoly</b>	\$nmos_meImplPoly	private	Minimum implant enclosure of poly on field
<b>meImplCont</b>	\$nmos_meImplCont	private	Minimum implant enclosure of contact
<b>meWellDiff</b>	\$nmos_meWellDiff	private	Minimum well enclosure of diffusion

## CDF gpdk180 nmos3

include macro mos3n

CDF siminfo

<b>simulator</b>	auLvs
<b>componentName</b>	"nmos"

CDF siminfo

<b>simulator</b>	auCdl
<b>componentName</b>	nil
<b>modelName</b>	"nmos"

CDF cellview

<b>symbol</b>	pas_std	nmos3	symbol
<b>auCdl</b>	pas_std	nmos3	spectre
<b>auLvs</b>	pas_std	nmos3	spectre
<b>spectre</b>	pas_std	nmos3	spectre
<b>ads</b>	pas_std	nmos3	spectre

## nmos

CDF gpd180 nmos

CDF device params

<b>grid</b>	\$grid	private	MFG Grid
<b>scale</b>	1.0e-6	private	Dimension scale factor
<b>model</b>	"nmos1"	private	Device Model Name
<b>category</b>	"mos"	private	Library Manager Category
<b>perimGateEdge</b>	\$mos_perimGateEdge	private	Use four sides to calculate ps, pd
<b>aggregateFingers</b>	\$mos_aggregateFingers	private	Aggregate fingers: affects simM, as,ad,ps,pd

CDF device params

<b>dwGate</b>	2.0	private	Default channel width
<b>mwGate</b>	{2A}	private	Minimum channel Width
<b>xwGate</b>	50.00	private	Maximum channel Width
<b>mlGate</b>	{5A}	private	Minimum channel Length
<b>msDiff</b>	\$mos_msDiff	private	Minimum diffusion spacing
<b>mxGate</b>	\$mos_mxGate	private	Minimum gate extension
<b>msGate</b>	\$mos_msGate	private	Minimum gate spacing
<b>meDiffGate</b>	\$mos_meDiffGate	private	Minimum diffusion enclosure of gate
<b>msGateCont</b>	\$mos_msGateCont	private	Minimum gate to contact spacing
<b>mwPoly</b>	\$mos_mwPoly	private	Minimum poly width on field
<b>msDiffPoly</b>	\$mos_msDiffPoly	private	Minimum diffusion to poly spacing
<b>mwCont</b>	\$mos_mwCont	private	Minimum contact width
<b>msCont</b>	\$mos_msCont	private	Minimum contact spacing
<b>masCont</b>	\$mos_masCont	private	Minimum contact array spacing
<b>meDiffCont</b>	\$mos_meDiffCont	private	Minimum diffusion enclosure of contact
<b>meeDiffCont</b>	\$mos_meeDiffCont	private	Minimum diffusion end enclosure of contact
<b>mePolyCont</b>	\$mos_mePolyCont	private	Minimum poly enclosure of contact
<b>meePolyCont</b>	\$mos_meePolyCont	private	Minimum poly end enclosure of contact
<b>mwM1</b>	\$mos_mwM1	private	Minimum metal 1 width
<b>msM1</b>	\$mos_msM1	private	Minimum metal 1 spacing
<b>meM1Cont</b>	\$mos_meM1Cont	private	Minimum metal 1 enclosure of contact
<b>meeM1Cont</b>	\$mos_meeM1Cont	private	Minimum metal 1 end enclosure of contact

CDF device params

<b>meImplDiff</b>	\$nmos_meImplDiff	private	Minimum implant enclosure of diffusion
<b>meImplGate</b>	\$nmos_meImplGate	private	Minimum implant enclosure of gate
<b>meImplGate</b>	\$nmos_meImplGate	private	Minimum implant end enclosure of gate
<b>meImplPoly</b>	\$nmos_meImplPoly	private	Minimum implant enclosure of poly on field
<b>meImplCont</b>	\$nmos_meImplCont	private	Minimum implant enclosure of contact
<b>meWellDiff</b>	\$nmos_meWellDiff	private	Minimum well enclosure of diffusion

# CDF gpdk180 nmos

include macro mos4

CDF siminfo

<b>simulator</b>	auLvs
<b>componentName</b>	"\$cell"

CDF siminfo

<b>simulator</b>	auCdl
<b>componentName</b>	nil
<b>modelName</b>	"nmos"

CDF cellview

<b>symbol</b>	pas_std	nmos4	symbol	
<b>auCdl</b>	pas_std	nmos4	symbol	
<b>auLvs</b>	pas_std	nmos4	symbol	
<b>spectre</b>	pas_std	nmos4	symbol	
<b>ads</b>	pas_std	nmos4	symbol	
<b>ivpcell</b>	\$lib	\$cell	symbol	5

CDF device params

<b>msTapDiff</b>	\$tap_nmos_msTapDiff	private	Minimum tap diffusion to device diffusion spacing
<b>msTapImpl</b>	\$tap_nmos_msTapImpl	private	Minimum tap diffusion to device implant spacing
<b>meTimpTap</b>	\$tap_nmos_meTimpTap	private	Minimum tap implant enclosure of tap diffusion
<b>msTimpDiff</b>	\$tap_nmos_msTimpDiff	private	Minimum tap implant to device diffusion spacing
<b>msTimpImpl</b>	\$tap_nmos_msTimpImpl	private	Minimum tap implant to device implant spacing
<b>maTap</b>	\$tap_nmos_maTap	private	Minimum tap diffusion area
<b>maTimp</b>	\$tap_nmos_maTimp	private	Minimum tap implant area
<b>meWellTap</b>	\$tap_nmos_meWellTap	private	Minimum well enclosure of tap diffusion
<b>mwltap</b>	\$tap_nmos_mwltap	private	Minimum width of integrated tap
<b>msltapGate</b>	\$tap_nmos_msltapGate	private	Minimum integrated tap spacing to gate
<b>maltemp</b>	\$tap_nmos_maltemp	private	Minimum integrated tap implant area
<b>meltempTap</b>	\$tap_nmos_meltempTap	private	Minimum integrated tap implant enclosure of tap
<b>meltempCont</b>	\$tap_nmos_meltempCont	private	Minimum integrated tap enclosure of contact



## nmos3hv

CDF gpd180 nmos3hv

CDF device params

<b>grid</b>	\$grid	private	MFG Grid
<b>scale</b>	1.0e-6	private	Dimension scale factor
<b>model</b>	"nmos25"	private	Device Model Name
<b>category</b>	"mos"	private	Library Manager Category
<b>perimGateEdge</b>	\$mos_perimGateEdge	private	Use four sides to calculate ps, pd
<b>aggregateFingers</b>	\$mos_aggregateFingers	private	Aggregate fingers: affects simM, as,ad,ps,pd

CDF device params

<b>dwGate</b>	2.0	private	Default channel width
<b>mwGate</b>	{2A}	private	Minimum channel Width
<b>xwGate</b>	50.00	private	Maximum channel Width
<b>mlGate</b>	{5A}	private	Minimum channel Length
<b>msDiff</b>	\$mos_msDiff	private	Minimum diffusion spacing
<b>mxGate</b>	\$mos_mxGate	private	Minimum gate extension
<b>msGate</b>	\$mos_msGate	private	Minimum gate spacing
<b>meDiffGate</b>	\$mos_meDiffGate	private	Minimum diffusion enclosure of gate
<b>msGateCont</b>	\$mos_msGateCont	private	Minimum gate to contact spacing
<b>mwPoly</b>	\$mos_mwPoly	private	Minimum poly width on field
<b>msDiffPoly</b>	\$mos_msDiffPoly	private	Minimum diffusion to poly spacing
<b>mwCont</b>	\$mos_mwCont	private	Minimum contact width
<b>msCont</b>	\$mos_msCont	private	Minimum contact spacing
<b>masCont</b>	\$mos_masCont	private	Minimum contact array spacing
<b>meDiffCont</b>	\$mos_meDiffCont	private	Minimum diffusion enclosure of contact
<b>meeDiffCont</b>	\$mos_meeDiffCont	private	Minimum diffusion end enclosure of contact
<b>mePolyCont</b>	\$mos_mePolyCont	private	Minimum poly enclosure of contact
<b>meePolyCont</b>	\$mos_meePolyCont	private	Minimum poly end enclosure of contact
<b>mwM1</b>	\$mos_mwM1	private	Minimum metal 1 width
<b>msM1</b>	\$mos_msM1	private	Minimum metal 1 spacing
<b>meM1Cont</b>	\$mos_meM1Cont	private	Minimum metal 1 enclosure of contact
<b>meeM1Cont</b>	\$mos_meeM1Cont	private	Minimum metal 1 end enclosure of contact

CDF device params

<b>meImplDiff</b>	\$nmos_meImplDiff	private	Minimum implant enclosure of diffusion
<b>meImplGate</b>	\$nmos_meImplGate	private	Minimum implant enclosure of gate
<b>meImplGate</b>	\$nmos_meImplGate	private	Minimum implant end enclosure of gate
<b>meImplPoly</b>	\$nmos_meImplPoly	private	Minimum implant enclosure of poly on field
<b>meImplCont</b>	\$nmos_meImplCont	private	Minimum implant enclosure of contact
<b>meWellDiff</b>	\$nmos_meWellDiff	private	Minimum well enclosure of diffusion
<b>meVoltDiff</b>	\$nmos_meVoltDiff	private	Minimum thick oxide enclosure of thin oxide diffusion
<b>meVoltImpl</b>	\$nmos_meVoltImpl	private	Minimum thick oxide enclosure of implant

CDF gpdk180 nmos3hv

include macro mos3n

CDF siminfo

<b>simulator</b>	auLvs
<b>componentName</b>	"nmoshv"

CDF siminfo

<b>simulator</b>	auCdl
<b>componentName</b>	nil
<b>modelName</b>	"nmoshv"

CDF cellview

<b>symbol</b>	pas_std	nmos3_hv	symbol
<b>auCdl</b>	pas_std	nmos3_hv	spectre
<b>auLvs</b>	pas_std	nmos3_hv	spectre
<b>spectre</b>	pas_std	nmos3_hv	spectre
<b>ads</b>	pas_std	nmos3_hv	spectre

## nmoshv

CDF gpdk180 nmoshv

CDF device params

<b>grid</b>	\$grid	private	MFG Grid
<b>scale</b>	1.0e-6	private	Dimension scale factor
<b>model</b>	"nmos25"	private	Device Model Name
<b>category</b>	"mos"	private	Library Manager Category
<b>perimGateEdge</b>	\$mos_perimGateEdge	private	Use four sides to calculate ps, pd
<b>aggregateFingers</b>	\$mos_aggregateFingers	private	Aggregate fingers: affects simM, as,ad,ps,pd

CDF device params

<b>dwGate</b>	2.0	private	Default channel width
<b>mwGate</b>	{2A}	private	Minimum channel Width
<b>xwGate</b>	50.00	private	Maximum channel Width
<b>mlGate</b>	{5A}	private	Minimum channel Length
<b>msDiff</b>	\$mos_msDiff	private	Minimum diffusion spacing
<b>mxGate</b>	\$mos_mxGate	private	Minimum gate extension
<b>msGate</b>	\$mos_msGate	private	Minimum gate spacing
<b>meDiffGate</b>	\$mos_meDiffGate	private	Minimum diffusion enclosure of gate
<b>msGateCont</b>	\$mos_msGateCont	private	Minimum gate to contact spacing
<b>mwPoly</b>	\$mos_mwPoly	private	Minimum poly width on field
<b>msDiffPoly</b>	\$mos_msDiffPoly	private	Minimum diffusion to poly spacing
<b>mwCont</b>	\$mos_mwCont	private	Minimum contact width
<b>msCont</b>	\$mos_msCont	private	Minimum contact spacing
<b>masCont</b>	\$mos_masCont	private	Minimum contact array spacing
<b>meDiffCont</b>	\$mos_meDiffCont	private	Minimum diffusion enclosure of contact
<b>meeDiffCont</b>	\$mos_meeDiffCont	private	Minimum diffusion end enclosure of contact
<b>mePolyCont</b>	\$mos_mePolyCont	private	Minimum poly enclosure of contact
<b>meePolyCont</b>	\$mos_meePolyCont	private	Minimum poly end enclosure of contact
<b>mwM1</b>	\$mos_mwM1	private	Minimum metal 1 width
<b>msM1</b>	\$mos_msM1	private	Minimum metal 1 spacing
<b>meM1Cont</b>	\$mos_meM1Cont	private	Minimum metal 1 enclosure of contact
<b>meeM1Cont</b>	\$mos_meeM1Cont	private	Minimum metal 1 end enclosure of contact

CDF device params

<b>meImplDiff</b>	\$nmos_meImplDiff	private	Minimum implant enclosure of diffusion
<b>meImplGate</b>	\$nmos_meImplGate	private	Minimum implant enclosure of gate
<b>meImplGate</b>	\$nmos_meImplGate	private	Minimum implant end enclosure of gate
<b>meImplPoly</b>	\$nmos_meImplPoly	private	Minimum implant enclosure of poly on field
<b>meImplCont</b>	\$nmos_meImplCont	private	Minimum implant enclosure of contact
<b>meWellDiff</b>	\$nmos_meWellDiff	private	Minimum well enclosure of diffusion
<b>meVoltDiff</b>	\$nmos_meVoltDiff	private	Minimum thick oxide enclosure of thin oxide diffusion
<b>meVoltImpl</b>	\$nmos_meVoltImpl	private	Minimum thick oxide enclosure of implant

## CDF gpd180 nmoshv

include macro mos4

CDF siminfo

<b>simulator</b>	auLvs
<b>componentName</b>	"\$cell"

CDF siminfo

<b>simulator</b>	auCdl
<b>componentName</b>	nil
<b>modelName</b>	"\$cell"

CDF cellview

<b>symbol</b>	pas_std	nmos4_hv	symbol	
<b>auCdl</b>	pas_std	nmos4_hv	symbol	
<b>auLvs</b>	pas_std	nmos4_hv	symbol	
<b>spectre</b>	pas_std	nmos4_hv	symbol	
<b>ads</b>	pas_std	nmos4_hv	symbol	
<b>ivpcell</b>	\$lib	\$cell	symbol	5

CDF device params

<b>msTapDiff</b>	\$tap_nmos_msTapDiff	private	Minimum tap diffusion to device diffusion spacing
<b>msTapImpl</b>	\$tap_nmos_msTapImpl	private	Minimum tap diffusion to device implant spacing
<b>meTimpTap</b>	\$tap_nmos_meTimpTap	private	Minimum tap implant enclosure of tap diffusion
<b>msTimpDiff</b>	\$tap_nmos_msTimpDiff	private	Minimum tap implant to device diffusion spacing
<b>msTimpImpl</b>	\$tap_nmos_msTimpImpl	private	Minimum tap implant to device implant spacing
<b>maTap</b>	\$tap_nmos_maTap	private	Minimum tap diffusion area
<b>maTimp</b>	\$tap_nmos_maTimp	private	Minimum tap implant area
<b>meWellTap</b>	\$tap_nmos_meWellTap	private	Minimum well enclosure of tap diffusion
<b>mwltap</b>	\$tap_nmos_mwltap	private	Minimum width of integrated tap
<b>msltapGate</b>	\$tap_nmos_msltapGate	private	Minimum integrated tap spacing to gate
<b>maltemp</b>	\$tap_nmos_maltemp	private	Minimum integrated tap implant area
<b>meltempTap</b>	\$tap_nmos_meltempTap	private	Minimum integrated tap implant enclosure of tap
<b>meltempCont</b>	\$tap_nmos_meltempCont	private	Minimum integrated tap enclosure of contact
<b>msTapVolt</b>	\$tap_nmos_msTapVolt	public	Minimum tap diffusion to thick oxide spacing

# nmosrf

CDF gpdk180 nmosrf

CDF device params

<b>grid</b>	\$grid	private	MFG Grid
<b>scale</b>	1.0e-6	private	Dimension scale factor
<b>model</b>	"nmos_rf"	private	Device Model Name
<b>category</b>	"mos"	private	Library Manager Category
<b>perimGateEdge</b>	\$mos_perimGateEdge	private	Use four sides to calculate ps, pd
<b>aggregateFingers</b>	\$mos_aggregateFingers	private	Aggregate fingers: affects simM, as,ad,ps,pd

CDF device params

<b>dwGate</b>	2.0	private	Default channel width
<b>mwGate</b>	1.5	private	Minimum channel Width
<b>xwGate</b>	10.00	private	Maximum channel Width
<b>mlGate</b>	0.18	private	Minimum channel Length
<b>xlGate</b>	0.5	private	Maximum channel Length
<b>xnFingers</b>	64	private	Maximum number of fingers
<b>msDiff</b>	\$mos_msDiff	private	Minimum diffusion spacing
<b>mxGate</b>	\$mos_mxGate	private	Minimum gate extension
<b>msGate</b>	\$mos_msGate	private	Minimum gate spacing
<b>meDiffGate</b>	\$mos_meDiffGate	private	Minimum diffusion enclosure of gate
<b>msGateCont</b>	\$mos_msGateCont	private	Minimum gate to contact spacing
<b>mwPoly</b>	\$mos_mwPoly	private	Mnimum poly width on field
<b>msDiffPoly</b>	\$mos_msDiffPoly	private	Minimum diffusion to poly spacing
<b>mwCont</b>	\$mos_mwCont	private	Minimum contact width
<b>msCont</b>	\$mos_msCont	private	Minimum contact spacing
<b>masCont</b>	\$mos_masCont	private	Minimum contact array spacing
<b>meDiffCont</b>	\$mos_meDiffCont	private	Minimum diffusion enclosure of contact
<b>meeDiffCont</b>	\$mos_meeDiffCont	private	Minimum diffusion end enclosure of contact
<b>mePolyCont</b>	\$mos_mePolyCont	private	Minimum poly enclure of contact
<b>meePolyCont</b>	\$mos_meePolyCont	private	Minimum poly end enclosure of contact
<b>mwM1</b>	\$mos_mwM1	private	Minimum metal 1 width
<b>msM1</b>	\$mos_msM1	private	Minimum metal 1 spacing
<b>meM1Cont</b>	\$mos_meM1Cont	private	Minimum metal 1 enclosure of contact
<b>meeM1Cont</b>	\$mos_meeM1Cont	private	Minimum metal 1 end enclosure of contact

CDF device params

<b>meImplDiff</b>	\$nmos_meImplDiff	private	Minimum implant enclosure of diffusion
<b>meImplGate</b>	\$nmos_meImplGate	private	Minimum implant enclosure of gate
<b>meImplGate</b>	\$nmos_meImplGate	private	Minimum implant end enclosure of gate
<b>meImplPoly</b>	\$nmos_meImplPoly	private	Minimum implant enclosure of poly on field
<b>meImplCont</b>	\$nmos_meImplCont	private	Minimum implant enclosure of contact
<b>meWellDiff</b>	\$nmos_meWellDiff	private	Minimum well enclosure of diffusion
<b>meVoltDiff</b>	\$nmos_meVoltDiff	private	Minimum thick oxide enclosure of thin oxide diffusion
<b>meVoltImpl</b>	\$nmos_meVoltImpl	private	Minimum thick oxide enclosure of implant

## CDF gpd180 nmosrf

include macro mos4RF

CDF siminfo

<b>simulator</b>	auLvs
<b>componentName</b>	"\$cell"

CDF siminfo

<b>simulator</b>	auCdl
<b>componentName</b>	nil
<b>modelName</b>	"\$cell"

CDF cellview

<b>symbol</b>	baseline_gpd1	nmosrf	symbol	
<b>auCdl</b>	baseline_gpd1	nmosrf	symbol	
<b>auLvs</b>	baseline_gpd1	nmosrf	symbol	
<b>spectre</b>	baseline_gpd1	nmosrf	symbol	
<b>ads</b>	baseline_gpd1	nmosrf	symbol	
<b>ivpcell</b>	\$lib	\$cell	symbol	5

CDF device params

<b>msTapDiff</b>	\$tap_nmos_msTapDiff	private	Minimum tap diffusion to device diffusion spacing
<b>msTapImpl</b>	\$tap_nmos_msTapImpl	private	Minimum tap diffusion to device implant spacing
<b>meTimpTap</b>	\$tap_nmos_meTimpTap	private	Minimum tap implant enclosure of tap diffusion
<b>msTimpDiff</b>	\$tap_nmos_msTimpDiff	private	Minimum tap implant to device diffusion spacing
<b>msTimpImpl</b>	\$tap_nmos_msTimpImpl	private	Minimum tap implant to device implant spacing
<b>maTap</b>	\$tap_nmos_maTap	private	Minimum tap diffusion area
<b>maTimp</b>	\$tap_nmos_maTimp	private	Minimum tap implant area
<b>meWellTap</b>	\$tap_nmos_meWellTap	private	Minimum well enclosure of tap diffusion
<b>mwltap</b>	\$tap_nmos_mwltap	private	Minimum width of integrated tap
<b>msltapGate</b>	\$tap_nmos_msltapGate	private	Minimum integrated tap spacing to gate
<b>malimp</b>	\$tap_nmos_malimp	private	Minimum integrated tap implant area
<b>meltimpltap</b>	\$tap_nmos_meltimpltap	private	Minimum integrated tap implant enclosure of tap
<b>meltimpCont</b>	\$tap_nmos_meltimpCont	private	Minimum integrated tap enclosure of contact
<b>msTapVolt</b>	\$tap_nmos_msTapVolt	public	Minimum tap diffusion to thick oxide spacing

## pmos3

CDF gpd180 pmos3

CDF device params

<b>grid</b>	\$grid	private	MFG Grid
<b>scale</b>	1.0e-6	private	Dimension scale factor
<b>model</b>	"pmos1"	private	Device Model Name
<b>category</b>	"mos"	private	Library Manager Category
<b>perimGateEdge</b>	\$mos_perimGateEdge	private	Use four sides to calculate ps, pd
<b>aggregateFingers</b>	\$mos_aggregateFingers	private	Aggregate fingers: affects simM, as,ad,ps,pd

CDF device params

<b>dwGate</b>	2.0	private	Default channel width
<b>mwGate</b>	{2A}	private	Minimum channel Width
<b>xwGate</b>	50.00	private	Maximum channel Width
<b>mlGate</b>	{5A}	private	Minimum channel Length
<b>msDiff</b>	\$mos_msDiff	private	Minimum diffusion spacing
<b>mxGate</b>	\$mos_mxGate	private	Minimum gate extension
<b>msGate</b>	\$mos_msGate	private	Minimum gate spacing
<b>meDiffGate</b>	\$mos_meDiffGate	private	Minimum diffusion enclosure of gate
<b>msGateCont</b>	\$mos_msGateCont	private	Minimum gate to contact spacing
<b>mwPoly</b>	\$mos_mwPoly	private	Minimum poly width on field
<b>msDiffPoly</b>	\$mos_msDiffPoly	private	Minimum diffusion to poly spacing
<b>mwCont</b>	\$mos_mwCont	private	Minimum contact width
<b>msCont</b>	\$mos_msCont	private	Minimum contact spacing
<b>masCont</b>	\$mos_masCont	private	Minimum contact array spacing
<b>meDiffCont</b>	\$mos_meDiffCont	private	Minimum diffusion enclosure of contact
<b>meeDiffCont</b>	\$mos_meeDiffCont	private	Minimum diffusion end enclosure of contact
<b>mePolyCont</b>	\$mos_mePolyCont	private	Minimum poly enclosure of contact
<b>meePolyCont</b>	\$mos_meePolyCont	private	Minimum poly end enclosure of contact
<b>mwM1</b>	\$mos_mwM1	private	Minimum metal 1 width
<b>msM1</b>	\$mos_msM1	private	Minimum metal 1 spacing
<b>meM1Cont</b>	\$mos_meM1Cont	private	Minimum metal 1 enclosure of contact
<b>meeM1Cont</b>	\$mos_meeM1Cont	private	Minimum metal 1 end enclosure of contact

CDF device params

<b>meImplDiff</b>	\$pmos_meImplDiff	private	Minimum implant enclosure of diffusion
<b>meImplGate</b>	\$pmos_meImplGate	private	Minimum implant enclosure of gate
<b>meImplGate</b>	\$pmos_meImplGate	private	Minimum implant end enclosure of gate
<b>meImplPoly</b>	\$pmos_meImplPoly	private	Minimum implant enclosure of poly on field
<b>meImplCont</b>	\$pmos_meImplCont	private	Minimum implant enclosure of contact
<b>meWellDiff</b>	\$pmos_meWellDiff	private	Minimum well enclosure of diffusion

CDF gpdk180 pmos3

include macro mos3p

CDF siminfo

<b>simulator</b>	auLvs
<b>componentName</b>	"pmos"

CDF siminfo

<b>simulator</b>	auCdl
<b>componentName</b>	nil
<b>modelName</b>	"pmos"

CDF cellview

<b>symbol</b>	pas_std	pmos3	symbol
<b>auCdl</b>	pas_std	pmos3	spectre
<b>auLvs</b>	pas_std	pmos3	spectre
<b>spectre</b>	pas_std	pmos3	spectre
<b>ads</b>	pas_std	pmos3	spectre



## pmos

CDF gpd180 pmos

CDF device params

<b>grid</b>	\$grid	private	MFG Grid
<b>scale</b>	1.0e-6	private	Dimension scale factor
<b>model</b>	"pmos1"	private	Device Model Name
<b>category</b>	"mos"	private	Library Manager Category
<b>perimGateEdge</b>	\$mos_perimGateEdge	private	Use four sides to calculate ps, pd
<b>aggregateFingers</b>	\$mos_aggregateFingers	private	Aggregate fingers: affects simM, as,ad,ps,pd

CDF device params

<b>dwGate</b>	2.0	private	Default channel width
<b>mwGate</b>	{2A}	private	Minimum channel Width
<b>xwGate</b>	50.00	private	Maximum channel Width
<b>mlGate</b>	{5A}	private	Minimum channel Length
<b>msDiff</b>	\$mos_msDiff	private	Minimum diffusion spacing
<b>mxGate</b>	\$mos_mxGate	private	Minimum gate extension
<b>msGate</b>	\$mos_msGate	private	Minimum gate spacing
<b>meDiffGate</b>	\$mos_meDiffGate	private	Minimum diffusion enclosure of gate
<b>msGateCont</b>	\$mos_msGateCont	private	Minimum gate to contact spacing
<b>mwPoly</b>	\$mos_mwPoly	private	Minimum poly width on field
<b>msDiffPoly</b>	\$mos_msDiffPoly	private	Minimum diffusion to poly spacing
<b>mwCont</b>	\$mos_mwCont	private	Minimum contact width
<b>msCont</b>	\$mos_msCont	private	Minimum contact spacing
<b>masCont</b>	\$mos_masCont	private	Minimum contact array spacing
<b>meDiffCont</b>	\$mos_meDiffCont	private	Minimum diffusion enclosure of contact
<b>meeDiffCont</b>	\$mos_meeDiffCont	private	Minimum diffusion end enclosure of contact
<b>mePolyCont</b>	\$mos_mePolyCont	private	Minimum poly enclosure of contact
<b>meePolyCont</b>	\$mos_meePolyCont	private	Minimum poly end enclosure of contact
<b>mwM1</b>	\$mos_mwM1	private	Minimum metal 1 width
<b>msM1</b>	\$mos_msM1	private	Minimum metal 1 spacing
<b>meM1Cont</b>	\$mos_meM1Cont	private	Minimum metal 1 enclosure of contact
<b>meeM1Cont</b>	\$mos_meeM1Cont	private	Minimum metal 1 end enclosure of contact

CDF device params

<b>meImplDiff</b>	\$pmos_meImplDiff	private	Minimum implant enclosure of diffusion
<b>meImplGate</b>	\$pmos_meImplGate	private	Minimum implant enclosure of gate
<b>meImplGate</b>	\$pmos_meImplGate	private	Minimum implant end enclosure of gate
<b>meImplPoly</b>	\$pmos_meImplPoly	private	Minimum implant enclosure of poly on field
<b>meImplCont</b>	\$pmos_meImplCont	private	Minimum implant enclosure of contact
<b>meWellDiff</b>	\$pmos_meWellDiff	private	Minimum well enclosure of diffusion

## CDF gpd180 pmos

include macro mos4

CDF siminfo

<b>simulator</b>	auLvs
<b>componentName</b>	"\$cell"

CDF siminfo

<b>simulator</b>	auCdl
<b>componentName</b>	nil
<b>modelName</b>	"\$cell"

CDF cellview

<b>symbol</b>	pas_std	pmos4	symbol	
<b>auCdl</b>	pas_std	pmos4	symbol	
<b>auLvs</b>	pas_std	pmos4	symbol	
<b>spectre</b>	pas_std	pmos4	symbol	
<b>ads</b>	pas_std	pmos4	symbol	
<b>ivpcell</b>	\$lib	\$cell	symbol	5

CDF device params

<b>msTapDiff</b>	\$tap_pmos_msTapDiff	private	Minimum tap diffusion to device diffusion spacing
<b>msTapImpl</b>	\$tap_pmos_msTapImpl	private	Minimum tap diffusion to device implant spacing
<b>meTimpTap</b>	\$tap_pmos_meTimpTap	private	Minimum tap implant enclosure of tap diffusion
<b>msTimpDiff</b>	\$tap_pmos_msTimpDiff	private	Minimum tap implant to device diffusion spacing
<b>msTimpImpl</b>	\$tap_pmos_msTimpImpl	private	Minimum tap implant to device implant spacing
<b>maTap</b>	\$tap_pmos_maTap	private	Minimum tap diffusion area
<b>maTimp</b>	\$tap_pmos_maTimp	private	Minimum tap implant area
<b>meWellTap</b>	\$tap_pmos_meWellTap	private	Minimum well enclosure of tap diffusion
<b>mwltap</b>	\$tap_pmos_mwltap	private	Minimum width of integrated tap
<b>msltapGate</b>	\$tap_pmos_msltapGate	private	Minimum integrated tap spacing to gate
<b>maltemp</b>	\$tap_pmos_maltemp	private	Minimum integrated tap implant area
<b>meltempTap</b>	\$tap_pmos_meltempTap	private	Minimum integrated tap implant enclosure of tap
<b>meltempCont</b>	\$tap_pmos_meltempCont	private	Minimum integrated tap enclosure of contact
<b>msTapVolt</b>	\$tap_pmos_msTapVolt	public	Minimum tap diffusion to thick oxide spacing

## pmos3hv

CDF gpd180 pmos3hv

CDF device params

<b>grid</b>	\$grid	private	MFG Grid
<b>scale</b>	1.0e-6	private	Dimension scale factor
<b>model</b>	"pmos25"	private	Device Model Name
<b>category</b>	"mos"	private	Library Manager Category
<b>perimGateEdge</b>	\$mos_perimGateEdge	private	Use four sides to calculate ps, pd
<b>aggregateFingers</b>	\$mos_aggregateFingers	private	Aggregate fingers: affects simM, as,ad,ps,pd

CDF device params

<b>dwGate</b>	2.0	private	Default channel width
<b>mwGate</b>	{2A}	private	Minimum channel Width
<b>xwGate</b>	50.00	private	Maximum channel Width
<b>mlGate</b>	{5A}	private	Minimum channel Length
<b>msDiff</b>	\$mos_msDiff	private	Minimum diffusion spacing
<b>mxGate</b>	\$mos_mxGate	private	Minimum gate extension
<b>msGate</b>	\$mos_msGate	private	Minimum gate spacing
<b>meDiffGate</b>	\$mos_meDiffGate	private	Minimum diffusion enclosure of gate
<b>msGateCont</b>	\$mos_msGateCont	private	Minimum gate to contact spacing
<b>mwPoly</b>	\$mos_mwPoly	private	Minimum poly width on field
<b>msDiffPoly</b>	\$mos_msDiffPoly	private	Minimum diffusion to poly spacing
<b>mwCont</b>	\$mos_mwCont	private	Minimum contact width
<b>msCont</b>	\$mos_msCont	private	Minimum contact spacing
<b>masCont</b>	\$mos_masCont	private	Minimum contact array spacing
<b>meDiffCont</b>	\$mos_meDiffCont	private	Minimum diffusion enclosure of contact
<b>meeDiffCont</b>	\$mos_meeDiffCont	private	Minimum diffusion end enclosure of contact
<b>mePolyCont</b>	\$mos_mePolyCont	private	Minimum poly enclosure of contact
<b>meePolyCont</b>	\$mos_meePolyCont	private	Minimum poly end enclosure of contact
<b>mwM1</b>	\$mos_mwM1	private	Minimum metal 1 width
<b>msM1</b>	\$mos_msM1	private	Minimum metal 1 spacing
<b>meM1Cont</b>	\$mos_meM1Cont	private	Minimum metal 1 enclosure of contact
<b>meeM1Cont</b>	\$mos_meeM1Cont	private	Minimum metal 1 end enclosure of contact

CDF device params

<b>meImplDiff</b>	\$pmos_meImplDiff	private	Minimum implant enclosure of diffusion
<b>meImplGate</b>	\$pmos_meImplGate	private	Minimum implant enclosure of gate
<b>meImplGate</b>	\$pmos_meImplGate	private	Minimum implant end enclosure of gate
<b>meImplPoly</b>	\$pmos_meImplPoly	private	Minimum implant enclosure of poly on field
<b>meImplCont</b>	\$pmos_meImplCont	private	Minimum implant enclosure of contact
<b>meWellDiff</b>	\$pmos_meWellDiff	private	Minimum well enclosure of diffusion
<b>meVoltDiff</b>	\$pmos_meVoltDiff	private	Minimum thick oxide enclosure of thin oxide diffusion
<b>meVoltImpl</b>	\$pmos_meVoltImpl	private	Minimum thick oxide enclosure of implant

CDF gpdk180 pmos3hv

include macro mos3p

CDF siminfo

<b>simulator</b>	auLvs
<b>componentName</b>	"pmoshv"

CDF siminfo

<b>simulator</b>	auCdl
<b>componentName</b>	nil
<b>modelName</b>	"pmoshv"

CDF cellview

<b>symbol</b>	pas_std	pmos3_hv	symbol
<b>auCdl</b>	pas_std	pmos3_hv	spectre
<b>auLvs</b>	pas_std	pmos3_hv	spectre
<b>spectre</b>	pas_std	pmos3_hv	spectre
<b>ads</b>	pas_std	pmos3_hv	spectre

**pmoshv**

CDF gpdk180 pmoshv

CDF device params

<b>grid</b>	\$grid	private	MFG Grid
<b>scale</b>	1.0e-6	private	Dimension scale factor
<b>model</b>	"pmos25"	private	Device Model Name
<b>category</b>	"mos"	private	Library Manager Category
<b>perimGateEdge</b>	\$mos_perimGateEdge	private	Use four sides to calculate ps, pd
<b>aggregateFingers</b>	\$mos_aggregateFingers	private	Aggregate fingers: affects simM, as,ad,ps,pd

CDF device params

<b>dwGate</b>	2.0	private	Default channel width
<b>mwGate</b>	{2A}	private	Minimum channel Width
<b>xwGate</b>	50.0	private	Maximum channel Width
<b>mlGate</b>	{5A}	private	Minimum channel Length
<b>msDiff</b>	\$mos_msDiff	private	Minimum diffusion spacing
<b>mxGate</b>	\$mos_mxGate	private	Minimum gate extension
<b>msGate</b>	\$mos_msGate	private	Minimum gate spacing
<b>meDiffGate</b>	\$mos_meDiffGate	private	Minimum diffusion enclosure of gate
<b>msGateCont</b>	\$mos_msGateCont	private	Minimum gate to contact spacing
<b>mwPoly</b>	\$mos_mwPoly	private	Minimum poly width on field
<b>msDiffPoly</b>	\$mos_msDiffPoly	private	Minimum diffusion to poly spacing
<b>mwCont</b>	\$mos_mwCont	private	Minimum contact width
<b>msCont</b>	\$mos_msCont	private	Minimum contact spacing
<b>masCont</b>	\$mos_masCont	private	Minimum contact array spacing
<b>meDiffCont</b>	\$mos_meDiffCont	private	Minimum diffusion enclosure of contact
<b>meeDiffCont</b>	\$mos_meeDiffCont	private	Minimum diffusion end enclosure of contact
<b>mePolyCont</b>	\$mos_mePolyCont	private	Minimum poly enclosure of contact
<b>meePolyCont</b>	\$mos_meePolyCont	private	Minimum poly end enclosure of contact
<b>mwM1</b>	\$mos_mwM1	private	Minimum metal 1 width
<b>msM1</b>	\$mos_msM1	private	Minimum metal 1 spacing
<b>meM1Cont</b>	\$mos_meM1Cont	private	Minimum metal 1 enclosure of contact
<b>meeM1Cont</b>	\$mos_meeM1Cont	private	Minimum metal 1 end enclosure of contact

CDF device params

<b>meImplDiff</b>	\$pmos_meImplDiff	private	Minimum implant enclosure of diffusion
<b>meImplGate</b>	\$pmos_meImplGate	private	Minimum implant enclosure of gate
<b>meImplGate</b>	\$pmos_meImplGate	private	Minimum implant end enclosure of gate
<b>meImplPoly</b>	\$pmos_meImplPoly	private	Minimum implant enclosure of poly on field
<b>meImplCont</b>	\$pmos_meImplCont	private	Minimum implant enclosure of contact
<b>meWellDiff</b>	\$pmos_meWellDiff	private	Minimum well enclosure of diffusion
<b>meVoltDiff</b>	\$pmos_meVoltDiff	private	Minimum thick oxide enclosure of thin oxide diffusion
<b>meVoltImpl</b>	\$pmos_meVoltImpl	private	Minimum thick oxide enclosure of implant

# CDF gpdk180 pmoshv

include macro mos4

CDF siminfo

<b>simulator</b>	auLvs
<b>componentName</b>	"\$cell"

CDF siminfo

<b>simulator</b>	auCdl
<b>componentName</b>	nil
<b>modelName</b>	"\$cell"

CDF cellview

<b>symbol</b>	pas_std	pmos4_hv	symbol	
<b>auCdl</b>	pas_std	pmos4_hv	symbol	
<b>auLvs</b>	pas_std	pmos4_hv	symbol	
<b>spectre</b>	pas_std	pmos4_hv	symbol	
<b>ads</b>	pas_std	pmos4_hv	symbol	
<b>ivpcell</b>	\$lib	\$cell	symbol	5

CDF device params

<b>msTapDiff</b>	\$tap_pmos_msTapDiff	private	Minimum tap diffusion to device diffusion spacing
<b>msTapImpl</b>	\$tap_pmos_msTapImpl	private	Minimum tap diffusion to device implant spacing
<b>meTimpTap</b>	\$tap_pmos_meTimpTap	private	Minimum tap implant enclosure of tap diffusion
<b>msTimpDiff</b>	\$tap_pmos_msTimpDiff	private	Minimum tap implant to device diffusion spacing
<b>msTimpImpl</b>	\$tap_pmos_msTimpImpl	private	Minimum tap implant to device implant spacing
<b>maTap</b>	\$tap_pmos_maTap	private	Minimum tap diffusion area
<b>maTimp</b>	\$tap_pmos_maTimp	private	Minimum tap implant area
<b>meWellTap</b>	\$tap_pmos_meWellTap	private	Minimum well enclosure of tap diffusion
<b>mwltap</b>	\$tap_pmos_mwltap	private	Minimum width of integrated tap
<b>msltapGate</b>	\$tap_pmos_msltapGate	private	Minimum integrated tap spacing to gate
<b>maltemp</b>	\$tap_pmos_maltemp	private	Minimum integrated tap implant area
<b>meltempTap</b>	\$tap_pmos_meltempTap	private	Minimum integrated tap implant enclosure of tap
<b>meltempCont</b>	\$tap_pmos_meltempCont	private	Minimum integrated tap enclosure of contact
<b>msTapVolt</b>	\$tap_pmos_msTapVolt	public	Minimum tap diffusion to thick oxide spacing

CDF gpd180 pmosrf

**pmosrf**

CDF device params

<b>grid</b>	\$grid	private	MFG Grid
<b>scale</b>	1.0e-6	private	Dimension scale factor
<b>model</b>	"pmos_rf"	private	Device Model Name
<b>category</b>	"mos"	private	Library Manager Category
<b>perimGateEdge</b>	\$mos_perimGateEdge	private	Use four sides to calculate ps, pd
<b>aggregateFingers</b>	\$mos_aggregateFingers	private	Aggregate fingers: affects simM, as,ad,ps,pd

CDF device params

<b>dwGate</b>	2.0	private	Default channel width
<b>mwGate</b>	1.5	private	Minimum channel Width
<b>xwGate</b>	10.0	private	Maximum channel Width
<b>mlGate</b>	0.18	private	Minimum channel Length
<b>xlGate</b>	0.5	private	Maximum channel Length
<b>xnFingers</b>	64	private	Maximum number of fingers
<b>msDiff</b>	\$mos_msDiff	private	Minimum diffusion spacing
<b>mxGate</b>	\$mos_mxGate	private	Minimum gate extension
<b>msGate</b>	\$mos_msGate	private	Minimum gate spacing
<b>meDiffGate</b>	\$mos_meDiffGate	private	Minimum diffusion enclosure of gate
<b>msGateCont</b>	\$mos_msGateCont	private	Minimum gate to contact spacing
<b>mwPoly</b>	\$mos_mwPoly	private	Minimum poly width on field
<b>msDiffPoly</b>	\$mos_msDiffPoly	private	Minimum diffusion to poly spacing
<b>mwCont</b>	\$mos_mwCont	private	Minimum contact width
<b>msCont</b>	\$mos_msCont	private	Minimum contact spacing
<b>masCont</b>	\$mos_masCont	private	Minimum contact array spacing
<b>meDiffCont</b>	\$mos_meDiffCont	private	Minimum diffusion enclosure of contact
<b>meeDiffCont</b>	\$mos_meeDiffCont	private	Minimum diffusion end enclosure of contact
<b>mePolyCont</b>	\$mos_mePolyCont	private	Minimum poly enclosure of contact
<b>meePolyCont</b>	\$mos_meePolyCont	private	Minimum poly end enclosure of contact
<b>mwM1</b>	\$mos_mwM1	private	Minimum metal 1 width
<b>msM1</b>	\$mos_msM1	private	Minimum metal 1 spacing
<b>meM1Cont</b>	\$mos_meM1Cont	private	Minimum metal 1 enclosure of contact
<b>meeM1Cont</b>	\$mos_meeM1Cont	private	Minimum metal 1 end enclosure of contact

CDF device params

<b>meImplDiff</b>	\$pmos_meImplDiff	private	Minimum implant enclosure of diffusion
<b>meImplGate</b>	\$pmos_meImplGate	private	Minimum implant enclosure of gate
<b>meImplGate</b>	\$pmos_meImplGate	private	Minimum implant end enclosure of gate
<b>meImplPoly</b>	\$pmos_meImplPoly	private	Minimum implant enclosure of poly on field
<b>meImplCont</b>	\$pmos_meImplCont	private	Minimum implant enclosure of contact
<b>meWellDiff</b>	\$pmos_meWellDiff	private	Minimum well enclosure of diffusion
<b>meVoltDiff</b>	\$pmos_meVoltDiff	private	Minimum thick oxide enclosure of thin oxide diffusion
<b>meVoltImpl</b>	\$pmos_meVoltImpl	private	Minimum thick oxide enclosure of implant

## CDF gpd180 pmosrf

include macro mos4RF

CDF siminfo

<b>simulator</b>	auLvs
<b>componentName</b>	"\$cell"

CDF siminfo

<b>simulator</b>	auCdl
<b>componentName</b>	nil
<b>modelName</b>	"\$cell"

CDF cellview

<b>symbol</b>	baseline_gpd1	pmosrf	symbol	
<b>auCdl</b>	baseline_gpd1	pmosrf	symbol	
<b>auLvs</b>	baseline_gpd1	pmosrf	symbol	
<b>spectre</b>	baseline_gpd1	pmosrf	symbol	
<b>ads</b>	baseline_gpd1	pmosrf	symbol	
<b>ivpcell</b>	\$lib	\$cell	symbol	5

CDF device params

<b>msTapDiff</b>	\$tap_pmos_msTapDiff	private	Minimum tap diffusion to device diffusion spacing
<b>msTapImpl</b>	\$tap_pmos_msTapImpl	private	Minimum tap diffusion to device implant spacing
<b>meTimpTap</b>	\$tap_pmos_meTimpTap	private	Minimum tap implant enclosure of tap diffusion
<b>msTimpDiff</b>	\$tap_pmos_msTimpDiff	private	Minimum tap implant to device diffusion spacing
<b>msTimpImpl</b>	\$tap_pmos_msTimpImpl	private	Minimum tap implant to device implant spacing
<b>maTap</b>	\$tap_pmos_maTap	private	Minimum tap diffusion area
<b>maTimp</b>	\$tap_pmos_maTimp	private	Minimum tap implant area
<b>meWellTap</b>	\$tap_pmos_meWellTap	private	Minimum well enclosure of tap diffusion
<b>mwltap</b>	\$tap_pmos_mwltap	private	Minimum width of integrated tap
<b>msltapGate</b>	\$tap_pmos_msltapGate	private	Minimum integrated tap spacing to gate
<b>malimp</b>	\$tap_pmos_malimp	private	Minimum integrated tap implant area
<b>melimpltap</b>	\$tap_pmos_melimpltap	private	Minimum integrated tap implant enclosure of tap
<b>melimpCont</b>	\$tap_pmos_melimpCont	private	Minimum integrated tap enclosure of contact
<b>msTapVolt</b>	\$tap_pmos_msTapVolt	public	Minimum tap diffusion to thick oxide spacing



**PCells****PCell SKILL Procedures****MOS Abutment****Skill Procedures**

```

procedure( gpdK180_mosAbut( iA iB pA pB pASide connection event @optional (group nil)
) ;_Jun 25 03 amir 186
let( (
    apA apB avA avB delta
    ( epsilon 0.0001 )
)

case( event
( 1 ;; compute abutment offset
0.0
)
( 2 ;; adjust pcell for abutment
apA = pA~>abutParam
apB = pB~>abutParam
avA = get(iA apA)
avB = get(iB apB)
;; save current state
dbReplaceProp( group
                sprintf(nil "abutInfo_%s" iA~>name)
                "list"
                list(
                    list(concat(apA) avA)
                )
)
dbReplaceProp( group
                sprintf(nil "abutInfo_%s" iB~>name)
                "list"
                list(
                    list(concat(apB) avB)
                )
)

case( connection
( 1 ;; no third connection, remove both contacts
delta = pB~>gateWidth - pA~>gateWidth
cond(
( delta > epsilon
avA = 6
avB = 2
)
( delta < -epsilon
avA = 2
avB = 6
)
( t
avA = 4
avB = 4
)
)
)
)...
```

## Skill Procedures

## Metal Coverage

```

procedure(gpdk180_mtlCvgUserFunc(info) ;_Dec 13 02 amir 511
  let((dbId col cols mtlCL inc newV oldV bot top max handle)
    dbId = info->rodObj->dbId
    col = dbId->col
    cols = dbId->cols
    mtlCL = evalstring(sprintf(nil ""( nil %s )" info->paramVal))
    inc = info->increment
    max = dbId->maxLength - dbId->minLength
    handle = concat(info->handleName)
    newV = ""
    for(i 0 cols
      oldV = get(mtlCL concat(i))
      if( i==col then
        bot = or(car(oldV) 0.0)
        top = or(cadr(oldV) 0.0)
        caseq( handle
          ( (lowerLeft lowerCenter lowerRight)
            max = max - top
            cond(
              ( bot - inc > max
                bot = max
                ;; fprintf( stderr "Unable to stretch beyond limits!\n" )
              )
              ( bot - inc < 0
                bot = 0
                ;; fprintf( stderr "Unable to stretch beyond limits!\n" )
              )
              ( t
                bot = bot - inc
              )
            ) ; ** cond not **
          )
          ( (upperLeft upperCenter upperRight)
            max = max - bot
            cond(
              ( top - inc > max
                top = max
                ;; fprintf( stderr "Unable to stretch beyond limits!\n" )
              )
              ( top - inc < 0
                top = 0
                ;; fprintf( stderr "Unable to stretch beyond limits!\n" )
              )
              ( t
                top = top - inc
              )
            ) ; ** cond not **
          )
        )
      ) ; ** caseq handle **
    )
    unless( and(zerop(top) zerop(bot))...

```

## PCell Macros

### mos (macro)

pcell macro mos

1: include

deviceProps

```
dpt = PasGetDeviceProps( cv )      ; device property table
scale = dpt->scale
epsilon = grid / 10.0

meDiffCont = dpt->meDiffCont
meeDiffCont = dpt->meeDiffCont || meDiffCont
meDiffGate = dpt->meDiffGate
msGateCont = dpt->msGateCont
msDiffPoly = dpt->msDiffPoly
msDiffGate = dpt->msDiffGate || msDiffPoly
mePolyCont = dpt->mePolyCont
meePolyCont = dpt->meePolyCont || mePolyCont

mxGate = dpt->mxGate
msGate = dpt->msGate

mwCont = dpt->mwCont
msCont = dpt->msCont
mwM1 = dpt->mwM1
msM1 = dpt->msM1
meM1Cont = dpt->meM1Cont
meeM1Cont = dpt->meeM1Cont || meM1Cont
```

pcell macro mos

4: include

```
gw = PasCeiling( ( fw / scale ) grid t )
gl = PasCeiling( ( l / scale ) grid t )
gc = max( fix( fingers ) 1 )
mIDiff = mwCont + ( 2.0 * meeDiffCont )
dogbone = ( gw + epsilon ) < mIDiff

awSDM = PasCeiling( ( sdMtlWidth / scale ) grid t )
cntSpan = awSDM - ( 2.0 * meM1Cont )
lwSDM = awSDM
rwSDM = awSDM

tmp = when( cntSpan + epsilon < mwCont
    cntSpan = mwCont
    awSDM = mwCont + ( 2.0 * meM1Cont )
    println("Assigned value for 'sdMtlWidth' is too small; value is overridden")
)

leftSource = not( switchSD )
rightSource = PasXor( evenp( gc ) switchSD )
leftCnt = nil
rightCnt = nil

lwDiff = nil   rwDiff = nil
lhStep = 0    rhStep = 0
```

pcell macro mos

6: include

1: if not( dogbone )

```

alDiff = gw
dwDiff = max( msGate ( cntSpan + ( 2.0 * msGateCont ) ) )      swDiff = dwDiff

daDC = PasCeiling( ( ( dwDiff - cntSpan ) / 2.0 ) grid t )
saDC = daDC      raDC = msGateCont      laDC = msGateCont

swStep = 0      swStep = 0      lwStep = 0      rwStep = 0
bhStep = 0      thStep = 0

lwDiff = max( meDiffGate ( meDiffCont + cntSpan + msGateCont ) )      rwDiff = lwDiff

```

2: if dogbone

```

alDiff = mlDiff
dwDiff = cntSpan + ( 2.0 * meDiffCont )      swDiff = dwDiff

daDC = meDiffCont      saDC = meDiffCont
raDC = meDiffCont      laDC = meDiffCont

dwStep = max( msDiffGate ( msGateCont - meDiffCont ) grid )
swStep = dwStep      lwStep = dwStep      rwStep = dwStep

bhStep = PasCeiling( ( ( alDiff - gw ) / 2.0 ) grid t )
thStep = alDiff - gw - bhStep

lwDiff = max( ( meDiffGate - lwStep ) ( meDiffCont + cntSpan + meDiffCont ) )      rwDiff = lwDiff

```

9: if leftCnt &amp;&amp; rightCnt

```

connectS = member(connectSD list("Source" "Both"))
connectD = member(connectSD list("Drain" "Both"))

```

pcell macro mos

8: include

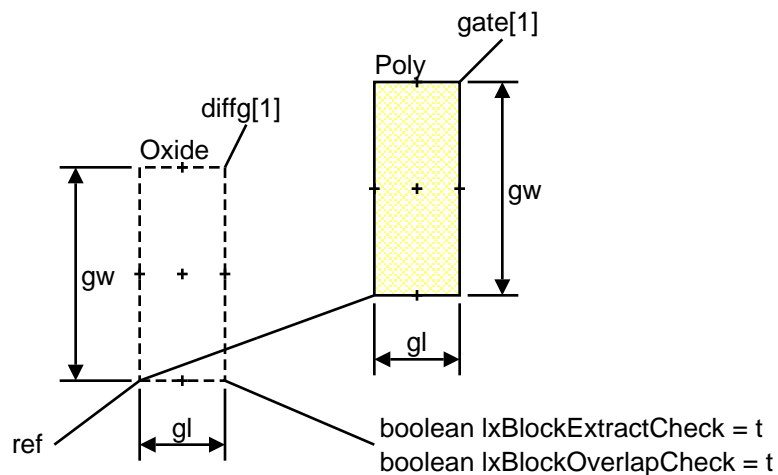
```

tmp = case( leftAbut
  ( 0      leftCnt = t )
  ( (-1 1) leftCnt = t )
  ( (-2 2) lwDiff = max( meDiffGate (msGate - msDiffPoly))  lhStep = bhStep )
  ( (-3 3) laDC = daDC  lwDiff = cntSpan + laDC  leftCnt = t )
  ( (-4 4) lwDiff = PasCeiling( (msGate/2.0) grid t )  lhStep = bhStep )
  ( (-5 5) lwDiff = daDC  lhStep = bhStep )
  ( (-6 6) lwDiff = msDiffPoly  lhStep = bhStep )
  ( (-7 7) lwDiff = max( msDiffGate ( msGateCont - meDiffCont ) )  lhStep = bhStep )
  ( t      error("leftAbut parameter value out of range!") )
)

tmp = case( rightAbut
  ( 0      rightCnt = t )
  ( (-1 1) rightCnt = t )
  ( (-2 2) rwDiff = max( meDiffGate (msGate - msDiffPoly))  rhStep = thStep )
  ( (-3 3) raDC = daDC  rwDiff = cntSpan + raDC  rightCnt = t )
  ( (-4 4) rwDiff = PasCeiling( (msGate/2.0) grid t )  rhStep = thStep )
  ( (-5 5) rwDiff = daDC  rhStep = thStep )
  ( (-6 6) rwDiff = msDiffPoly  rhStep = thStep )
  ( (-7 7) rwDiff = max( msDiffGate ( msGateCont - meDiffCont ) )  rhStep = thStep )
  ( t      error("rightAbut parameter value out of range!") )
)

```

10: include

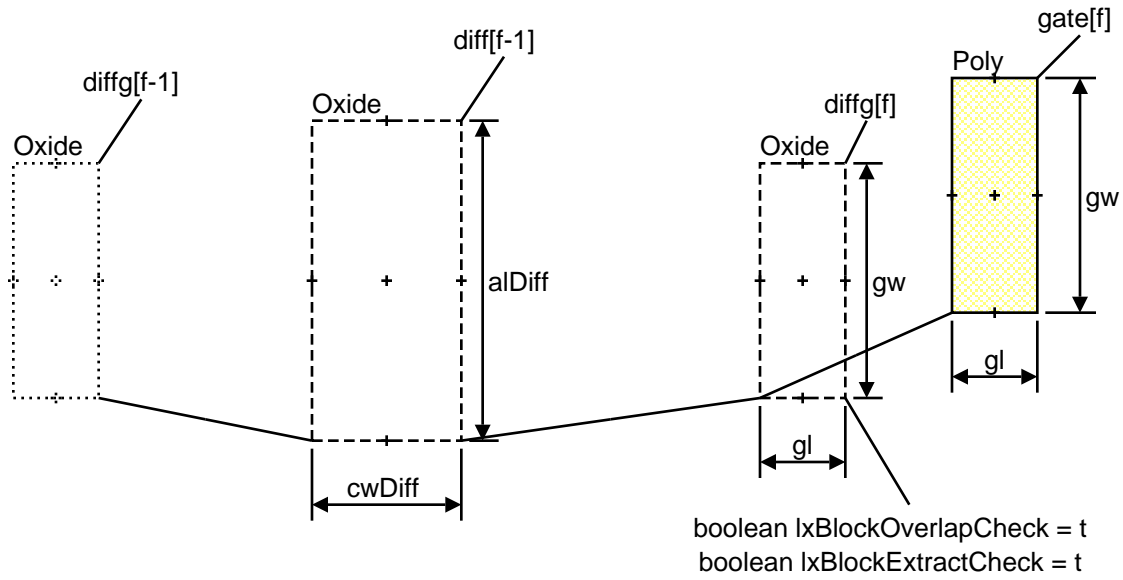


pcell macro mos

11: for f 2 gc

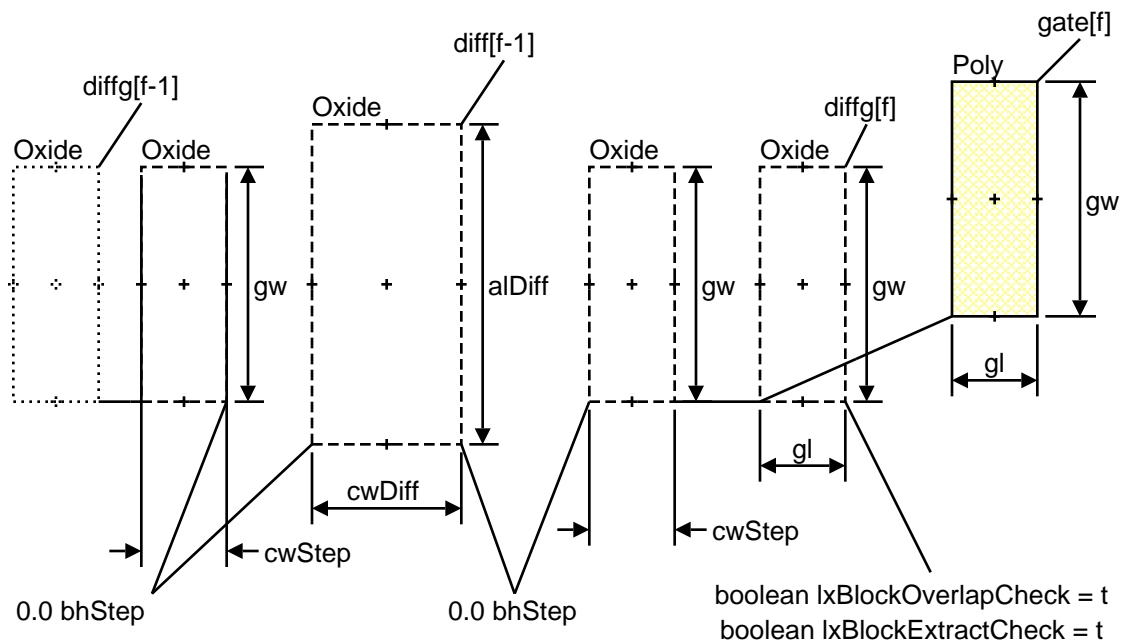
isSource = PasXor( evenp(f) switchSD )  
 cwDiff = if( isSource then swDiff else dwDiff )

1: if not(dogbone)



1: if dogbone

cwStep = if( isSource then swStep else dwStep )

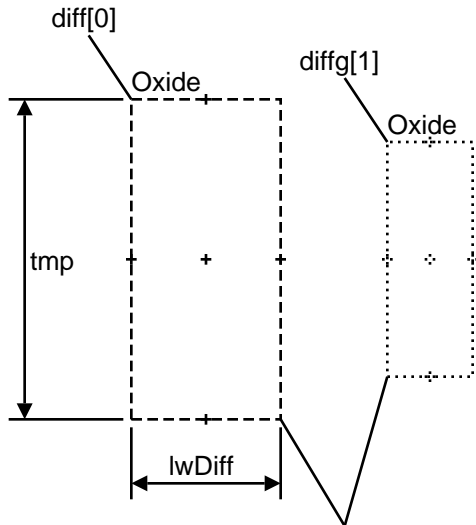


pcell macro mos

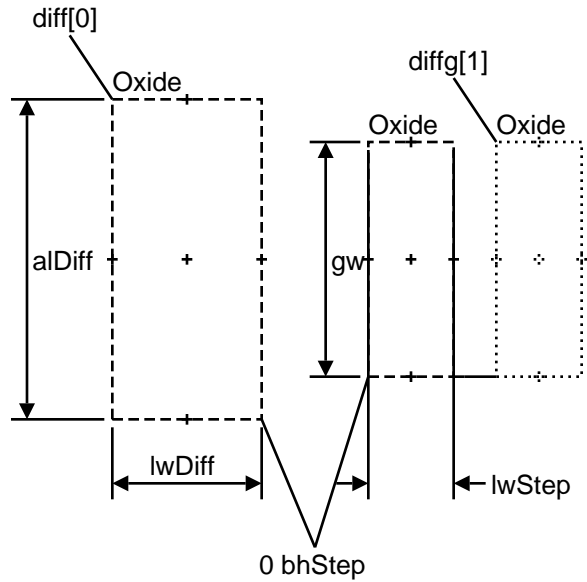
12: include

if not( dogbone && leftCnt )

tmp = if( dogbone then gw else alDiff )

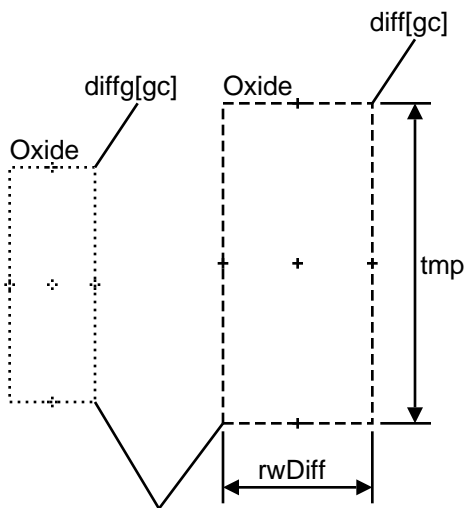


if dogbone && leftCnt

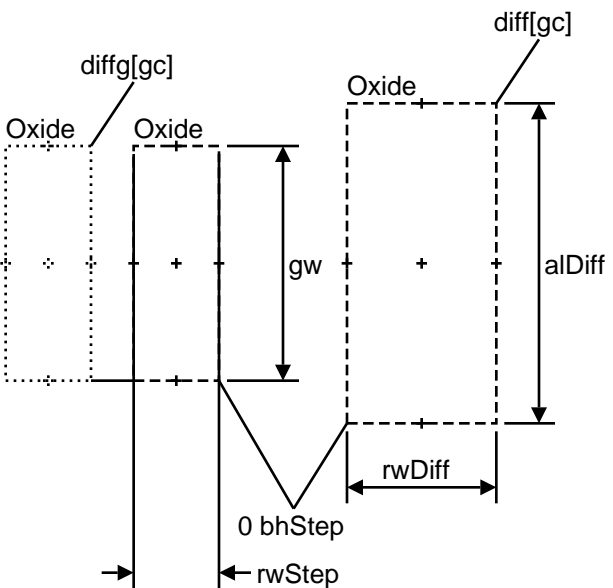


if not( dogbone && rightCnt )

tmp = if( dogbone then gw else alDiff )



if dogbone && rightCnt





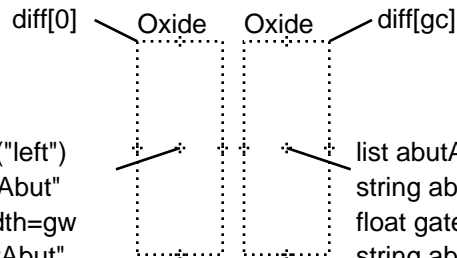
pcell macro mos

13: include

1: include

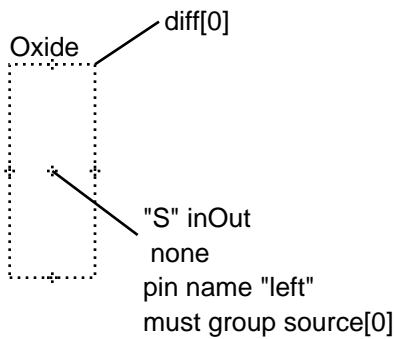
`tmp = dpt->msDiff || 0.0`

list abutAccessDir=list("left")  
string abutFunction="gpd180\_mosAbut"  
float gateWidth=gw  
string abutParam="leftAbut"  
boolean isSource=leftSource  
list vxllnstSpacingDir=list("left")  
float vxllnstSpacingRule=tmp

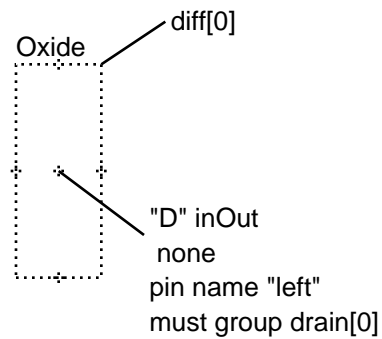


list abutAccessDir=list("right")  
string abutFunction="gpd180\_mosAbut"  
float gateWidth=gw  
string abutParam="rightAbut"  
boolean isSource=rightSource  
list vxllnstSpacingDir=list("right")  
float vxllnstSpacingRule=tmp

2: if leftSource

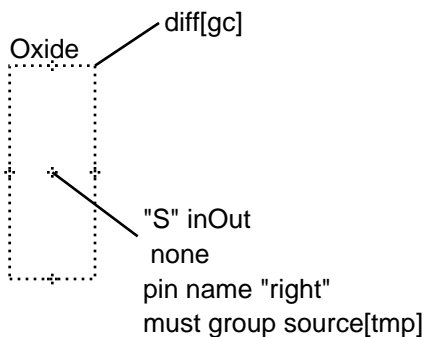


2: if not( leftSource )



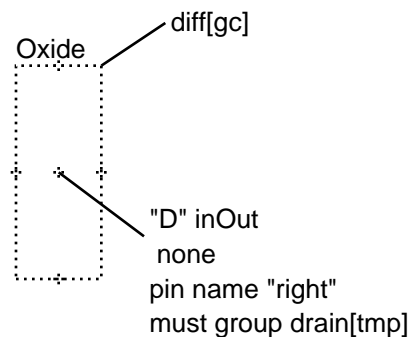
2: if rightSource

`tmp = if( connectS then 0 else gc )`



2: if not( rightSource )

`tmp = if( connectD then 0 else gc )`



pcell macro mos

14: include

```

ySep = meeDiffCont - meeM1Cont
mlM1 = mwCont + (2.0 * meeM1Cont)

xIM1 = gw - ySep - ySep
mtlCL = gpdk180_evalMtlCvg( mtlCvg grid )

```

for f 0 gc

1: if ( ( f > 0 && f < gc ) || ( f == 0 && leftCnt ) || ( f == gc && rightCnt ) ) mtlCvg

```

isSource = PasXor( evenp(f) switchSD )
delta = or( get( mtlCL concat( f ) ) list( 0.0 0.0 ) )
botSep = ySep + car( delta )
topSep = ySep + cadr( delta )
alM1 = alDiff - topSep - botSep
xSep = cond( ( f == 0 laDC ) ( f == gc raDC ) ( isSource saDC ) ( t daDC ) ) - meM1Cont
awM1 = cond( ( f==0 lwSDM ) ( f==gc rwSDM ) ( t awSDM ) )

```

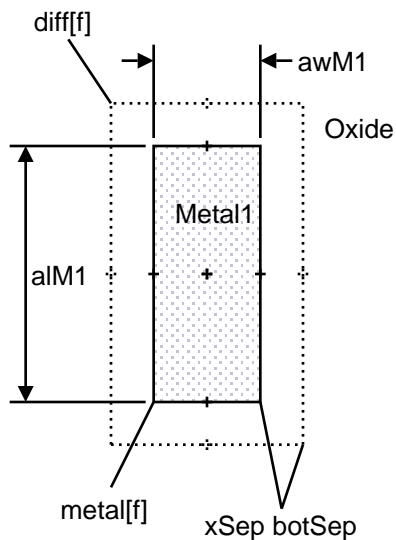
1: if alM1 + epsilon < mlM1

```

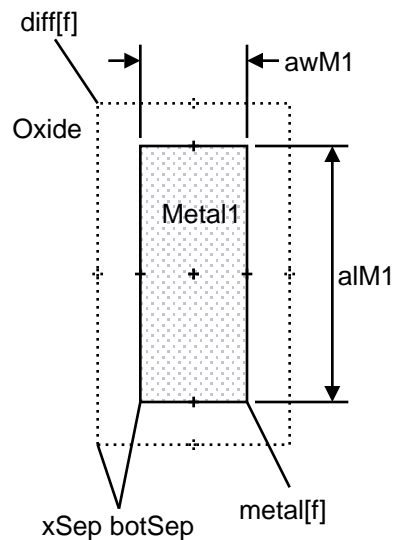
alM1 = xIM1
botSep = ySep
tmp = printf("Overconstraint metal coverage on column %d\n" f)

```

2: if f == 0



2: if f != 0

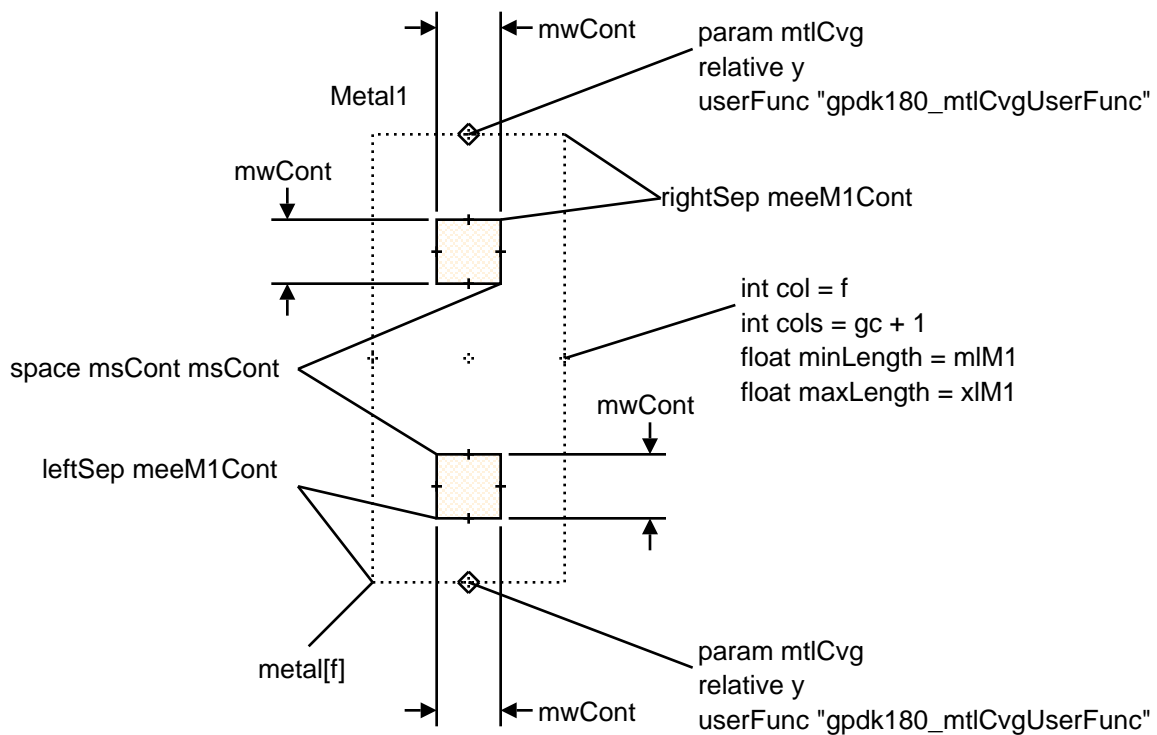


pcell macro mos

mtlCvg

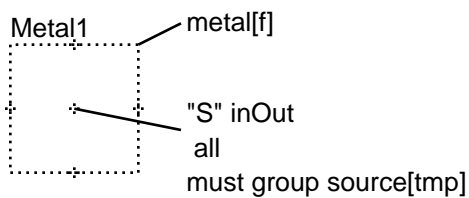
3: include

```
leftSep = cond( ( f==0  meM1Cont + lwSDM - awSDM ) ( t  meM1Cont ) )
rightSep = cond( ( f==gc  meM1Cont + rwSDM - awSDM ) ( t  meM1Cont ) )
```



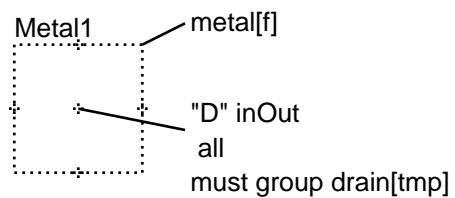
4: if isSource

```
tmp = if( connectS then 0 else f )
```



4: if not(isSource)

```
tmp = if( connectD then 0 else f )
```



pcell macro mos

16: if leftCnt && rightCnt

if connectS || connectD

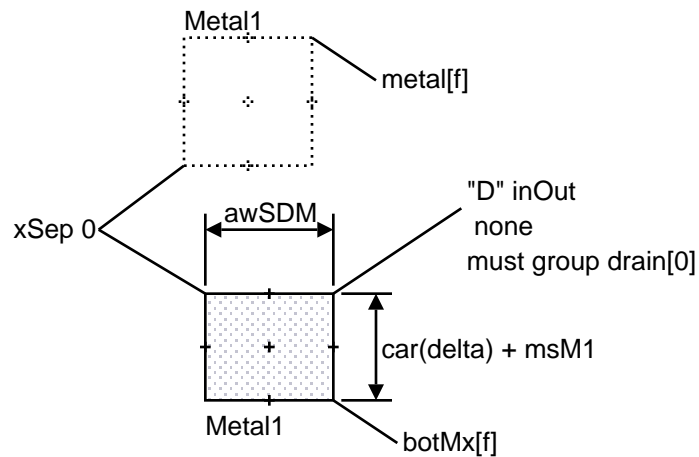
connectSD

for f 0 gc

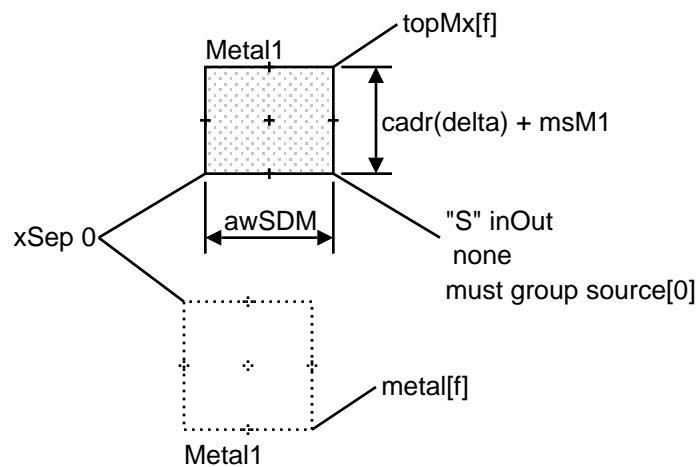
$\text{delta} = \text{or}(\text{get}(\text{mtlCL concat}(f)) \text{list}(0.0\ 0.0\ 0.0))$

$\text{xSep} = \text{cond}((f == 0 \text{ lwSDM} - \text{awSDM})(t\ 0))$

if connectD && PasXor(oddp(f) switchSD)



if connectS && PasXor(evenp(f) switchSD)



pcell macro mos

connectSD

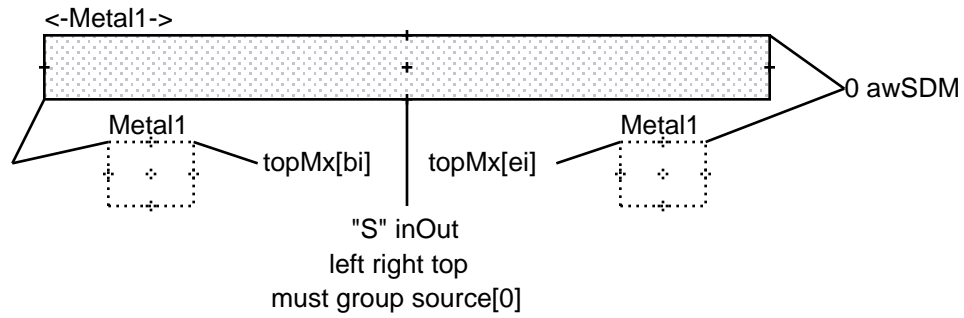
if connectS

connectS

1: include

```
bi = if( switchSD then 1 else 0 )
ei = if( PasXor( evenp( gc ) switchSD ) then gc else gc - 1 )
```

2: include



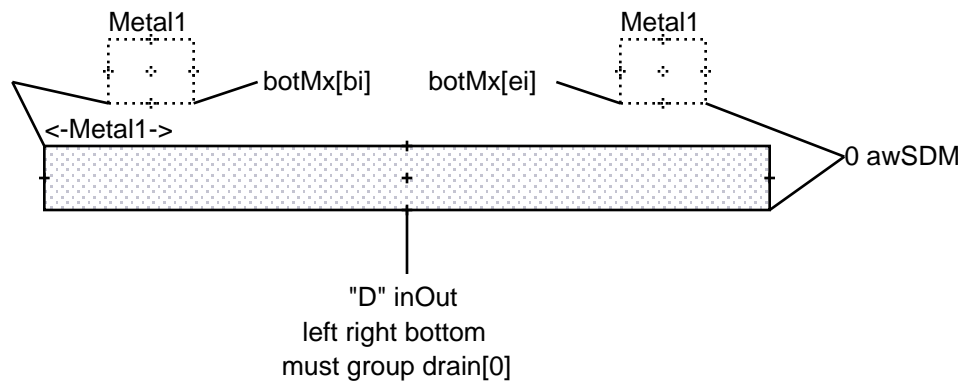
if connectD

connectD

1: include

```
bi = if( switchSD then 0 else 1 )
ei = if( PasXor( evenp( gc ) switchSD ) then gc - 1 else gc )
```

2: include



pcell macro mos

17: include

polyExt

```

connectTG = member( connectGates list( "Top" "Both" ) )
connectBG = member( connectGates list( "Bottom" "Both" ) )
alternateGC = equal( connectGates "Alternate" )

topExt = 0      botExt = 0      topPinExt = 0      botPinExt = 0
tmp = if( connectTG || alternateGC then
  topExt = msM1 + meM1Cont - mePolyCont + meeM1Cont - meeDiffCont
  when( connectS topExt = topExt + msM1 + awSDM )
  topExt = max(msDiffPoly topExt) + thStep
else
  if( connectBG || alternateGC then topExt = mxGate
  else topExt = msDiffPoly + thStep topPinExt = max(mxGate - topExt grid) )
)

tmp = if( connectBG || alternateGC then
  botExt = msM1 + meM1Cont - mePolyCont + meeM1Cont - meeDiffCont
  when( connectD botExt = botExt + msM1 + awSDM )
  botExt = max(msDiffPoly botExt) + bhStep
else
  if( connectTG || alternateGC then botExt = mxGate
  else botExt = msDiffPoly + bhStep botPinExt = max(mxGate - botExt grid) )
)

```

2: include

```

polyEnc = PasSizeSpacing( nil ?top ( topExt + topPinExt - thStep )
                          ?bottom ( botExt + botPinExt - bhStep ) )

```

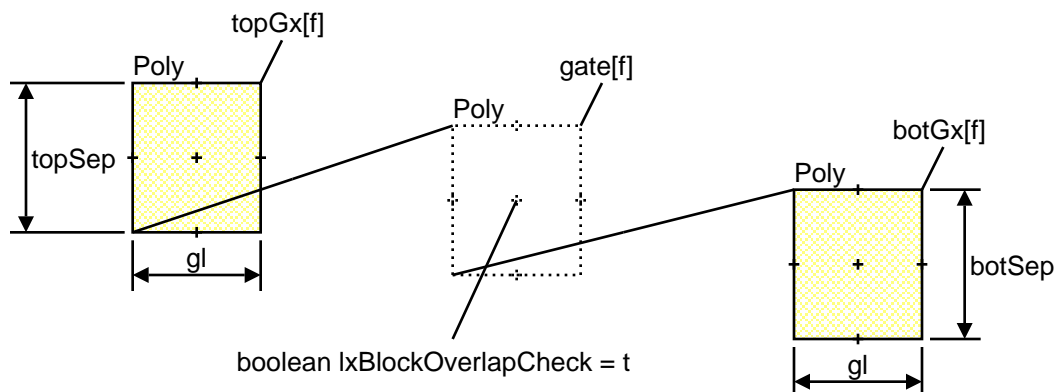
3: for f 1 gc

polyExtLoop

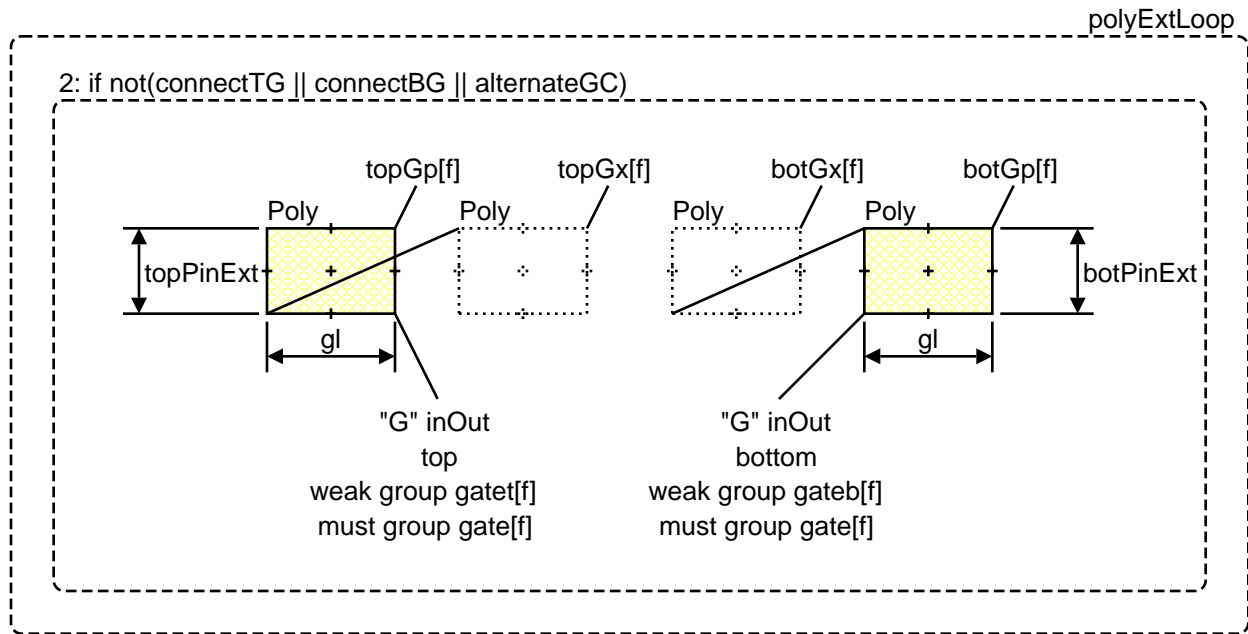
```

topSep = if( alternateGC && f == gc && oddp(f) then mxGate else topExt )
botSep = if( alternateGC && ( f == 1 || ( f == gc && evenp(f) ) ) then mxGate else botExt )

```



pcell macro mos



connectG

18: if ( connectTG || connectBG || alternateGC)

```

awPoly = mwCont + ( 2.0 * mePolyCont )
mIPoly = mwCont + ( 2.0 * meePolyCont )
leftSep = 0          rightSep = 0
tmp = when( gc == 1 && mIPoly > gl
  leftSep = PasCeiling( ( ( mIPoly - gl ) / 2.0 ) grid t )
  rightSep = mIPoly - gl - leftSep
)
polyEnc = PasSizeSpacing( polyEnc ?left leftSep ?right rightSep )
awM1 = mwCont + ( 2.0 * meM1Cont )
hSep = meePolyCont - meeM1Cont
vSep = mePolyCont - meM1Cont
ei = if( alternateGC then max(gc - 1 1) else 1 )

2: if connectTG || alternateGC
  polyEnc = PasSizeSpacing( polyEnc ?top awPoly )

2: if connectBG || ( alternateGC && gc > 2 )
  polyEnc = PasSizeSpacing( polyEnc ?bottom awPoly )

3: for f 1 ei
  index = if( alternateGC then min(f + 1 gc) else gc )

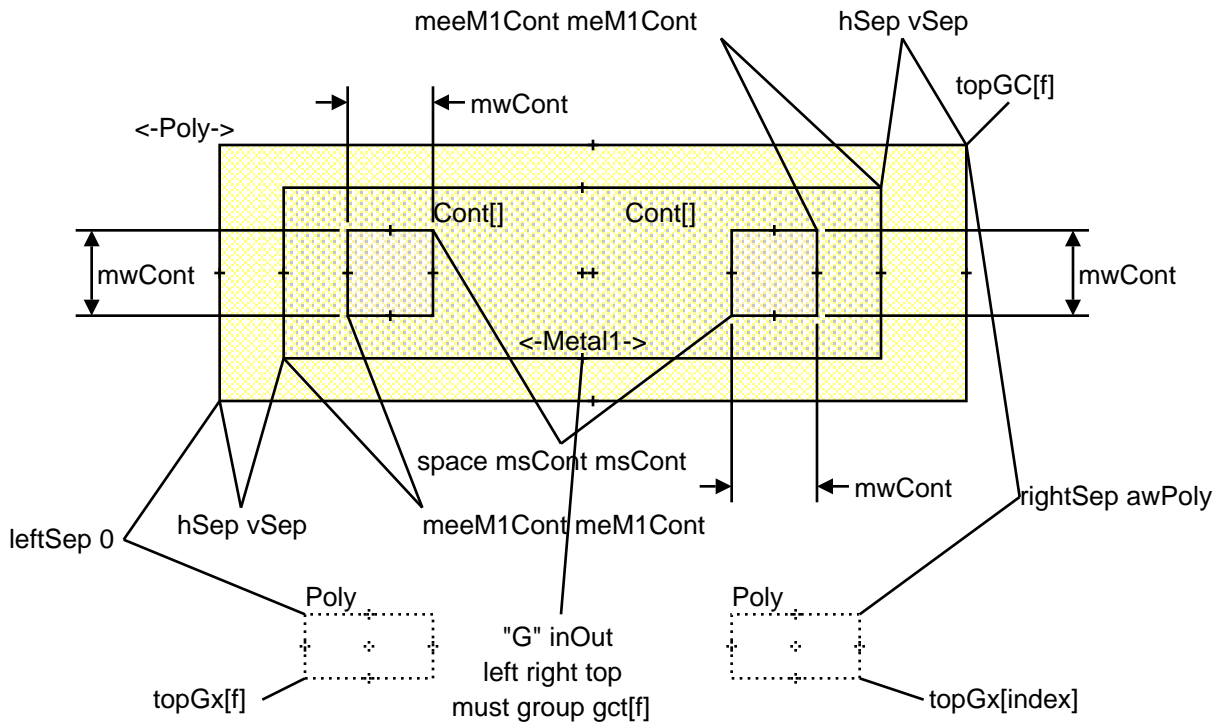
```

connectGLoop

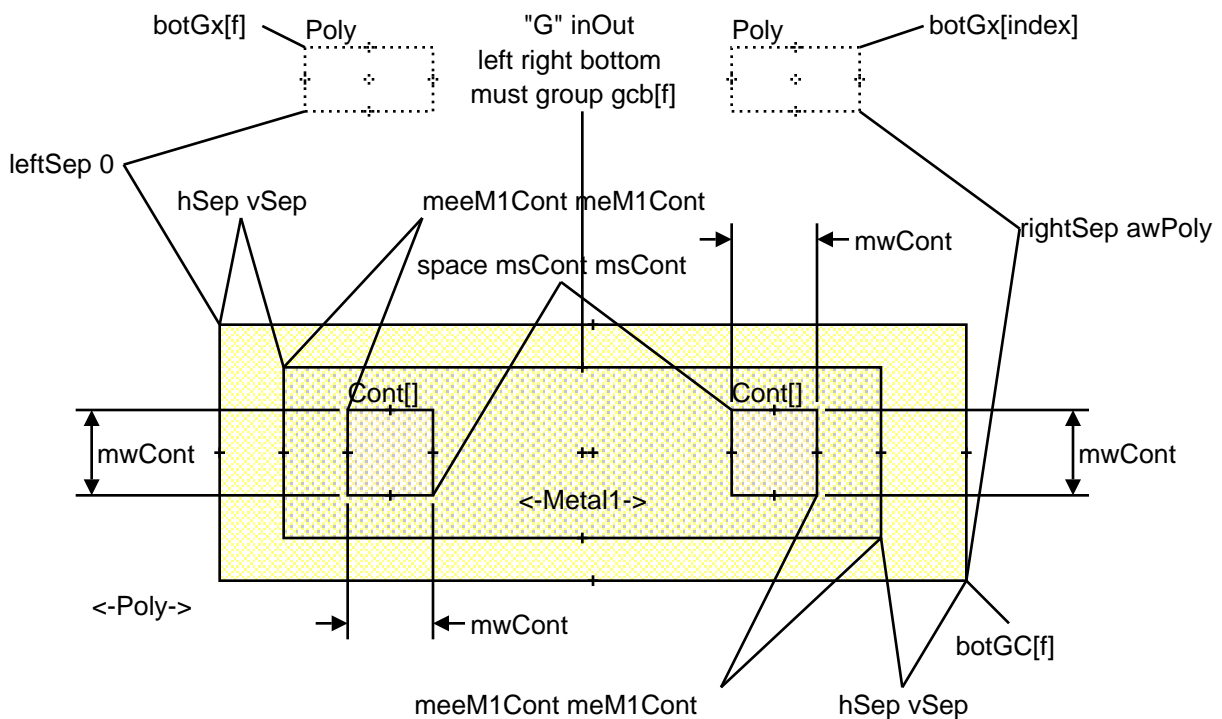
pcell macro mos

connectGLoop

2: if connectTG || (alternateGC && oddp(f))



2: if connectBG || (alternateGC && evenp(f))





# **mos\_nplus (macro)    mos\_pplus (macro)**

pcell macro mos\_implant

Pcell Macro Table

\$cell	\$pcLayer1
mos_nplus	Nimp
mos_pplus	Pimp

2: include

```
meImplDiff = dpt->meImplDiff
meImplPoly = dpt->meImplPoly
meImplGate = dpt->meImplGate
meImplGate = when( meImplGate dpt->meImplGate || meImplGate )
meImplCont = dpt->meImplCont
```

20: include

1: include

preimplant

1: include

```
implEnc = PasSizeSpacing( nil ?all meImplDiff )
botSep = meImplDiff + lhStep    topSep = meImplDiff + rhStep

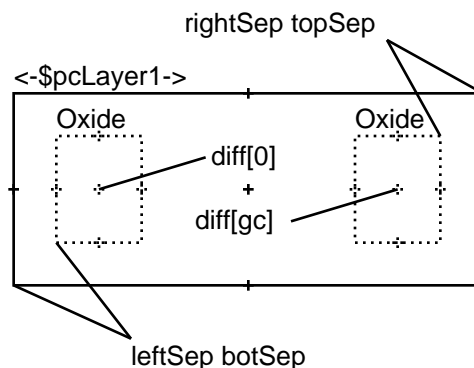
leftSep = cond(                      rightSep = cond(
  ( abs( leftAbut ) < 3 meImplDiff )    ( abs( rightAbut ) < 3 meImplDiff )
  ( abs( leftAbut ) > 5 -meImplDiff )    ( abs( rightAbut ) > 5 -meImplDiff )
  ( t 0 ) )                              ( t 0 ) )
```

3: include

```
test = if( onep(gc) then lwDiff + lwStep + leftSep + gl + rwDiff + rwStep + rightSep > epsilon else t )
```

2: if test

implant



pcell macro mos\_implant

implant

2: if meImplGate || meImplPoly

implantGate

```
topSep = if( meImplPoly then polyEnc->top + meImplPoly + thStep
            else meImplGate )
```

```
botSep = if( meImplPoly then polyEnc->bottom + meImplPoly + bhStep
            else meImplGate )
```

```
implEnc = PasConstrainSpacing( implEnc ?top ( topSep - thStep )
                               ?bottom ( botSep - bhStep ) )
```

```
leftSep = cond(
  ( leftAbut ==0 polyEnc->left + ( meImplPoly || meImplGate ) )
  ( abs( leftAbut ) < 3 lwDiff + meImplDiff )
  ( abs( leftAbut ) > 5 lwDiff - meImplDiff )
  ( t lwDiff ) )
```

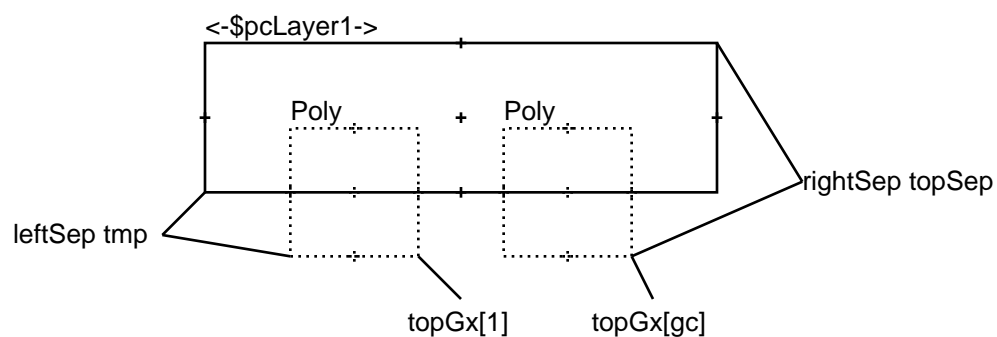
```
rightSep = cond(
  ( rightAbut ==0 polyEnc->right + ( meImplPoly || meImplGate ) )
  ( abs( rightAbut ) < 3 rwDiff + meImplDiff )
  ( abs( rightAbut ) > 5 rwDiff - meImplDiff )
  ( t rwDiff ) )
```

pcell macro mos\_implant

implantGate

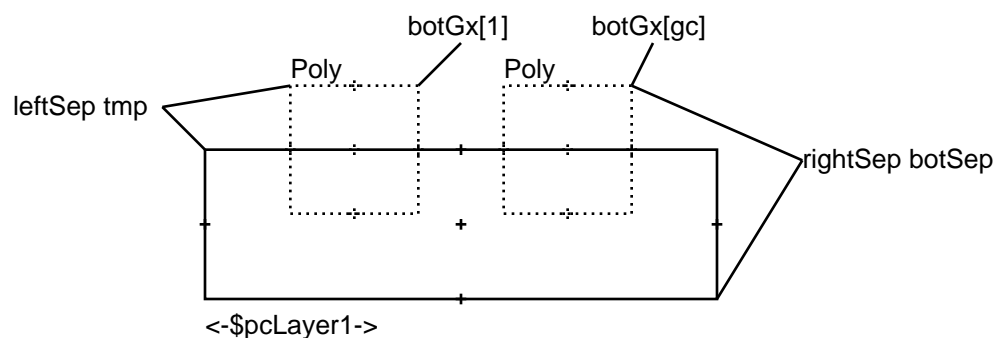
2: include

$tmp = meImplDiff + thStep$



3: include

$tmp = meImplDiff + bhStep$



**mos\_rf (macro)**

pcell macro mos\_rf

2: include

```
meRfDiff = dpt->meVoltDiff
meRfImpl = dpt->meVoltImpl
```

22: include

mos\_rf

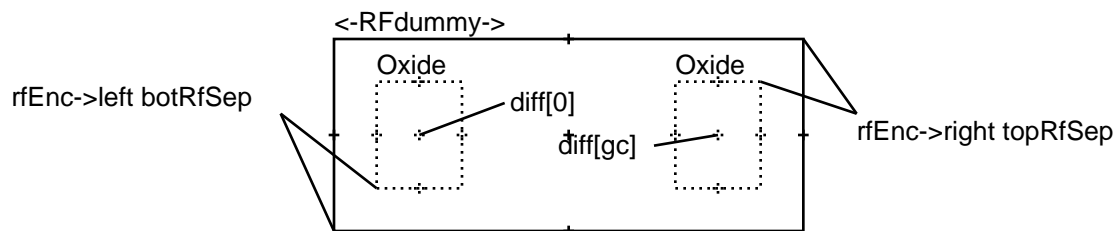
```
rfEnc = PasSizeSpacing( nil ?all meVoltDiff )
```

1: if meRfImpl

```
rfEnc = PasConstrainSpacing( rfEnc ?top ( implEnc->top + meRfImpl )
                             ?bottom ( implEnc->bottom + meRfImpl )
                             ?left ( implEnc->left + meRfImpl )
                             ?right ( implEnc->right + meRfImpl ) )
```

2: include

```
botRfSep = rfEnc->bottom + lhStep
topRfSep = rfEnc->top + rhStep
```



## mos\_volt (macro)

pcell macro mos\_volt

2: include

```
meVoltDiff = dpt->meVoltDiff
meVoltImpl = dpt->meVoltImpl
```

22: include

mos\_volt

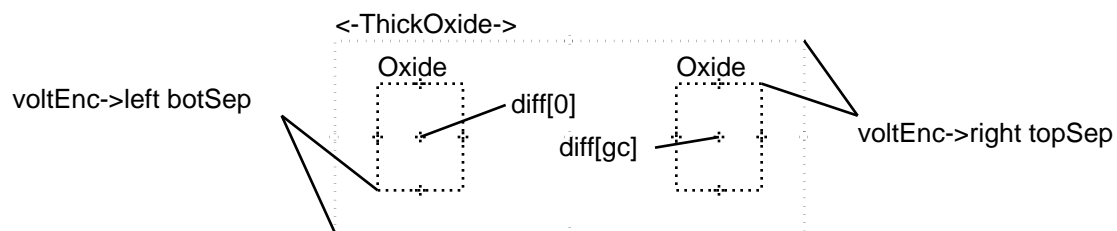
```
voltEnc = PasSizeSpacing( nil ?all meVoltDiff )
```

1: if meVoltImpl

```
voltEnc = PasConstrainSpacing( voltEnc ?top ( implEnc->top + meVoltImpl )
                                ?bottom ( implEnc->bottom + meVoltImpl )
                                ?left ( implEnc->left + meVoltImpl )
                                ?right ( implEnc->right + meVoltImpl ) )
```

2: include

```
botSep = voltEnc->bottom + lhStep
topSep = voltEnc->top + rhStep
```



## mos\_volt\_tap (macro)

pcell macro mos\_volt\_tap

include macro mos\_volt

2: include

msTapVolt = dpt->msTapVolt

mos\_volt

3: include

tapSp = PasConstrainSpacing( tapSp ?top ( voltEnc->top + msTapVolt )  
?bottom ( voltEnc->bottom + msTapVolt)  
?left ( voltEnc->left + msTapVolt)  
?right ( voltEnc->right + msTapVolt) )

## mos\_nw (macro) mos\_pw (macro)

pcell macro mos\_well

Pcell Macro Table

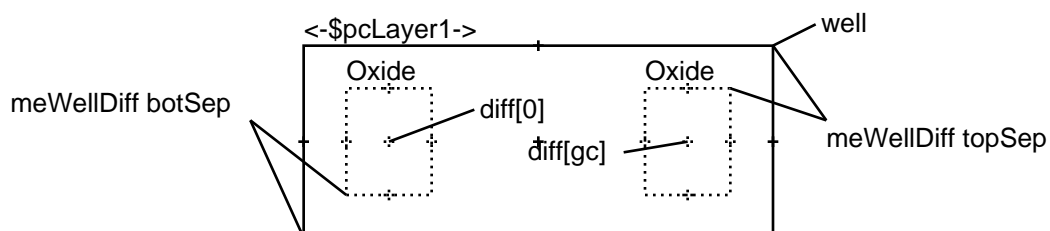
\$cell	\$pcLayer1
mos_nw	Nwell
mos_pw	Pwell

2: include

meWellDiff = dpt->meWellDiff

40: include

botSep = meWellDiff + lhStep  
topSep = meWellDiff + rhStep



# mos\_nw\_tap (macro) mos\_nbl\_tap (macro) mos\_pw\_tap (macro) mos\_cap\_tap (macro)

pcell macro mos\_well\_tap

Pcell Macro Table

\$cell	\$pcLayer1	\$diffEnc	\$tapEnc	\$name
mos_nw_tap	Nwell	meWellDiff	meWellTap	well
mos_pw_tap	Pwell	meWellDiff	meWellTap	well
mos_nbl_tap	Nburied	meNblDiff	meNblTap	nbl
mos_cap_tap	Capdum	meCapDiff	meCapTap	cap

2: include

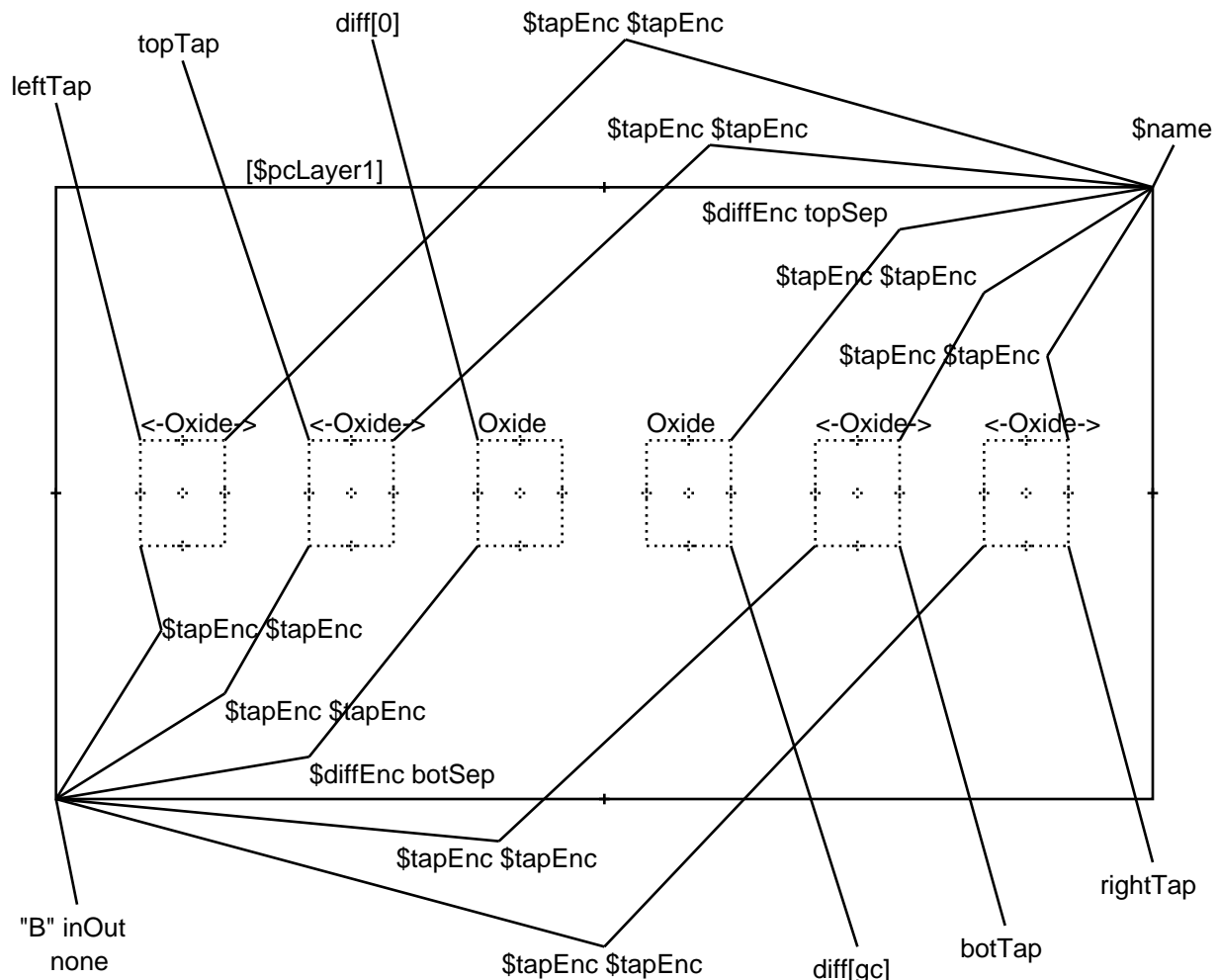
\$diffEnc = dpt->\$diffEnc

\$tapEnc = dpt->\$tapEnc

40: include

botSep = \$diffEnc + lhStep

topSep = \$diffEnc + rhStep



# mosltap (macro)

pcell macro mosltap

Pcell Macro Table

\$cell	\$pcLayer1
mosltap_nplus	Nimp
mosltap_pplus	Pimp

2: include

```

mwltap = dpt->mwltap || 0
msltapGate = dpt->msltapGate
maltimp = dpt->maltimp
meltimpCont = dpt->meltimpCont
meltimpltap = dpt->meltimpltap
leftltap = tap == "Integrated" && leftTap && leftAbut == 0
rightltap = tap == "Integrated" && rightTap && rightAbut == 0
    
```

7: if leftltap || rightltap

```

awltap = max( mwltap ( meDiffCont + mwCont + max( meltimpCont ( msCont - meImplCont ) ) ) )
    
```

if leftltap

```

lwDiff = max( ( msltapGate - lwStep ) ( laDC + cntSpan + meImplCont ) ) + awltap
lwSDM = lwDiff - laDC - meDiffCont + meM1Cont + meM1Cont
    
```

if rightltap

```

rwDiff = max( ( msltapGate - rwStep ) ( raDC + cntSpan + meImplCont ) ) + awltap
rwSDM = rwDiff - raDC - meDiffCont + meM1Cont + meM1Cont
    
```

preimplant

2: if leftltap

```

leftSep = -awltap
    
```

2: if rightltap

```

rightSep = -awltap
    
```



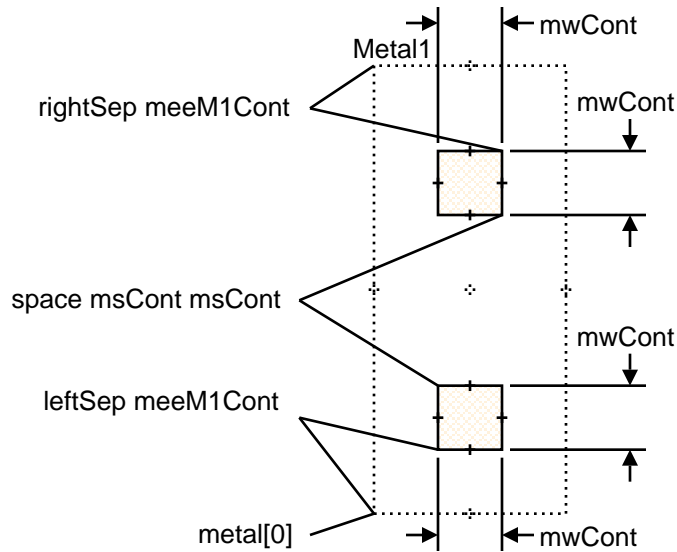
pcell macro mosltap

16: if leftltap || rightltap

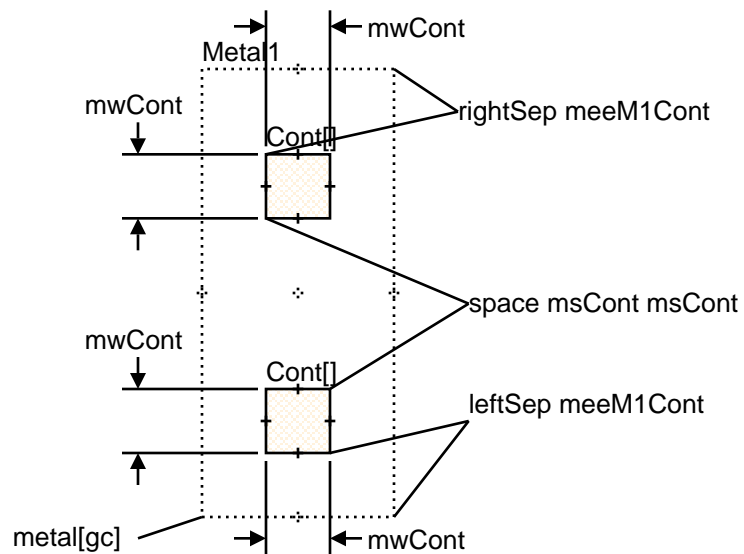
leftSep = meM1Cont

rightSep = meM1Cont + mwCont

if leftltap



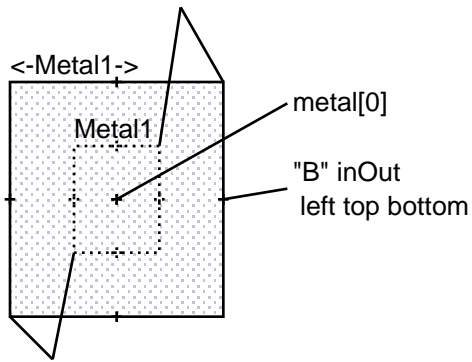
if rightltap



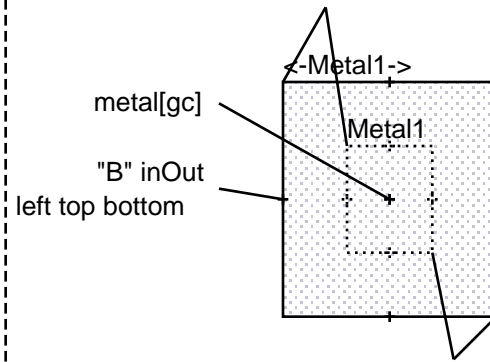
pcell macro mosltap

15: include

if leftltap



if rightltap



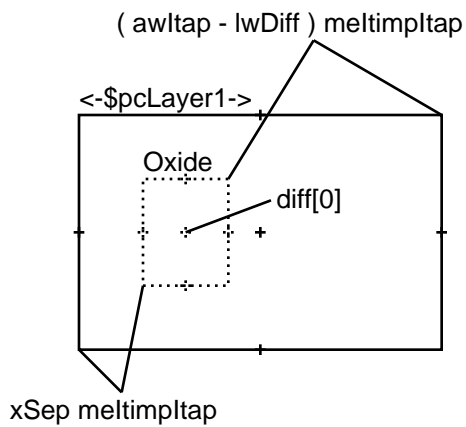
21: if leftltap || rightltap

$xSep = mltimptap$

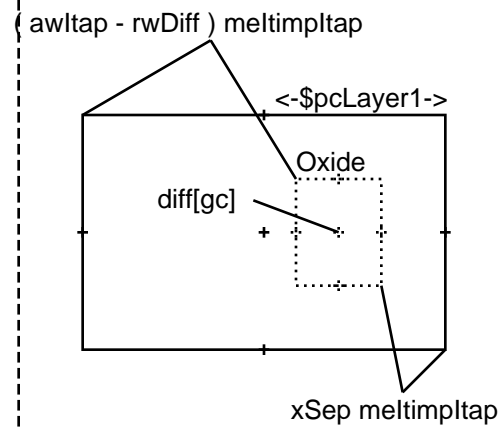
2: if maltimp

$xSep = \max( mltimptap \text{ PasCeiling}( ( maltimp / ( alDiff + ( 2 * mltimptap ) ) ) - awltap \text{ grid } t ) )$

3: if leftltap



4: if rightltap



## mosTap (macro)

pcell macro mosTap

```
deTap = tap == "Detached" && ( leftTap || rightTap || topTap || bottomTap )
```

1: if deTap

```
msTapDiff = dpt->msTapDiff  
meTimpTap = dpt->meTimpTap  
msTapImpl = dpt->msTapImpl || 0.0  
msTimpDiff = dpt->msTimpDiff || 0.0  
msTimpImpl = dpt->msTimpImpl || 0.0  
maTap = dpt->maTap
```

7: if deTap

```
awTap = ( max( 1 tapCntRows ) * ( mwCont + msCont ) ) - msCont + ( 2.0 * meDiffCont )  
tapSp = PasConstrainSpacing( nil ?all max( msTapDiff ( meTimpTap + msTimpDiff ) ) )
```

if topTap || bottomTap

```
tmp = msM1 + meM1Cont - meDiffCont + meeM1Cont - meeDiffCont  
tapSp = PasConstrainSpacing( tapSp ?top tmp ?bottom tmp )
```



pcell macro mosTap

31: if deTap &amp;&amp; ( topTap || bottomTap )

```

lxSep = if( leftTap then tapSp->left + awTap else 0.0)
rxSep = if( rightTap then tapSp->right + awTap else 0.0)

```

if topTap

```

lySep = tapSp->top + lhStep
rySep = tapSp->top + rhStep

```

param tapExtension

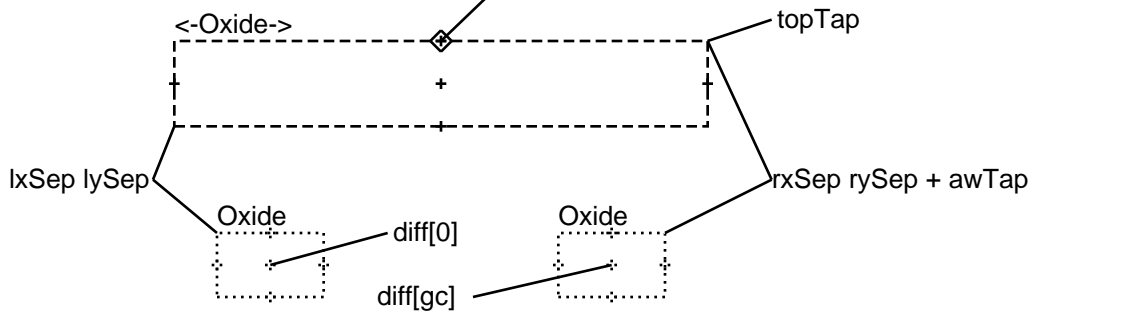
relative y

dispName Top Tap Separation

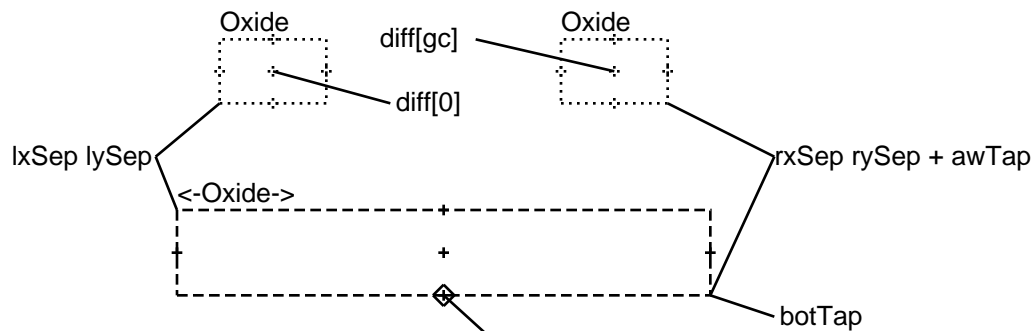
dispExpr tapExt-&gt;top

userData list( nil 'side 'top )

userFunc "gpd180\_mosTapUserFunc"



if bottomTap



```

lySep = tapSp->bottom + lhStep
rySep = tapSp->bottom + rhStep

```

param tapExtension

relative y

dispName Bottom Tap Separation

dispExpr tapExt-&gt;bottom

userData list( nil 'side 'bottom )

userFunc "gpd180\_mosTapUserFunc"

pcell macro mosTap

31: if deTap && ( leftTap || rightTap )

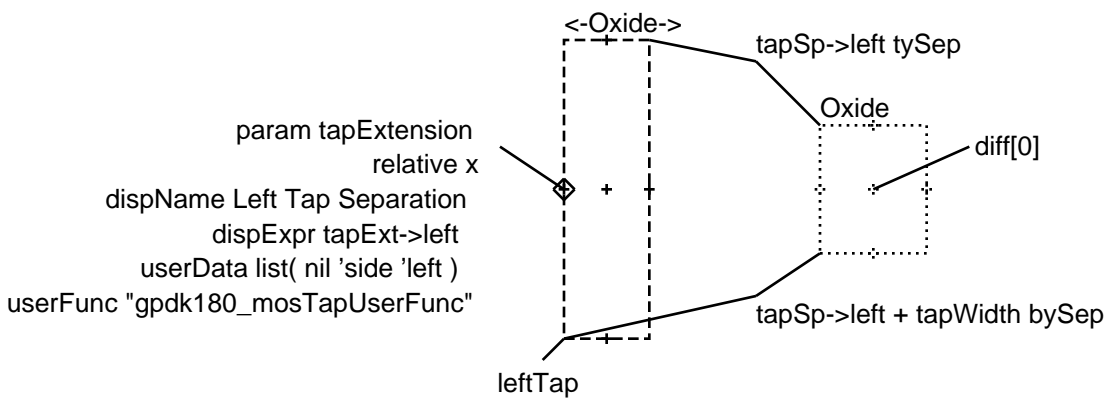
```
tySep = if( topTap then tapSp->top else 0.0 )
bySep = if( bottomTap then tapSp->bottom else 0.0 )
tapWidth = awTap
```

if maTap && not( topTap || bottomTap )

```
tapWidth = max( tapWidth PasCeiling( ( maTap / alDiff ) grid ) )
```

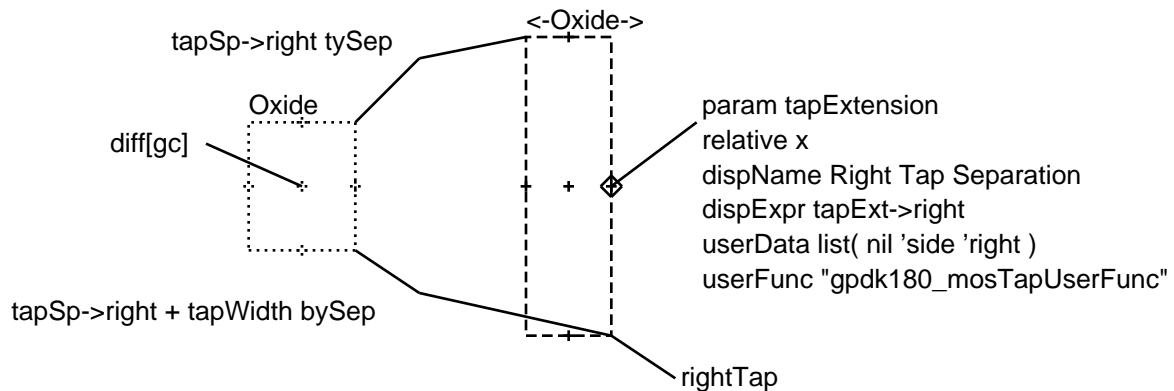
if leftTap

```
tySep = tySep + lhStep
bySep = bySep + lhStep
```



if rightTap

```
tySep = tySep + rhStep
bySep = bySep + rhStep
```



```
32: if deTap && ( topTap || bottomTap )
```

```
ySep = meDiffCont - meM1Cont
```

The diagram illustrates the cross-sectional structure of a 1T1R device. It shows a central channel region (dotted pattern) flanked by two contact regions (orange pattern). The top layer is labeled "meem1Cont" and the bottom layer is labeled "meem1Cont". The channel region is labeled "space msCont msCont". The contact regions are labeled "Cont[]". The device is bounded by "xSep" and "ySep" lines. The top contact region is labeled "botTap". The channel region is labeled "B" inOut all. The channel region is also labeled "<-Metal1->" and "<-Oxide->". The channel region is labeled "mwCont" and "xSep ySep". The channel region is labeled "mwCont" and "xSep ySep". The channel region is labeled "mwCont" and "xSep ySep".

pcell macro mosTap

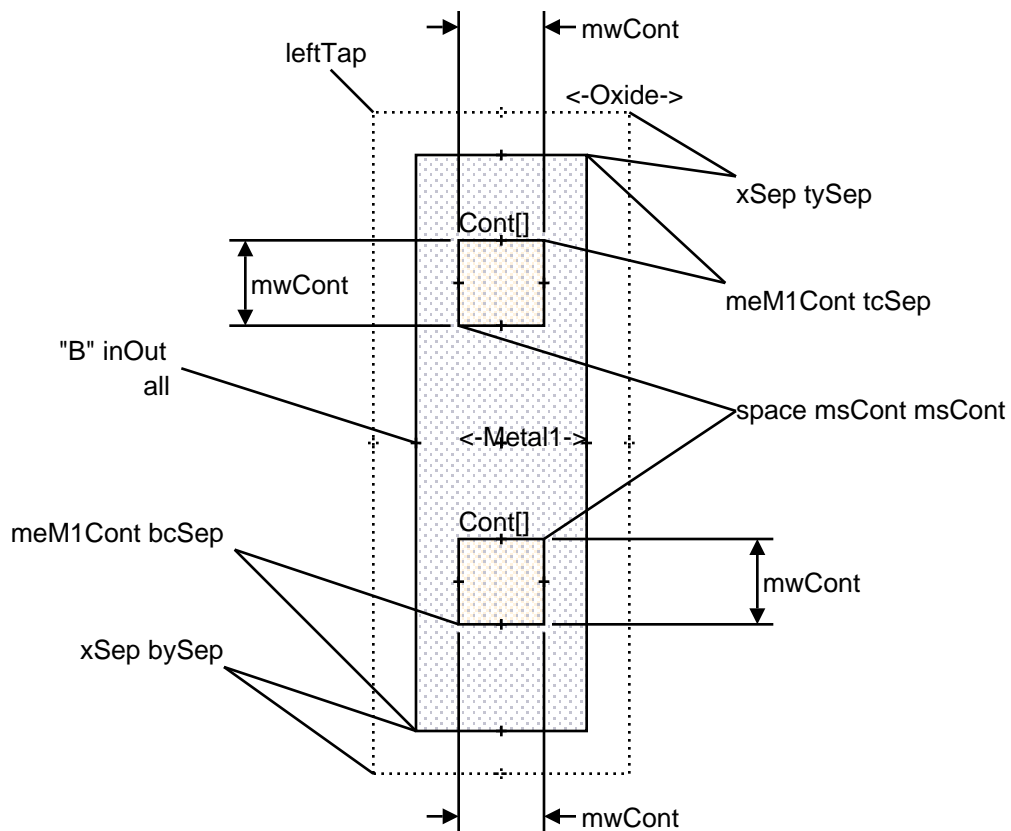
32: if deTap && ( leftTap || rightTap )

leftRightTap

```

xSep = meDiffCont - meM1Cont
ySep = meDiffCont - meeM1Cont
tySep = if( topTap then -xSep else ySep )
bySep = if( bottomTap then -xSep else ySep )
tcSep = if( topTap then msCont - meM1Cont else meeM1Cont )
bcSep = if( bottomTap then msCont - meM1Cont else meeM1Cont )
    
```

if leftTap

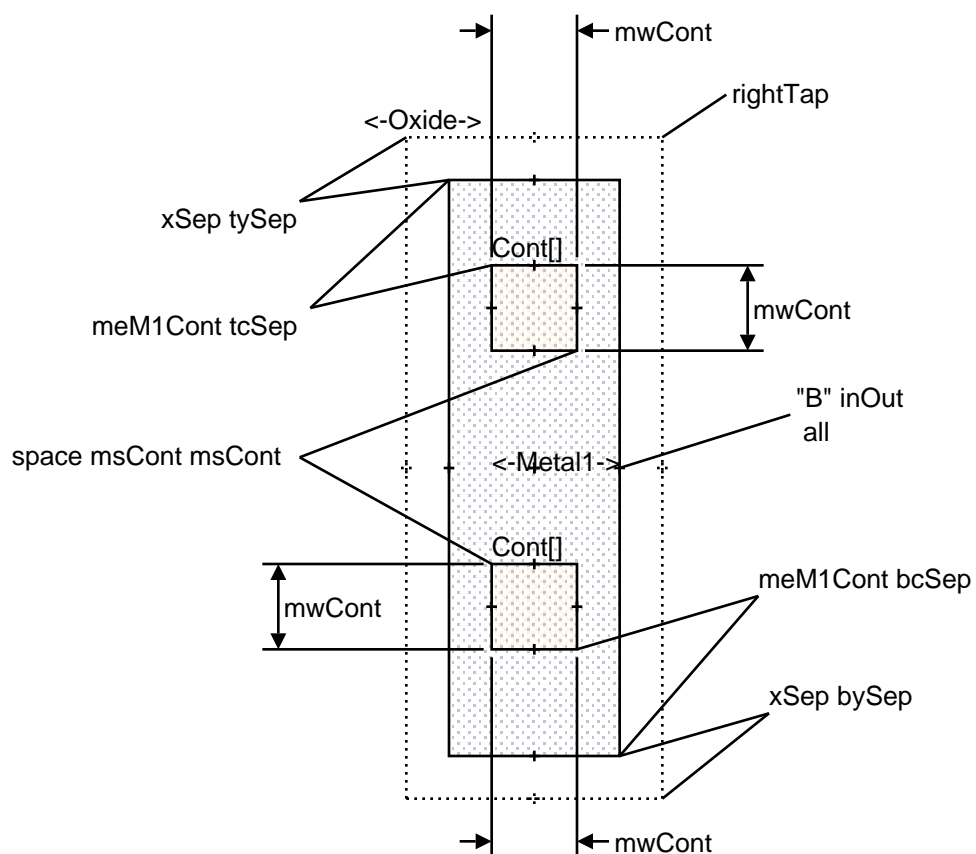




pcell macro mosTap

leftRightTap

if rightTap

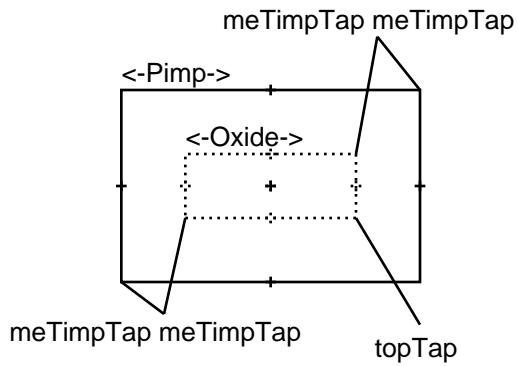


# mosTap\_pplus (macro)

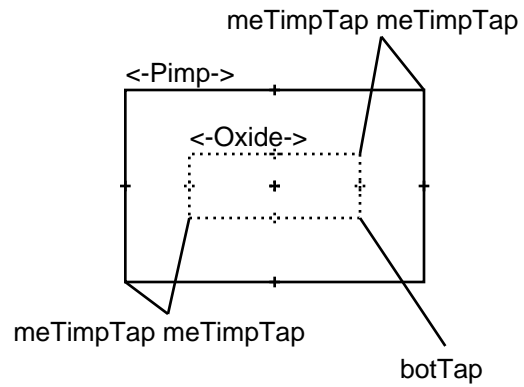
pcell macro mosTap\_pplus

33: if deTap && ( topTap || bottomTap )

if topTap



if bottomTap

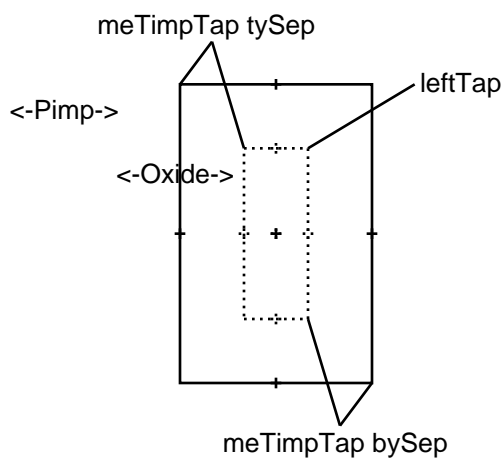


33: if deTap && ( leftTap || rightTap )

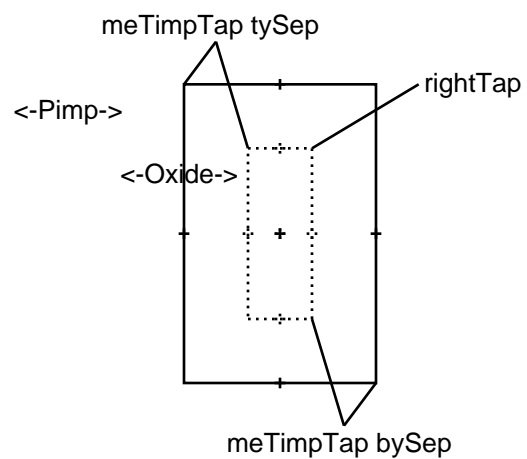
tySep = if( topTap then -meTimpTap else meTimpTap )

bySep = if( bottomTap then -meTimpTap else meTimpTap )

if leftTap



if rightTap

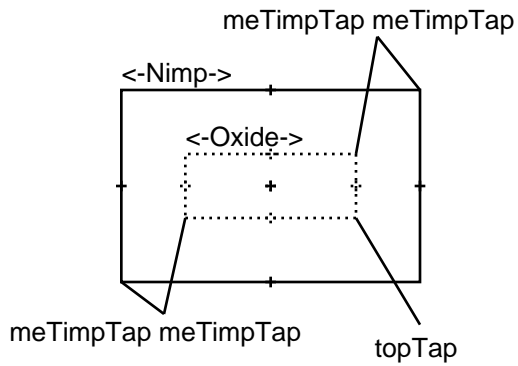


## mosTap\_nplus (macro)

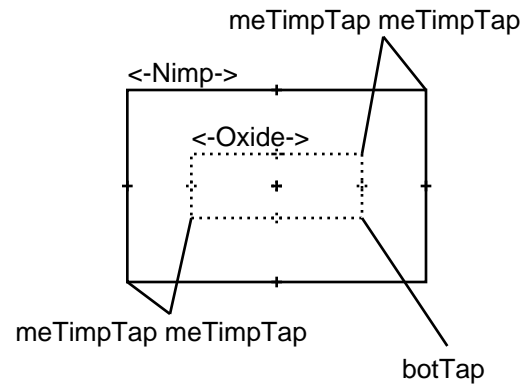
pcell macro mosTap\_nplus

33: if deTap && ( topTap || bottomTap )

if topTap



if bottomTap

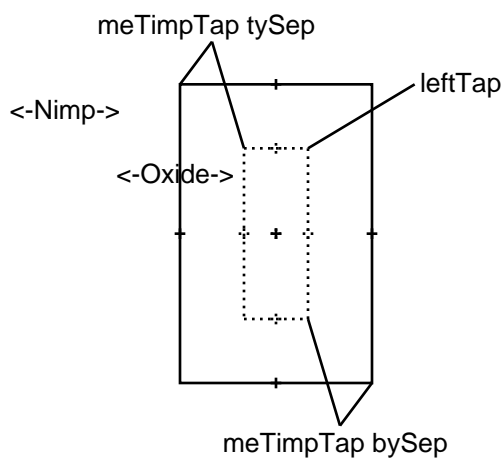


33: if deTap && ( leftTap || rightTap )

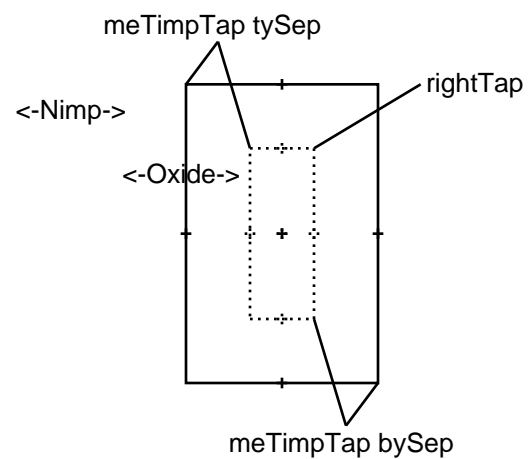
tySep = if( topTap then -meTimpTap else meTimpTap )

bySep = if( bottomTap then -meTimpTap else meTimpTap )

if leftTap



if rightTap



## PCell Devices

pcell gpdk180 nmos3

**nmos3**

PCell formal params

<b>fw</b>	float	CDF
<b>l</b>	float	CDF
<b>fingers</b>	float	CDF
<b>connectGates</b>	string	CDF
<b>connectSD</b>	string	CDF
<b>switchSD</b>	boolean	CDF
<b>mtlCvg</b>	string	CDF
<b>sdMtlWidth</b>	float	CDF
<b>leftAbut</b>	int	CDF
<b>rightAbut</b>	int	CDF

include macro mos

include macro mos\_nplus

pcell gpdk180 pmos3

**pmos3**

PCell formal params

<b>fw</b>	float	CDF
<b>l</b>	float	CDF
<b>fingers</b>	float	CDF
<b>connectGates</b>	string	CDF
<b>connectSD</b>	string	CDF
<b>switchSD</b>	boolean	CDF
<b>mtlCvg</b>	string	CDF
<b>sdMtlWidth</b>	float	CDF
<b>leftAbut</b>	int	CDF
<b>rightAbut</b>	int	CDF

include macro mos

include macro mos\_pplus

include macro mos\_nw

pcell gpdk180 nmos3hv

**nmos3hv**

PCell formal params

<b>fw</b>	float	CDF
<b>l</b>	float	CDF
<b>fingers</b>	float	CDF
<b>connectGates</b>	string	CDF
<b>connectSD</b>	string	CDF
<b>switchSD</b>	boolean	CDF
<b>mtlCvg</b>	string	CDF
<b>sdMtlWidth</b>	float	CDF
<b>leftAbut</b>	int	CDF
<b>rightAbut</b>	int	CDF

include macro mos

include macro mos\_nplus

include macro mos\_volt

pcell gpdk180 pmos3hv

**pmos3hv**

PCell formal params

<b>fw</b>	float	CDF
<b>l</b>	float	CDF
<b>fingers</b>	float	CDF
<b>connectGates</b>	string	CDF
<b>connectSD</b>	string	CDF
<b>switchSD</b>	boolean	CDF
<b>mtlCvg</b>	string	CDF
<b>sdMtlWidth</b>	float	CDF
<b>leftAbut</b>	int	CDF
<b>rightAbut</b>	int	CDF

include macro mos

include macro mos\_pplus

include macro mos\_nw

include macro mos\_volt

pcell gpd180 nmos

**nmos**

PCell formal params

<b>fw</b>	float	CDF
<b>l</b>	float	CDF
<b>fingers</b>	float	CDF
<b>connectGates</b>	string	CDF
<b>connectSD</b>	string	CDF
<b>switchSD</b>	boolean	CDF
<b>mtlCvg</b>	string	CDF
<b>sdMtlWidth</b>	float	CDF
<b>leftAbut</b>	int	CDF
<b>rightAbut</b>	int	CDF
<b>tap</b>	string	CDF
<b>topTap</b>	boolean	CDF
<b>bottomTap</b>	boolean	CDF
<b>leftTap</b>	boolean	CDF
<b>rightTap</b>	boolean	CDF
<b>tapExtension</b>	string	CDF
<b>tapCntRows</b>	int	CDF

include macro mos

include macro mos\_nplus

include macro mosltap\_pplus

include macro mosTap

include macro mosTap\_pplus

pcell gpd180 pmos

**pmos**

PCell formal params

<b>fw</b>	float	CDF
<b>l</b>	float	CDF
<b>fingers</b>	float	CDF
<b>connectGates</b>	string	CDF
<b>connectSD</b>	string	CDF
<b>switchSD</b>	boolean	CDF
<b>mtlCvg</b>	string	CDF
<b>sdMtlWidth</b>	float	CDF
<b>leftAbut</b>	int	CDF
<b>rightAbut</b>	int	CDF
<b>tap</b>	string	CDF
<b>topTap</b>	boolean	CDF
<b>bottomTap</b>	boolean	CDF
<b>leftTap</b>	boolean	CDF
<b>rightTap</b>	boolean	CDF
<b>tapExtension</b>	string	CDF
<b>tapCntRows</b>	int	CDF

include macro mos

include macro mos\_pplus

include macro mos\_nw\_tap

include macro mosltap\_nplus

include macro mosTap

include macro mosTap\_nplus

pcell gpd180 nmoshv

**nmoshv**

PCell formal params

<b>fw</b>	float	CDF
<b>l</b>	float	CDF
<b>fingers</b>	float	CDF
<b>connectGates</b>	string	CDF
<b>connectSD</b>	string	CDF
<b>switchSD</b>	boolean	CDF
<b>mtlCvg</b>	string	CDF
<b>sdMtlWidth</b>	float	CDF
<b>leftAbut</b>	int	CDF
<b>rightAbut</b>	int	CDF
<b>tap</b>	string	CDF
<b>topTap</b>	boolean	CDF
<b>bottomTap</b>	boolean	CDF
<b>leftTap</b>	boolean	CDF
<b>rightTap</b>	boolean	CDF
<b>tapExtension</b>	string	CDF
<b>tapCntRows</b>	int	CDF

include macro mos

include macro mos\_nplus

include macro mosltap\_pplus

include macro mosTap

include macro mosTap\_pplus

include macro mos\_volt\_tap

pcell gpd180 pmoshv

**pmoshv**

PCell formal params

<b>fw</b>	float	CDF
<b>l</b>	float	CDF
<b>fingers</b>	float	CDF
<b>connectGates</b>	string	CDF
<b>connectSD</b>	string	CDF
<b>switchSD</b>	boolean	CDF
<b>mtlCvg</b>	string	CDF
<b>sdMtlWidth</b>	float	CDF
<b>leftAbut</b>	int	CDF
<b>rightAbut</b>	int	CDF
<b>tap</b>	string	CDF
<b>topTap</b>	boolean	CDF
<b>bottomTap</b>	boolean	CDF
<b>leftTap</b>	boolean	CDF
<b>rightTap</b>	boolean	CDF
<b>tapExtension</b>	string	CDF
<b>tapCntRows</b>	int	CDF

include macro mos

include macro mos\_pplus

include macro mos\_nw\_tap

include macro mosltap\_nplus

include macro mosTap

include macro mosTap\_nplus

include macro mos\_volt\_tap

pcell gpd180 nmosrf

**nmosrf**

PCell formal params

<b>fw</b>	float	CDF
<b>l</b>	float	CDF
<b>fingers</b>	float	CDF
<b>connectGates</b>	string	CDF
<b>connectSD</b>	string	CDF
<b>switchSD</b>	boolean	CDF
<b>mtlCvg</b>	string	CDF
<b>sdMtlWidth</b>	float	CDF
<b>leftAbut</b>	int	CDF
<b>rightAbut</b>	int	CDF
<b>tap</b>	string	CDF
<b>topTap</b>	boolean	CDF
<b>bottomTap</b>	boolean	CDF
<b>leftTap</b>	boolean	CDF
<b>rightTap</b>	boolean	CDF
<b>tapExtension</b>	string	CDF
<b>tapCntRows</b>	int	CDF

```
include macro mos
include macro mos_nplus
include macro mos_rf
include macro mosltap_pplus
include macro mosTap
include macro mosTap_pplus
include macro mos_volt_tap
```

pcell gpd180 pmosrf

**pmosrf**

PCell formal params

<b>fw</b>	float	CDF
<b>l</b>	float	CDF
<b>fingers</b>	float	CDF
<b>connectGates</b>	string	CDF
<b>connectSD</b>	string	CDF
<b>switchSD</b>	boolean	CDF
<b>mtlCvg</b>	string	CDF
<b>sdMtlWidth</b>	float	CDF
<b>leftAbut</b>	int	CDF
<b>rightAbut</b>	int	CDF
<b>tap</b>	string	CDF
<b>topTap</b>	boolean	CDF
<b>bottomTap</b>	boolean	CDF
<b>leftTap</b>	boolean	CDF
<b>rightTap</b>	boolean	CDF
<b>tapExtension</b>	string	CDF
<b>tapCntRows</b>	int	CDF

```
include macro mos
include macro mos_pplus
include macro mos_rf
include macro mos_nw_tap
include macro mosltap_nplus
include macro mosTap
include macro mosTap_nplus
include macro mos_volt_tap
```

## Library RESISTOR Definitions



## Global Resistor Parameters

Global Parameters

<b>nplus_rho</b>	50	Sheet Rho for nplusres	
<b>pplus_rho</b>	158	Sheet Rho for pplusres	
<b>poly_rho</b>	7.5	Sheet Rho for polyres	
<b>polyh_rho</b>	352	Sheet Rho for polyhres	
<b>nwell_rho</b>	415	Sheet Rho for nwellres	
<b>m1_rho</b>	0.01	Sheet Rho for nwellres	
<b>m2_rho</b>	0.01	Sheet Rho for nwellres	
<b>m3_rho</b>	0.01	Sheet Rho for nwellres	
<b>m4_rho</b>	0.01	Sheet Rho for nwellres	
<b>m5_rho</b>	0.01	Sheet Rho for nwellres	
<b>m6_rho</b>	0.01	Sheet Rho for nwellres	

## Poly Resistor

### Global Parameters

<b>rpoly_mwDevice</b>	0.6	Minimum poly resistor width	
<b>rpoly_mlDevice</b>	0.6	Minimum poly resistor length	
<b>rpoly_mnSquares</b>	1	Minimum number of squares	
<b>rpoly_mwRes</b>	{5A}	Minimum poly width	
<b>rpoly_msRes</b>	{5B}	Minimum poly spacing	
<b>rpoly_mwCont</b>	{6A}	Minimum & maximum contact width	
<b>rpoly_msCont</b>	{6B}	Minimum contact spacing	
<b>rpoly_meResCont</b>	{6D}	Minimum poly enclosure of contact	
<b>rpoly_meeResCont</b>	{6D}	Minimum poly end enclosure of contact	
<b>rpoly_mwM1</b>	{7A}	Minimum metal 1 width	
<b>rpoly_msM1</b>	{7B}	Minimum metal 1 spacing	
<b>rpoly_meM1Cont</b>	{7C}	Minimum metal 1 enclosure of contact	
<b>rpoly_meeM1Cont</b>	{7C}	Minimum metal 1 end enclosure of contact	

### Global Parameters

<b>rnpoly_melmplRes</b>	0.0	Minimum n+ implant enclosure of resistor	
<b>rppoly_melmplRes</b>	0.0	Minimum p+ implant enclosure of resistor	
<b>rpoly_meldRes</b>	0.05	Minimum recognition layer enclosure of resistor body	
<b>rpoly_meSbRes</b>	0.0	Minimum Sb enclosure of resistor	
<b>rpoly_msSbCont</b>	0.0	Minimum Sb to contact spacing	
<b>rpoly_msSbRes</b>	0.0	Minimum Sb to poly spacing	
<b>rppoly_meWellRes</b>	0.0	Minimum well enclosure of resistor	

## Diffusion Resistor

### Global Parameters

<b>rdiff_mwDevice</b>	0.6	Minimum diff resistor width	
<b>rdiff_mlDevice</b>	0.6	Minimum diff resistor length	
<b>rdiff_mnSquares</b>	1	Minimum number of squares	
<b>rdiff_mwRes</b>	{2A}	Minimum diffusion width	
<b>rdiff_msRes</b>	{2B}	Minimum diffusion spacing	
<b>rdiff_mwCont</b>	{6A}	Minimum & maximum contact width	
<b>rdiff_msCont</b>	{6B}	Minimum contact spacing	
<b>rdiff_meResCont</b>	{6C}	Minimum diff enclosure of contact	
<b>rdiff_meeResCont</b>	{6C}	Minimum diff end enclosure of contact	
<b>rdiff_mwM1</b>	{7A}	Minimum metal 1 width	
<b>rdiff_msM1</b>	{7B}	Minimum metal 1 spacing	
<b>rdiff_meM1Cont</b>	{7C}	Minimum metal 1 enclosure of contact	
<b>rdiff_meeM1Cont</b>	{7C}	Minimum metal 1 end enclosure of contact	

### Global Parameters

<b>rndiff_melmplRes</b>	{3C}	Minimum n+ implant enclosure of resistor	
<b>rpdiff_melmplRes</b>	{4C}	Minimum p+ implant enclosure of resistor	
<b>rdiff_meldRes</b>	0.05	Minimum recognition layer enclosure of resistor body	
<b>rdiff_meSbRes</b>	0.0	Minimum Sb enclosure of resistor	
<b>rdiff_msSbCont</b>	0.0	Minimum Sb to contact spacing	
<b>rdiff_msSbRes</b>	0.0	Minimum Sb to diff spacing	
<b>rpdiff_meWellRes</b>	{2C}	Minium well enclosure of resistor	

## NWell Resistor

### Global Parameters

<b>rnw_mwDevice</b>	1.6	Minimum diff resistor width	
<b>rnw_mlDevice</b>	2.0	Minimum diff resistor length	
<b>rnw_mnSquares</b>	1	Minimum number of squares	
<b>rnw_mwRes</b>	{1A}	Minimum nwell width	
<b>rnw_msRes</b>	{1B}	Minimum nwell spacing	
<b>rnw_mwDiff</b>	{2A}	Minimum diffusion width	
<b>rnw_msDiff</b>	{2B}	Minimum diffusion spacing	
<b>rnw_meResDiff</b>	{2C}	Minimum nwell enclosure of diffusion	
<b>rnw_melmplDiff</b>	{3C}	Minimum implant enclosure of diffusion	
<b>rnw_mwCont</b>	{6A}	Minimum & maximum contact width	
<b>rnw_msCont</b>	{6B}	Minimum contact spacing	
<b>rnw_meDiffCont</b>	{6C}	Minimum diff enclosure of contact	
<b>rnw_meeDiffCont</b>	{6C}	Minimum diff end enclosure of contact	
<b>rnw_mwM1</b>	{7A}	Minimum metal 1 width	
<b>rnw_msM1</b>	{7B}	Minimum metal 1 spacing	
<b>rnw_meM1Cont</b>	{7C}	Minimum metal 1 enclosure of contact	
<b>rnw_meeM1Cont</b>	{7C}	Minimum metal 1 end enclosure of contact	

### Global Parameters

<b>rnw_meldRes</b>	0.05	Minimum recognition layer enclosure of resistor body	
--------------------	------	--	--

## Callbacks

### Resistor Callback Procedures

#### CDF Callback

```

procedure(gpdk180_resCB(param "s")                                ;_Feb 21 03 amir 606
  let(
    (
      cdfId cellName libName procName paramId
      dpt grid scale sGrid epsilon doList
      mode min max commit
      r w sl ssegs psegs cntRows calcDpl
    )

    ;; set local variable to global cdfgData
    cdfId = cdfgData

    caseq( concat(cdfId~>type)
      ( ( cellData baseCellData userCellData )
        cellName = cdfId~>id~>name
        libName = cdfId~>id~>lib~>name
      )
      ( instData
        cellName = cdfId~>id~>master~>cellName
        libName = cdfId~>id~>master~>libName
      )
      ( t
        PasError("gpdk180_resCB: unexpected CDF data type - %L!" cdfId~>type)
        error("\tCallback aborted abnormally!")
      )
    ) ; ** caseq concat **

    procName = sprintf(nil "%s/%s callback" libName cellName)

    unless( paramId = get(cdfId param)
      PasError("%s: parameter not found in CDF - %s" procName param)
      error("\tCallback aborted abnormally!")
    )

    dpt = PasGetPdkParams(libName cellName)
    grid = dpt->grid
    scale = dpt->scale || 1e-6
    sGrid = grid * scale
    epsilon = sGrid / 10.0
    doList = tconc(nil nil)
    mode = 'extreme
  ...

```

## Resistor Spectre netlist procedure

### CDF Callback

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; File:          ptag_resNetlistProc.il
;
; Description:   Custom resistor netlist routine to handle series and
;               parallel resistor configuration correctly.
;               This routine expects the following parameter
;               to be defined on the resistor CDF.
;
;               segments - int
;               connection - cyclic field ("Parallel" && "Series")
;               sl - string
;               l - string
;               w - string
;
;               It will create a resistor instantiation for each
;               series or parallel resistor segment.
;               The series configuration is a little more trickier
;               since it has to create some dummy nets on the
;               fly. It will check to make sure those nets do
;               not conflict with any existing nets.
;
;               Example of a series configuration:
;               One single resistor in schematic:
;               connection = "Series"
;               sl = 83.335u
;               w = 2u
;               segments = 4
;               model = rnpoly
;
;               Resulting Spectre Netlist:
;
;               //Series configuration of R0
;               R0_1__dmy0  a R0_1__dmy486 rnpoly l=83.335u w=2u dw=0.0 m=1
;               R0_2__dmy0  R0_1__dmy486 R0_2__dmy486 rnpoly l=83.335u w=2u dw=0.0 m=1
;               R0_3__dmy0  R0_2__dmy486 R0_3__dmy486 rnpoly l=83.335u w=2u dw=0.0 m=1
;               R0_4__dmy0  R0_3__dmy486 x_1 rnpoly l=83.335u w=2u dw=0.0 m=1
;               //End of R0
;
;               Example of a parallel configuration:
;               One single resistor in schematic:
;               connection = "Parallel"
;               sl = 33.335u
;               w = 2u
;               segments = 2
;               model = rnpoly
;
;               Resulting Spectre Netlist:
;               // Parallel configuration of R2...

```

## CDF Macro Definitions

### resParamsA - Resistor CDF Parameters (Part A)

CDF macro resParamsA

CDF parameters

<b>name</b>	"resType"
<b>prompt</b>	"Resistor type"
<b>defValue</b>	"\$cell"
<b>type</b>	"string"
<b>display</b>	"nil"
<b>editable</b>	"nil"

CDF parameters

<b>name</b>	"calculatedParam"
<b>prompt</b>	"Calculated Parameter"
<b>defValue</b>	"Length"
<b>choices</b>	'("Resistance" "Length")
<b>type</b>	"radio"
<b>display</b>	"t"
<b>callback</b>	"gpdK180_resCB( 'calculatedParam )"

CDF parameters

<b>name</b>	"model"
<b>prompt</b>	"Model Name"
<b>defValue</b>	gpdK180_resValue( 'model ?id cdfId ?returnString t )
<b>type</b>	"string"
<b>display</b>	"gpdK180_resDisplay( 'model)"
<b>editable</b>	"nil"
<b>parseAsCEL</b>	"yes"

CDF parameters

<b>name</b>	"r"
<b>prompt</b>	"Total Resistance"
<b>units</b>	"resistance"
<b>defValue</b>	gpdK180_resValue( 'r ?id cdfId ?returnString t )
<b>type</b>	"string"
<b>display</b>	"t"
<b>editable</b>	"nequal(cdfgData->calculatedParam->value \"Resistance\")"
<b>callback</b>	"gpdK180_resCB( 'r )"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF macro resParamsA

CDF parameters

<b>name</b>	"w"
<b>prompt</b>	"Width"
<b>units</b>	"lengthMetric"
<b>defValue</b>	gpdk180_resValue( 'w ?id cdfld ?returnString t )
<b>type</b>	"string"
<b>display</b>	"t"
<b>editable</b>	"nequal(cdfgData->calculatedParam->value \"Width\")"
<b>callback</b>	"gpdk180_resCB( 'w )"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"



## resParamsB\_uniSeg - Resistor CDF Parameters (Part B)

CDF macro resParamsB\_uniSeg

CDF parameters

<b>name</b>	"l"
<b>prompt</b>	"Length"
<b>units</b>	"lengthMetric"
<b>defValue</b>	gpdK180_resValue( 'l' ?id cdfld ?returnString t )
<b>type</b>	"string"
<b>display</b>	"t"
<b>editable</b>	"nequal(cdfgData->calculatedParam->value \"Length\")"
<b>callback</b>	"gpdK180_resCB( 'l' )"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

CDF parameters

<b>name</b>	"m"
<b>prompt</b>	"Multiplier"
<b>defValue</b>	"1"
<b>type</b>	"string"
<b>display</b>	"t"
<b>editable</b>	"t"
<b>callback</b>	"gpdK180_resCB( 'm' )"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

CDF parameters

<b>name</b>	"s"
<b>prompt</b>	"S Factor"
<b>defValue</b>	"1"
<b>type</b>	"string"
<b>display</b>	"t"
<b>editable</b>	"t"
<b>callback</b>	"gpdK180_resCB( 's' )"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## resParamsB\_multiSeg - Resistor CDF Parameters (Part B)

CDF macro resParamsB\_multiSeg

CDF device params

<b>segmented</b>	t	private	Segmented Flag (do not modify)
------------------	---	---------	--------------------------------

CDF parameters

<b>name</b>	"l"
<b>prompt</b>	"Total Length"
<b>units</b>	"lengthMetric"
<b>defValue</b>	gpdK180_resValue( 'l' ?id cdfld ?returnString t )
<b>type</b>	"string"
<b>display</b>	"t"
<b>editable</b>	"nil"
<b>callback</b>	"gpdK180_resCB( 'l' )"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

CDF parameters

<b>name</b>	"sl"
<b>prompt</b>	"Segment Length"
<b>units</b>	"lengthMetric"
<b>defValue</b>	gpdK180_resValue( 'sl' ?id cdfld ?returnString t )
<b>type</b>	"string"
<b>display</b>	"t"
<b>editable</b>	"nequal(cdfgData->calculatedParam->value \"Length\")"
<b>callback</b>	"gpdK180_resCB( 'sl' )"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

CDF parameters

<b>name</b>	"effL"
<b>prompt</b>	"Effective Length"
<b>units</b>	"lengthMetric"
<b>defValue</b>	gpdK180_resValue( 'sl' ?id cdfld ?returnString t )
<b>type</b>	"string"
<b>display</b>	"nil"
<b>editable</b>	"nil"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF macro resParamsB\_multiSeg

## CDF parameters

<b>name</b>	"segments"
<b>prompt</b>	"Number of Segments"
<b>defValue</b>	gpdK180_resValue( 'segments ?id cdfld ?returnString nil)
<b>type</b>	"int"
<b>display</b>	"t"
<b>callback</b>	"gpdK180_resCB( 'segments )"

## CDF parameters

<b>name</b>	"connection"
<b>prompt</b>	"Connection"
<b>type</b>	"cyclic"
<b>defValue</b>	gpdK180_resValue( 'connection ?id cdfld )
<b>choices</b>	gpdK180_resValue( 'connection ?id cdfld ?type 'choices )
<b>display</b>	"t"
<b>callback</b>	"gpdK180_resCB( 'connection )"

## CDF parameters

<b>name</b>	"ssecs"
<b>prompt</b>	"Series Segments"
<b>defValue</b>	gpdK180_resValue( 'ssecs ?id cdfld ?returnString nil)
<b>type</b>	"int"
<b>display</b>	"nil"
<b>editable</b>	"nil"
<b>callback</b>	"gpdK180_resCB( 'ssecs )"

## CDF parameters

<b>name</b>	"psecs"
<b>prompt</b>	"Parallel Segments"
<b>defValue</b>	gpdK180_resValue( 'psecs ?id cdfld ?returnString nil)
<b>type</b>	"int"
<b>display</b>	"nil"
<b>editable</b>	"nil"
<b>callback</b>	"gpdK180_resCB( 'psecs )"

## resParamsC - Resistor CDF Parameters (Part C)

CDF macro resParamsC

CDF parameters

<b>name</b>	"cntRows"
<b>prompt</b>	"Contact Rows"
<b>defValue</b>	gpd180_resValue( 'cntRows ?id cdfld ?returnString nil)
<b>type</b>	"int"
<b>display</b>	"nil"
<b>callback</b>	"gpd180_resCB( 'cntRows )"

CDF parameters

<b>name</b>	"leftDummy"
<b>prompt</b>	"Left Dummy"
<b>type</b>	"boolean"
<b>defValue</b>	nil
<b>display</b>	"t"

CDF parameters

<b>name</b>	"rightDummy"
<b>prompt</b>	"Right Dummy"
<b>type</b>	"boolean"
<b>defValue</b>	nil
<b>display</b>	"t"

CDF parameters

<b>name</b>	"cntOnDummy"
<b>prompt</b>	"Contact on Dummy"
<b>type</b>	"boolean"
<b>defValue</b>	t
<b>display</b>	"gpd180_resDisplay( 'cntOnDummy )"

## resParamsTap - Resistor CDF Parameters (Tap)

CDF macro resParamsTap

CDF parameters

<b>name</b>	"showTapProps"
<b>prompt</b>	"Show Tap Properties"
<b>type</b>	"boolean"
<b>defValue</b>	nil
<b>display</b>	"t"

CDF parameters

<b>name</b>	"leftTap"
<b>prompt</b>	"Left Tap"
<b>type</b>	"boolean"
<b>defValue</b>	t
<b>display</b>	"gpd180_resDisplay( 'leftTap)"

CDF parameters

<b>name</b>	"rightTap"
<b>prompt</b>	"Right Tap"
<b>type</b>	"boolean"
<b>defValue</b>	nil
<b>display</b>	"gpd180_resDisplay( 'rightTap)"

CDF parameters

<b>name</b>	"topTap"
<b>prompt</b>	"Top Tap"
<b>type</b>	"boolean"
<b>defValue</b>	nil
<b>display</b>	"gpd180_resDisplay( 'topTap)"

CDF parameters

<b>name</b>	"bottomTap"
<b>prompt</b>	"Bottom Tap"
<b>type</b>	"boolean"
<b>defValue</b>	nil
<b>display</b>	"gpd180_resDisplay( 'bottomTap)"

## resParamsD - Resistor CDF Parameters (Part D)

CDF macro resParamsD

CDF parameters

<b>name</b>	"otherParams"
<b>prompt</b>	"Show Other Parameters"
<b>type</b>	"boolean"
<b>display</b>	"gpd180_resDisplay( 'otherParams)'"

CDF parameters

<b>name</b>	"rsh"
<b>prompt</b>	"Sheet Resistivity"
<b>units</b>	"resistance"
<b>defValue</b>	gpd180_resValue( 'rsh ?id cdfld ?returnString t )
<b>type</b>	"string"
<b>display</b>	"gpd180_resDisplay( 'rsh)'"
<b>editable</b>	"nil"
<b>callback</b>	"gpd180_resCB( 'rsh )"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF macro resParamsD

## CDF parameters

<b>name</b>	"rbody"
<b>prompt</b>	"Body Resistance"
<b>units</b>	"resistance"
<b>defValue</b>	gpd180_resValue( 'rbody ?id cdfld ?returnString t )
<b>type</b>	"string"
<b>display</b>	"gpd180_resDisplay( 'rbody )"
<b>editable</b>	"nil"
<b>callback</b>	"gpd180_resCB( 'rbody )"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"rcont"
<b>prompt</b>	"Contact Resistance"
<b>units</b>	"resistance"
<b>defValue</b>	gpd180_resValue( 'rcont ?id cdfld ?returnString t )
<b>type</b>	"string"
<b>display</b>	"gpd180_resDisplay( 'rcont )"
<b>editable</b>	"nil"
<b>callback</b>	"gpd180_resCB( 'rcont )"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"rend"
<b>prompt</b>	"End Resistance"
<b>units</b>	"resistance"
<b>defValue</b>	gpd180_resValue( 'rend ?id cdfld ?returnString t )
<b>type</b>	"string"
<b>display</b>	"gpd180_resDisplay( 'rend )"
<b>editable</b>	"nil"
<b>callback</b>	"gpd180_resCB( 'rend )"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF macro resParamsD

## CDF parameters

<b>name</b>	"tc1"
<b>prompt</b>	"Temperature Coefficient 1"
<b>defValue</b>	gpd180_resValue( 'tc1 ?id cdfld ?returnString t )
<b>type</b>	"string"
<b>display</b>	"gpd180_resDisplay( 'tc1)"
<b>editable</b>	"nil"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"tc2"
<b>prompt</b>	"Temperature Coefficient 2"
<b>defValue</b>	gpd180_resValue( 'tc2 ?id cdfld ?returnString t )
<b>type</b>	"string"
<b>display</b>	"gpd180_resDisplay( 'tc2)"
<b>editable</b>	"nil"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"



## CDF macro resParamsD

## CDF parameters

<b>name</b>	"vc1"
<b>prompt</b>	"Voltage Coefficient 1"
<b>defValue</b>	gpdk180_resValue( 'vc1 ?id cdfld ?returnString t )
<b>type</b>	"string"
<b>display</b>	"gpdk180_resDisplay( 'vc1 )"
<b>editable</b>	"nil"
<b>parseAsNumber</b>	"yes"

## CDF parameters

<b>name</b>	"vc2"
<b>prompt</b>	"Voltage Coefficient 2"
<b>defValue</b>	gpdk180_resValue( 'vc2 ?id cdfld ?returnString t )
<b>type</b>	"string"
<b>display</b>	"gpdk180_resDisplay( 'vc2 )"
<b>editable</b>	"nil"
<b>parseAsNumber</b>	"yes"

## CDF parameters

<b>name</b>	"dw"
<b>prompt</b>	"Delta Width"
<b>units</b>	"lengthMetric"
<b>defValue</b>	gpdk180_resValue( 'dw ?id cdfld ?returnString t )
<b>type</b>	"string"
<b>display</b>	"gpdk180_resDisplay( 'dw )"
<b>editable</b>	"nil"
<b>callback</b>	"gpdk180_resCB( 'dw )"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"dl"
<b>prompt</b>	"Delta Length"
<b>units</b>	"lengthMetric"
<b>defValue</b>	gpdk180_resValue( 'dl ?id cdfld ?returnString t )
<b>type</b>	"string"
<b>display</b>	"gpdk180_resDisplay( 'dl )"
<b>editable</b>	"nil"
<b>callback</b>	"gpdk180_resCB( 'dl )"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## resParamsE - Resistor CDF Parameters (Part E)

CDF macro resParamsE

CDF parameters

<b>name</b>	"trise"
<b>prompt</b>	"Temp rise from ambient"
<b>defValue</b>	""
<b>type</b>	"string"
<b>display</b>	"nil"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

CDF parameters

<b>name</b>	"resform"
<b>prompt</b>	"Resistance Form"
<b>defValue</b>	""
<b>type</b>	"string"
<b>display</b>	"gpd180_resDisplay( 'resform )"
<b>parseAsCEL</b>	"yes"

CDF parameters

<b>name</b>	"isnoisy"
<b>prompt</b>	"Generate noise?"
<b>defValue</b>	"yes"
<b>choices</b>	'("yes" "no")
<b>type</b>	"cyclic"
<b>display</b>	"gpd180_resDisplay( 'isnoisy )"
<b>parseAsCEL</b>	"yes"

CDF parameters

<b>name</b>	"scale"
<b>prompt</b>	"Scale factor"
<b>defValue</b>	""
<b>type</b>	"string"
<b>display</b>	"gpd180_resDisplay( 'scale )"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF macro resParamsE

## CDF parameters

<b>name</b>	"ac"
<b>prompt</b>	"AC resistance"
<b>units</b>	"resistance"
<b>defValue</b>	""
<b>type</b>	"string"
<b>display</b>	"gpd180_resDisplay( 'ac )"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"dtemp"
<b>prompt</b>	"Temperature difference"
<b>defValue</b>	""
<b>type</b>	"string"
<b>display</b>	"gpd180_resDisplay( 'dtemp )"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"hrc"
<b>prompt</b>	"Capacitance connected"
<b>units</b>	"capacitance"
<b>defValue</b>	""
<b>type</b>	"string"
<b>display</b>	"gpd180_resDisplay( 'hrc )"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"source"
<b>prompt</b>	"Source component"
<b>type</b>	"boolean"
<b>display</b>	"gpd180_resDisplay( 'source )"

## resParamsF - Resistor CDF Parameters (Part F)

CDF macro resParamsF

CDF parameters

<b>name</b>	"pasUpdateParamList"
<b>prompt</b>	"CDF Param Update List"
<b>defValue</b>	"w"
<b>type</b>	"string"
<b>display</b>	"nil"

CDF properties

<b>formInitProc</b>	"PasCdfFormInit"
<b>doneProc</b>	"PasCdfDone"
<b>buttonFieldWidth</b>	340
<b>fieldHeight</b>	35
<b>fieldWidth</b>	350
<b>promptWidth</b>	175
<b>instNameType</b>	"schematic"
<b>instDisplayMode</b>	"instName"
<b>netNameType</b>	"schematic"
<b>termSimType</b>	"DC"
<b>termDisplayMode</b>	"voltage"
<b>paramSimType</b>	"DC"
<b>paramEvaluate</b>	"nil nil nil t nil"
<b>paramDisplayMode</b>	"parameter"
<b>modelLabelSet</b>	"tc1 tc2 coefs"
<b>opPointLabelSet</b>	"v i pwr"
<b>paramLabelSet</b>	"-resType r w l"

## resSimInfo2 - 2 terminal Resistor Simulation Information

CDF macro resSimInfo2

CDF siminfo

<b>simulator</b>	auCdl
<b>netlistProcedure</b>	ansCdlCompPrim
<b>instParameters</b>	(R W L M)
<b>componentName</b>	nil
<b>termOrder</b>	(PLUS MINUS)
<b>propMapping</b>	(nil R r M m W w L l)
<b>namePrefix</b>	"R"

CDF siminfo

<b>simulator</b>	auLvs
<b>netlistProcedure</b>	ansLvsCompPrim
<b>instParameters</b>	(r l w m)
<b>termOrder</b>	(PLUS MINUS)
<b>deviceTerminals</b>	nil
<b>propMapping</b>	nil
<b>permuteRule</b>	"(p PLUS MINUS)"
<b>namePrefix</b>	"R"

CDF siminfo

<b>simulator</b>	ams
<b>isPrimitive</b>	t
<b>otherParameters</b>	(model)
<b>termOrder</b>	(PLUS MINUS)
<b>propMapping</b>	(nil sl effL)
<b>instParameters</b>	(model sl w)

## resSimInfo3 - 3 terminal Resistor Simulation Information

CDF macro resSimInfo3

CDF siminfo

<b>simulator</b>	auCdl
<b>netlistProcedure</b>	ansCdlCompPrim
<b>instParameters</b>	(R W L M)
<b>componentName</b>	nil
<b>termOrder</b>	(PLUS MINUS B)
<b>propMapping</b>	(nil R r M m W w L l)
<b>namePrefix</b>	"R"

CDF siminfo

<b>simulator</b>	auLvs
<b>netlistProcedure</b>	ansLvsCompPrim
<b>instParameters</b>	(r l w m)
<b>termOrder</b>	(PLUS MINUS B)
<b>deviceTerminals</b>	nil
<b>propMapping</b>	nil
<b>permuteRule</b>	"(p PLUS MINUS )"
<b>namePrefix</b>	"R"

CDF siminfo

<b>simulator</b>	ams
<b>isPrimitive</b>	t
<b>otherParameters</b>	(model)
<b>termOrder</b>	(PLUS MINUS B)
<b>propMapping</b>	(nil sl effL)
<b>instParameters</b>	(model sl w)

**res2\_uniSeg - 2 terminal Resistor CDF    res3\_uniSeg - 3 terminal Resistor CDF**

CDF macro res2\_uniSeg

```
include macro resParamsA
include macro resParamsB_uniSeg
include macro resParamsC
include macro resParamsD
include macro resParamsE
include macro resParamsF
include macro resSimInfo2
```

CDF macro res3\_uniSeg

```
include macro resParamsA
include macro resParamsB_uniSeg
include macro resParamsC
include macro resParamsTap
include macro resParamsD
include macro resParamsE
include macro resParamsF
include macro resSimInfo3
```

**res2\_multiSeg - 2 terminal Resistor CDF    res3\_multiSeg - 3 terminal Resistor CDF**

CDF macro res2\_multiSeg

```
include macro resParamsA
include macro resParamsB_multiSeg
include macro resParamsC
include macro resParamsD
include macro resParamsE
include macro resParamsF
include macro resSimInfo2
```

CDF macro res3\_multiSeg

```
include macro resParamsA
include macro resParamsB_multiSeg
include macro resParamsC

include macro resParamsD
include macro resParamsE
include macro resParamsF
include macro resSimInfo3
```

## Resistors Schematic Components

## resPrim

CDF macro resPrim

CDF parameters

<b>name</b>	"r"
<b>prompt</b>	"res"
<b>defValue</b>	"pPar(\"r\")"
<b>units</b>	"lengthMetric"
<b>type</b>	"string"
<b>display</b>	"nil"
<b>editable</b>	"nil"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

CDF parameters

<b>name</b>	"w"
<b>prompt</b>	"Width (m)"
<b>defValue</b>	"pPar(\"w\")"
<b>units</b>	"lengthMetric"
<b>type</b>	"string"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

CDF parameters

<b>name</b>	"l"
<b>prompt</b>	"Length (m)"
<b>defValue</b>	"pPar(\"l\")"
<b>units</b>	"lengthMetric"
<b>type</b>	"string"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

CDF parameters

<b>name</b>	"sl"
<b>prompt</b>	"Length (m)"
<b>defValue</b>	"pPar(\"sl\")"
<b>units</b>	"lengthMetric"
<b>type</b>	"string"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"



## resPrim2

CDF gpdk180 resPrim2

include macro resPrim

CDF siminfo

<b>simulator</b>	spectre
<b>otherParameters</b>	(model)
<b>termMapping</b>	(nil PLUS ":1" MINUS ":2" )
<b>termOrder</b>	(PLUS MINUS)
<b>instParameters</b>	(sl w)

CDF siminfo

<b>simulator</b>	ams
<b>isPrimitive</b>	t
<b>otherParameters</b>	(model)
<b>termOrder</b>	(PLUS MINUS)
<b>instParameters</b>	(model sl w)

CDF siminfo

<b>simulator</b>	ads
<b>netlistProcedure</b>	ADSSimSubcktCall
<b>otherParameters</b>	(model)
<b>instParameters</b>	(sl w)
<b>componentName</b>	nil
<b>termOrder</b>	(PLUS MINUS)
<b>termMapping</b>	(nil PLUS ":P1" MINUS ":P2")
<b>propMapping</b>	nil
<b>typeMapping</b>	nil
<b>uselib</b>	nil

CDF properties

<b>formInitProc</b>	""
<b>doneProc</b>	""
<b>buttonFieldWidth</b>	340
<b>fieldHeight</b>	35
<b>fieldWidth</b>	350
<b>promptWidth</b>	175
<b>opPointLabelSet</b>	"res v i"
<b>paramLabelSet</b>	"-model r w l"

CDF cellview

<b>symbol</b>	baseline_gpdk	res2	symbol
<b>spectre</b>	baseline_gpdk	res2	symbol
<b>ads</b>	baseline_gpdk	res2	symbol

## resPrim3

CDF gpdK180 resPrim3

include macro resPrim

CDF siminfo

<b>simulator</b>	spectre
<b>otherParameters</b>	(model)
<b>termMapping</b>	(nil PLUS ":1" MINUS ":2" B ":3" )
<b>termOrder</b>	(PLUS MINUS B)
<b>instParameters</b>	(sl w)

CDF siminfo

<b>simulator</b>	ams
<b>isPrimitive</b>	t
<b>otherParameters</b>	(model)
<b>termOrder</b>	(PLUS MINUS B)
<b>instParameters</b>	(model sl w)

CDF siminfo

<b>simulator</b>	ads
<b>netlistProcedure</b>	ADSSimSubcktCall
<b>otherParameters</b>	(model)
<b>instParameters</b>	(sl w)
<b>componentName</b>	nil
<b>termOrder</b>	(PLUS MINUS B)
<b>termMapping</b>	(nil PLUS ":P1" MINUS ":P2" B ":P3")
<b>propMapping</b>	nil
<b>typeMapping</b>	nil
<b>uselib</b>	nil

CDF properties

<b>formInitProc</b>	""
<b>doneProc</b>	""
<b>buttonFieldWidth</b>	340
<b>fieldHeight</b>	35
<b>fieldWidth</b>	350
<b>promptWidth</b>	175
<b>opPointLabelSet</b>	"res v i"
<b>paramLabelSet</b>	"-model r w l"

CDF cellview

<b>symbol</b>	baseline_gpdK	res3	symbol
<b>spectre</b>	baseline_gpdK	res3	symbol
<b>ads</b>	baseline_gpdK	res3	symbol

# Resistors

## polyres

CDF gpd180 polyres

include macro res2\_multiSeg

CDF siminfo

<b>simulator</b>	auLvs
<b>componentName</b>	"\$cell"

CDF siminfo

<b>simulator</b>	auCdl
<b>modelName</b>	"\$cell"

CDF device params

<b>grid</b>	\$grid	private	Manufacturing Grid Resolution
<b>scale</b>	1u	private	Dimension scale factor
<b>model</b>	"\$lib_\$cell"	private	
<b>rsh</b>	\$poly_rho	private	Sheet resistivity
<b>tc1</b>	0.0	private	Temperature Coefficient 1
<b>tc2</b>	0.0	private	Temperature Coefficient 2
<b>vc1</b>	0.0	private	Voltage Coefficient 1
<b>vc2</b>	0.0	private	Voltage Coefficient 2
<b>dw</b>	0.0	private	Delta Width
<b>dl</b>	0.0	private	Delta Length
<b>rcont</b>	lambda( (dpl) 5.0 / (dpl->rows * dpl->cols) )	private	Contact Resistance per head
<b>rend</b>	0.0	private	End Resistance per head
<b>category</b>	"res res/poly"	private	Library Manager Category
<b>schPcellPrimSymbol</b>	'( "\$lib" "resPrim2" "symbol" )	public	Sch PCell Prim Symbol View
<b>schPcellPinSymbol</b>	'( "basic" "iopin" "symbol" )	public	Schem PCell Pin Symbol View

# CDF gpd180 polyres

## CDF device params

<b>mwDevice</b>	\$rpoly_mwDevice	private	Minimum resistor width
<b>mlDevice</b>	\$rpoly_mlDevice	private	Minimum resistor length
<b>mnSquares</b>	\$rpoly_mnSquares	private	Minimum number of squares
<b>mwRes</b>	\$rpoly_mwRes	private	Minimum poly width
<b>msRes</b>	\$rpoly_msRes	private	Minimum spacing
<b>mwCont</b>	\$rpoly_mwCont	private	Minimum contact width
<b>msCont</b>	\$rpoly_msCont	private	Minimum contact spacing
<b>meResCont</b>	\$rpoly_meResCont	private	Minimum poly enclosure of contact
<b>meeResCont</b>	\$rpoly_meeResCont	private	Minimum poly end enclosure of contact
<b>mwM1</b>	\$rpoly_mwM1	private	Minimum metal 1 width
<b>msM1</b>	\$rpoly_msM1	private	Minimum metal 1 spacing
<b>meM1Cont</b>	\$rpoly_meM1Cont	private	Minimum metal 1 enclosure of contact
<b>meeM1Cont</b>	\$rpoly_meeM1Cont	private	Minimum metal 1 end enclosure of contact

## CDF device params

<b>melmplRes</b>	\$rnpoly_melmplRes	private	Minimum imlant enclosure of poly
<b>meldRes</b>	\$rpoly_meldRes	private	Minimum recognition layer enclosure of poly
<b>msldCont</b>	\$rpoly_msSbCont	private	Minimum recognition layer to contact spacing
<b>msldRes</b>	\$rpoly_msSbRes	private	Minium recognition layer to poly spacing

## CDF cellview

<b>symbol</b>	baseline_gpd1	res2	symbol	
<b>auCdl</b>	baseline_gpd1	res2	symbol	
<b>auLvs</b>	baseline_gpd1	res2	symbol	
<b>ivpcell</b>	\$lib	\$cell	symbol	5

## polyhres

CDF gpdk180 polyhres

include macro res2\_multiSeg

CDF siminfo

<b>simulator</b>	auLvs
<b>componentName</b>	"\$cell"

CDF siminfo

<b>simulator</b>	auCdl
<b>modelName</b>	"\$cell"

CDF device params

<b>grid</b>	\$grid	private	Manufacturing Grid Resolution
<b>scale</b>	1u	private	Dimension scale factor
<b>model</b>	"\$lib_\$cell"	private	
<b>rsh</b>	\$polyh_rho	private	Sheet resistivity
<b>tc1</b>	0.0	private	Temperature Coefficient 1
<b>tc2</b>	0.0	private	Temperature Coefficient 2
<b>vc1</b>	0.0	private	Voltage Coefficient 1
<b>vc2</b>	0.0	private	Voltage Coefficient 2
<b>dw</b>	0.0	private	Delta Width
<b>dl</b>	0.0	private	Delta Length
<b>rcont</b>	lambda( (dpl) 5.0 / (dpl->rows * dpl->cols) )	private	Contact Resistance per head
<b>rend</b>	lambda( (dpl) 7.5 / dpl->w )	private	End Resistance per head
<b>category</b>	"res res/poly"	private	Library Manager Category
<b>schPcellPrimSymbol</b>	'( "\$lib" "resPrim2" "symbol" )	public	Sch PCell Prim Symbol View
<b>schPcellPinSymbol</b>	'( "basic" "iopin" "symbol" )	public	Schem PCell Pin Symbol View

## CDF gpdK180 polyhres

## CDF device params

<b>mwDevice</b>	\$rpoly_mwDevice	private	Minimum resistor width
<b>mlDevice</b>	\$rpoly_mlDevice	private	Minimum resistor length
<b>mnSquares</b>	\$rpoly_mnSquares	private	Minimum number of squares
<b>mwRes</b>	\$rpoly_mwRes	private	Minimum poly width
<b>msRes</b>	\$rpoly_msRes	private	Minimum spacing
<b>mwCont</b>	\$rpoly_mwCont	private	Minimum contact width
<b>msCont</b>	\$rpoly_msCont	private	Minimum contact spacing
<b>meResCont</b>	\$rpoly_meResCont	private	Minimum poly enclosure of contact
<b>meeResCont</b>	\$rpoly_meeResCont	private	Minimum poly end enclosure of contact
<b>mwM1</b>	\$rpoly_mwM1	private	Minimum metal 1 width
<b>msM1</b>	\$rpoly_msM1	private	Minimum metal 1 spacing
<b>meM1Cont</b>	\$rpoly_meM1Cont	private	Minimum metal 1 enclosure of contact
<b>meeM1Cont</b>	\$rpoly_meeM1Cont	private	Minimum metal 1 end enclosure of contact

## CDF device params

<b>meImplRes</b>	\$rpoly_meImplRes	private	Minimum implant enclosure of poly
<b>meldRes</b>	\$rpoly_meldRes	private	Minimum recognition layer enclosure of poly
<b>msldCont</b>	\$rpoly_msSbCont	private	Minimum recognition layer to contact spacing
<b>msldRes</b>	\$rpoly_msSbRes	private	Minimum recognition layer to poly spacing

## CDF cellview

<b>symbol</b>	baseline_gpdK	res2	symbol	
<b>auCdl</b>	baseline_gpdK	res2	symbol	
<b>auLvs</b>	baseline_gpdK	res2	symbol	
<b>ivpcell</b>	\$lib	\$cell	symbol	5

## nplusres

CDF gpd180 nplusres

include macro res3\_multiSeg

CDF siminfo

<b>simulator</b>	auLvs
<b>componentName</b>	"\$cell"

CDF siminfo

<b>simulator</b>	auCdl
<b>modelName</b>	"\$cell"

CDF device params

<b>grid</b>	\$grid	private	Manufacturing Grid Resolution
<b>scale</b>	1u	private	Dimension scale factor
<b>model</b>	"\$lib_\$cell"	private	
<b>rsh</b>	\$nplus_rho	private	Sheet resistivity
<b>tc1</b>	0.0	private	Temperature Coefficient 1
<b>tc2</b>	0.0	private	Temperature Coefficient 2
<b>vc1</b>	0.0	private	Voltage Coefficient 1
<b>vc2</b>	0.0	private	Voltage Coefficient 2
<b>dw</b>	0.0	private	Delta Width
<b>dl</b>	0.0	private	Delta Length
<b>rcont</b>	lambda( dpl) 7.5 / (dpl->rows * dpl->cols) )	private	Contact Resistance per head
<b>rend</b>	0.0	private	End Resistance per head
<b>category</b>	"res res/diff"	private	Library Manager Category
<b>schPcellPrimSymbol</b>	'( "\$lib" "resPrim3" "symbol" )	public	Sch PCell Prim Symbol View
<b>schPcellPinSymbol</b>	'( "basic" "iopin" "symbol" )	public	Schem PCell Pin Symbol View

## CDF gpd180 nplusres

## CDF device params

<b>mwDevice</b>	\$rdiff_mwDevice	private	Minimum resistor width
<b>mlDevice</b>	\$rdiff_mlDevice	private	Minimum resistor length
<b>mnSquares</b>	\$rdiff_mnSquares	private	Minimum number of squares
<b>mwRes</b>	\$rdiff_mwRes	private	Minimum diffusion width
<b>msRes</b>	\$rdiff_msRes	private	Minimum diffusion spacing
<b>mwCont</b>	\$rdiff_mwCont	private	Minimum contact width
<b>msCont</b>	\$rdiff_msCont	private	Minimum contact spacing
<b>meResCont</b>	\$rdiff_meResCont	private	Minimum diffusion enclosure of contact
<b>meeResCont</b>	\$rdiff_meeResCont	private	Minimum diffusion end enclosure of contact
<b>mwM1</b>	\$rdiff_mwM1	private	Minimum metal 1 width
<b>msM1</b>	\$rdiff_msM1	private	Minimum metal 1 spacing
<b>meM1Cont</b>	\$rdiff_meM1Cont	private	Minimum metal 1 enclosure of contact
<b>meeM1Cont</b>	\$rdiff_meeM1Cont	private	Minimum metal 1 end enclosure of contact

## CDF device params

<b>melmplRes</b>	\$rndiff_melmplRes	private	Minimum implant enclosure of diff
<b>meldRes</b>	\$rdiff_meldRes	private	Minimum recognition layer enclosure of diff
<b>msldCont</b>	\$rdiff_msSbCont	private	Minimum recognition layer to contact spacing
<b>msldRes</b>	\$rdiff_msSbRes	private	Minimum recognition layer to diff spacing

## CDF cellview

<b>symbol</b>	baseline_gpd1	res3	symbol	
<b>auCdl</b>	baseline_gpd1	res3	symbol	
<b>auLvs</b>	baseline_gpd1	res3	symbol	
<b>ivpcell</b>	\$lib	\$cell	symbol	5



## pplusres

CDF gpdk180 pplusres

include macro res3\_multiSeg

CDF siminfo

<b>simulator</b>	auLvs
<b>componentName</b>	"\$cell"

CDF siminfo

<b>simulator</b>	auCdl
<b>modelName</b>	"\$cell"

CDF device params

<b>grid</b>	\$grid	private	Manufacturing Grid Resolution
<b>scale</b>	1u	private	Dimension scale factor
<b>model</b>	"\$lib_\$cell"	private	
<b>rsh</b>	\$pplus_rho	private	Sheet resistivity
<b>tc1</b>	0.0	private	Temperature Coefficient 1
<b>tc2</b>	0.0	private	Temperature Coefficient 2
<b>vc1</b>	0.0	private	Voltage Coefficient 1
<b>vc2</b>	0.0	private	Voltage Coefficient 2
<b>dw</b>	0.0	private	Delta Width
<b>dl</b>	0.0	private	Delta Length
<b>rcont</b>	lambda( dpl) 7.5 / (dpl->rows * dpl->cols) )	private	Contact Resistance per head
<b>rend</b>	0.0	private	End Resistance per head
<b>category</b>	"res res/diff"	private	Library Manager Category
<b>schPcellPrimSymbol</b>	'( "\$lib" "resPrim3" "symbol" )	public	Sch PCell Prim Symbol View
<b>schPcellPinSymbol</b>	'( "basic" "iopin" "symbol" )	public	Schem PCell Pin Symbol View

## CDF gpd180 pplusres

## CDF device params

<b>mwDevice</b>	\$rdiff_mwDevice	private	Minimum resistor width
<b>mlDevice</b>	\$rdiff_mlDevice	private	Minimum resistor length
<b>mnSquares</b>	\$rdiff_mnSquares	private	Minimum number of squares
<b>mwRes</b>	\$rdiff_mwRes	private	Minimum diffusion width
<b>msRes</b>	\$rdiff_msRes	private	Minimum diffusion spacing
<b>mwCont</b>	\$rdiff_mwCont	private	Minimum contact width
<b>msCont</b>	\$rdiff_msCont	private	Minimum contact spacing
<b>meResCont</b>	\$rdiff_meResCont	private	Minimum diffusion enclosure of contact
<b>meeResCont</b>	\$rdiff_meeResCont	private	Minimum diffusion end enclosure of contact
<b>mwM1</b>	\$rdiff_mwM1	private	Minimum metal 1 width
<b>msM1</b>	\$rdiff_msM1	private	Minimum metal 1 spacing
<b>meM1Cont</b>	\$rdiff_meM1Cont	private	Minimum metal 1 enclosure of contact
<b>meeM1Cont</b>	\$rdiff_meeM1Cont	private	Minimum metal 1 end enclosure of contact

## CDF device params

<b>meImplRes</b>	\$rpdiff_meImplRes	private	Minimum implant enclosure of diff
<b>meldRes</b>	\$rdiff_meldRes	private	Minimum recognition layer enclosure of diff
<b>msldCont</b>	\$rdiff_msSbCont	private	Minimum recognition layer to contact spacing
<b>msldRes</b>	\$rdiff_msSbRes	private	Minimum recognition layer to diff spacing
<b>meWellRes</b>	\$rpdiff_meWellRes	private	Minimum well enclosure of poly

## CDF cellview

<b>symbol</b>	baseline_gpd1	res3	symbol	
<b>auCdl</b>	baseline_gpd1	res3	symbol	
<b>auLvs</b>	baseline_gpd1	res3	symbol	
<b>ivpcell</b>	\$lib	\$cell	symbol	5

## nwellres

CDF gpd180 nwellres

include macro res2\_multiSeg

CDF siminfo

<b>simulator</b>	auLvs
<b>componentName</b>	"\$cell"

CDF siminfo

<b>simulator</b>	auCdl
<b>modelName</b>	"\$cell"

CDF device params

<b>grid</b>	\$grid	private	Manufacturing Grid Resolution
<b>scale</b>	1u	private	Dimension scale factor
<b>model</b>	"\$lib_\$cell"	private	
<b>rsh</b>	\$nwell_rho	private	Sheet resistivity
<b>tc1</b>	0.0	private	Temperature Coefficient 1
<b>tc2</b>	0.0	private	Temperature Coefficient 2
<b>vc1</b>	0.0	private	Voltage Coefficient 1
<b>vc2</b>	0.0	private	Voltage Coefficient 2
<b>dw</b>	0.0	private	Delta Width
<b>dl</b>	0.0	private	Delta Length
<b>rcont</b>	lambda( (dpl) 7.5 / (dpl->rows * dpl->cols) )	private	Contact Resistance per head
<b>rend</b>	lambda( (dpl) 160.0 / dpl->w )	private	End Resistance per head
<b>category</b>	"res res/nwell"	private	Library Manager Category
<b>schPcellPrimSymbol</b>	'( "\$lib" "resPrim2" "symbol" )	public	Sch PCell Prim Symbol View
<b>schPcellPinSymbol</b>	'( "basic" "iopin" "symbol" )	public	Schem PCell Pin Symbol View

CDF gpd180 nwellres

CDF device params

<b>mwDevice</b>	\$rnw_mwDevice	private	Minimum resistor width
<b>mlDevice</b>	\$rnw_mlDevice	private	Minimum resistor length
<b>mnSquares</b>	\$rnw_mnSquares	private	Minimum number of squares
<b>mwRes</b>	\$rnw_mwRes	private	Minimum nwell width
<b>msRes</b>	\$rnw_msRes	private	Minimum nwell spacing
<b>mwDiff</b>	\$rnw_mwDiff	private	Minimum diffusion width
<b>msDiff</b>	\$rnw_msDiff	private	Minimum diffusion spacing
<b>meResDiff</b>	\$rnw_meResDiff	private	Minimum nwell enclosure of diffusion
<b>meImplDiff</b>	\$rnw_meImplDiff	private	Minimum implant enclosure of diffusion
<b>mwCont</b>	\$rnw_mwCont	private	Minimum contact width
<b>msCont</b>	\$rnw_msCont	private	Minimum contact spacing
<b>meDiffCont</b>	\$rnw_meDiffCont	private	Minimum diffusion enclosure of contact
<b>meeDiffCont</b>	\$rnw_meeDiffCont	private	Minimum diffusion end enclosure of contact
<b>mwM1</b>	\$rnw_mwM1	private	Minimum metal 1 width
<b>msM1</b>	\$rnw_msM1	private	Minimum metal 1 spacing
<b>meM1Cont</b>	\$rnw_meM1Cont	private	Minimum metal 1 enclosure of contact
<b>meeM1Cont</b>	\$rnw_meeM1Cont	private	Minimum metal 1 end enclosure of contact

CDF device params

<b>meldRes</b>	\$rnw_meldRes	private	Minimum recognition layer enclosure of nwell
----------------	---------------	---------	--

CDF cellview

<b>symbol</b>	baseline_gpd1	res2	symbol	
<b>auCdl</b>	baseline_gpd1	res2	symbol	
<b>auLvs</b>	baseline_gpd1	res2	symbol	
<b>ivpcell</b>	\$lib	\$cell	symbol	5

# mxres

CDF gpdk180 mxres

include macro res2\_multiSeg

CDF siminfo

<b>simulator</b>	auLvs
<b>componentName</b>	"\$cell"

CDF siminfo

<b>simulator</b>	auCdl
<b>modelName</b>	"\$cell"

CDF Macro Table

<b>\$cell</b>	<b>\$m_rho</b>	<b>\$minW</b>	<b>\$minS</b>
m1res	\$m1_rho	{7A}	{7B}
m2res	\$m2_rho	{9A}	{9B}
m3res	\$m3_rho	{11A}	{11B}
m4res	\$m4_rho	{15A}	{15B}
m5res	\$m5_rho	{17A}	{17B}
m6res	\$m6_rho	{19A}	{19B}

CDF device params

<b>grid</b>	\$grid	private	Manufacturing Grid Resolution
<b>scale</b>	1u	private	Dimension scale factor
<b>model</b>	"\$lib_\$cell"	private	
<b>rsh</b>	\$m_rho	private	Sheet resistivity
<b>tc1</b>	0.0	private	Temperature Coefficient 1
<b>tc2</b>	0.0	private	Temperature Coefficient 2
<b>vc1</b>	0.0	private	Voltage Coefficient 1
<b>vc2</b>	0.0	private	Voltage Coefficient 2
<b>dw</b>	0.0	private	Delta Width
<b>dl</b>	0.0	private	Delta Length
<b>rcont</b>	0.0	private	Contact Resistance per head
<b>rend</b>	0.0	private	End Resistance per head
<b>category</b>	"res res/metal"	private	Library Manager Category
<b>schPcellPrimSymbol</b>	'( "\$lib" "resPrim2" "symbol" )	public	Sch PCell Prim Symbol View
<b>schPcellPinSymbol</b>	'( "basic" "iopin" "symbol" )	public	Schem PCell Pin Symbol View

## CDF gpdk180 mxres

## CDF device params

<b>mwDevice</b>	\$minW	private	Minimum resistor width
<b>mlDevice</b>	\$minW	private	Minimum resistor length
<b>mnSquares</b>	1	private	Minimum number of squares
<b>mwRes</b>	\$minW	private	Minimum poly width
<b>msRes</b>	\$minS	private	Minimum spacing
<b>mwCont</b>	{6A}	private	Minimum contact width
<b>msCont</b>	{6B}	private	Minimum contact spacing
<b>meResCont</b>	0.0	private	Minimum poly enclosure of contact
<b>meeResCont</b>	0.0	private	Minimum poly end enclosure of contact
<b>mwM1</b>	\$minW	private	Minimum metal 1 width
<b>msM1</b>	\$minS	private	Minimum metal 1 spacing
<b>meM1Cont</b>	0.0	private	Minimum metal 1 enclosure of contact
<b>meeM1Cont</b>	0.0	private	Minimum metal 1 end enclosure of contact

## CDF device params

<b>meldRes</b>	0.05	private	Minimum recognition layer enclosure of metal
<b>msldCont</b>	0.0	private	Minimum recognition layer to contact spacing
<b>msldRes</b>	0.0	private	Minium recognition layer to metal spacing

## CDF cellview

<b>symbol</b>	baseline_gpdk	res2	symbol	
<b>auCdl</b>	baseline_gpdk	res2	symbol	
<b>auLvs</b>	baseline_gpdk	res2	symbol	
<b>ivpcell</b>	\$lib	\$cell	symbol	5

## PCell Macros

### res\_uniSeg (macro)

pcell macro res\_uniSeg

PCell formal params

<b>w</b>	float	CDF
<b>l</b>	float	CDF
<b>leftDummy</b>	boolean	CDF
<b>rightDummy</b>	boolean	CDF
<b>cntOnDummy</b>	boolean	CDF
<b>cntRows</b>	int	CDF

include macro res\_connect

connection = "Series"

segments = 1

sl = l

### res\_multiSeg (macro)

pcell macro res\_multiSeg

PCell formal params

<b>w</b>	float	CDF
<b>sl</b>	float	CDF
<b>connection</b>	string	CDF
<b>segments</b>	int	CDF
<b>leftDummy</b>	boolean	CDF
<b>rightDummy</b>	boolean	CDF
<b>cntOnDummy</b>	boolean	CDF
<b>cntRows</b>	int	CDF

include macro res\_connect

### mres\_multiSeg (macro)

pcell macro mres\_multiSeg

PCell formal params

<b>w</b>	float	CDF
<b>sl</b>	float	CDF
<b>connection</b>	string	CDF
<b>segments</b>	int	CDF
<b>leftDummy</b>	boolean	CDF
<b>rightDummy</b>	boolean	CDF
<b>cntRows</b>	int	CDF

**res\_poly (macro)**

pcell macro res\_poly

1: include

```

dpt = PasGetDeviceProps( cv )
scale = dpt->scale
epsilon = grid / 10.0
mwRes = dpt->mwRes
mwCont = dpt->mwCont
msCont = dpt->msCont
meResCont = dpt->meResCont
meeResCont = dpt->meeResCont || meResCont
msldCont = dpt->msldCont || 0
msldRes = dpt->msldRes

msRes = dpt->msRes
mwM1 = dpt->mwM1
msM1 = dpt->msM1
meM1Cont = dpt->meM1Cont
meeM1Cont = dpt->meeM1Cont || meM1Cont

```

3: include

```

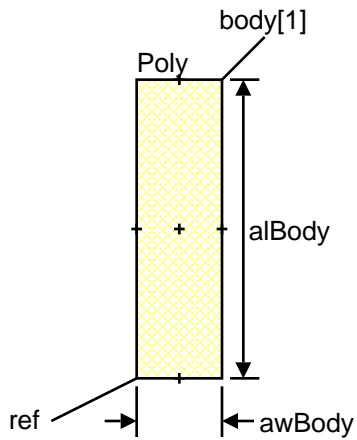
alBody = sl / scale
awBody = w / scale
sc = max(1 segments)
awHead = max( awBody mwRes (mwCont + (2.0 * meeResCont)) )
cntSpan = ( max(1 cntRows ) * ( mwCont + msCont ) ) - msCont
alHead = max( mwRes (cntSpan + (2.0 * meResCont)) )
asHead = msRes
asBody = asHead + awHead - awBody
dogbone = awBody + epsilon < awHead
xSepHB = if( dogbone then PasCeiling( ( awHead - awBody ) / 2.0 ) grid t ) else 0.0 )
bi = if( leftDummy then 0 else 1 )
ei = if( rightDummy then sc + 1 else sc )
xSepHM = meeResCont - meeM1Cont
ySepHM = meResCont - meM1Cont
alM1 = awHead - (2.0 * xSepHM)
awM1 = cntSpan + (2.0 * meM1Cont)
idOffset = msldCont - meResCont
idOffset = if( dogbone && msldRes then max( idOffset msldRes )
              else idOffset )
alBody = alBody + (2.0 * idOffset)
aeeResM1 = xSepHM

```

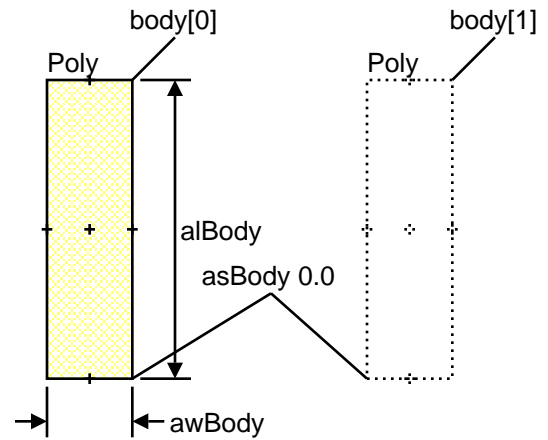


pcell macro res\_poly

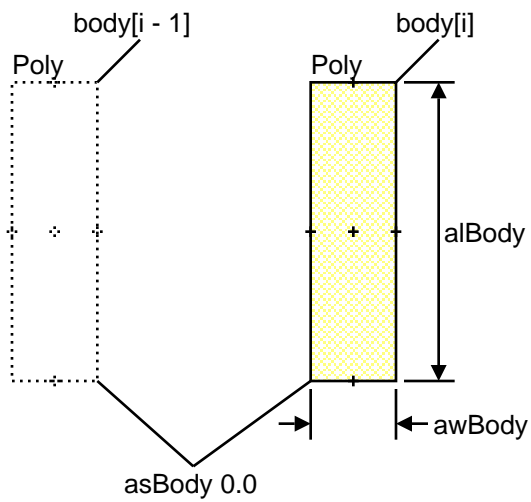
5: include



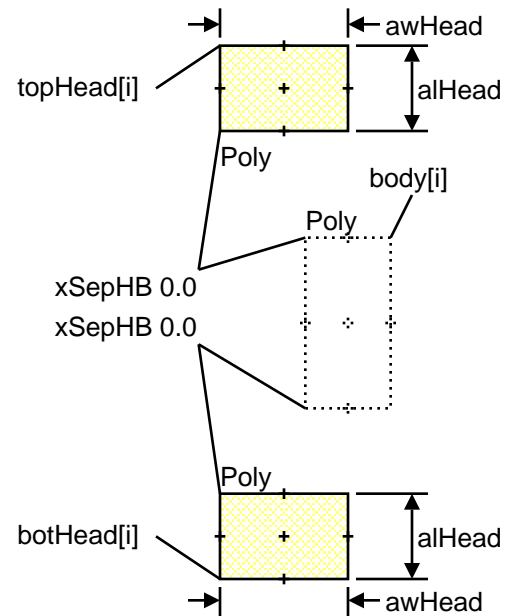
6: if leftDummy



7: for i 2 ei

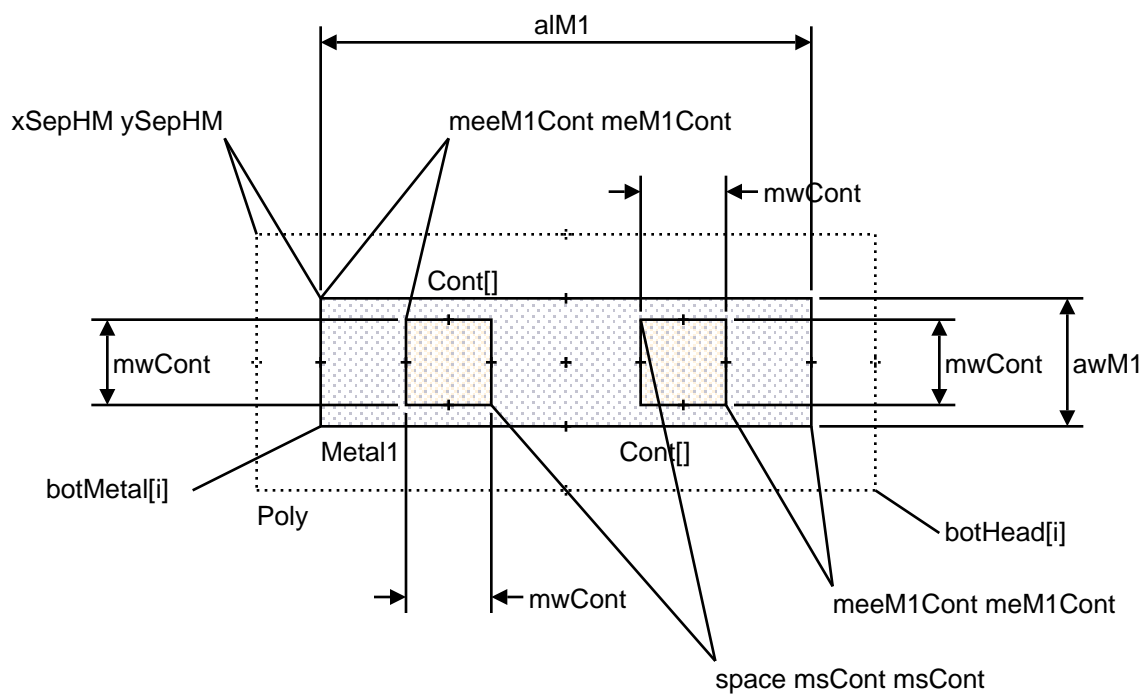


8: for i bi ei



9: for i bi ei

contact



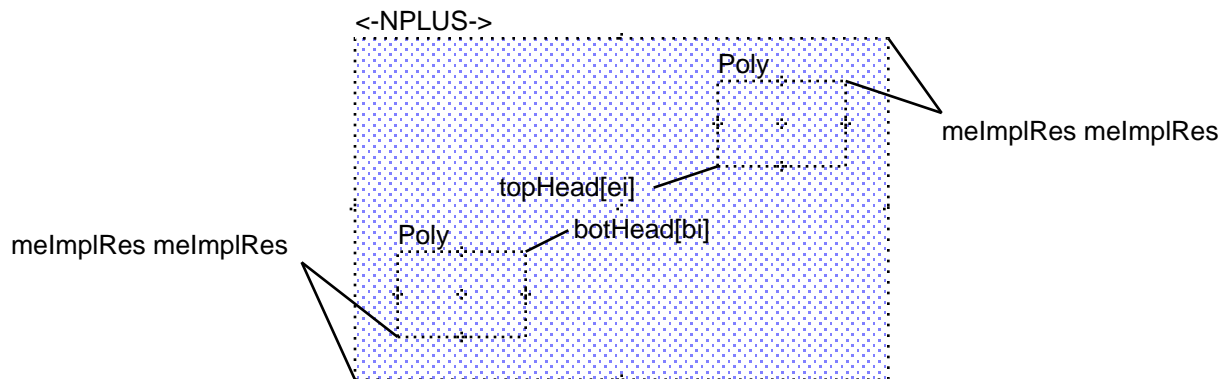
## res\_poly\_nplus (macro)

pcell macro res\_poly\_nplus

2: include

```
meImplRes = dpt->meImplRes
```

10: include



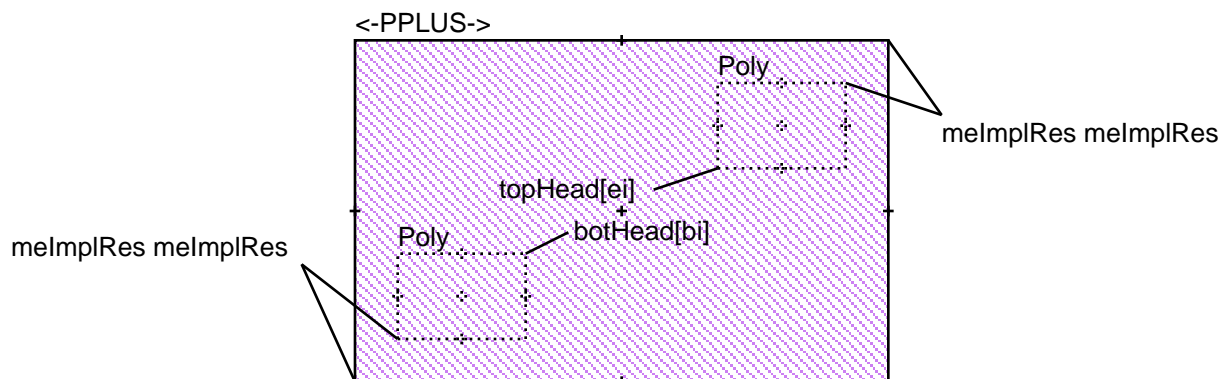
## res\_poly\_pplus (macro)

pcell macro res\_poly\_pplus

2: include

```
meImplRes = dpt->meImplRes
```

10: include



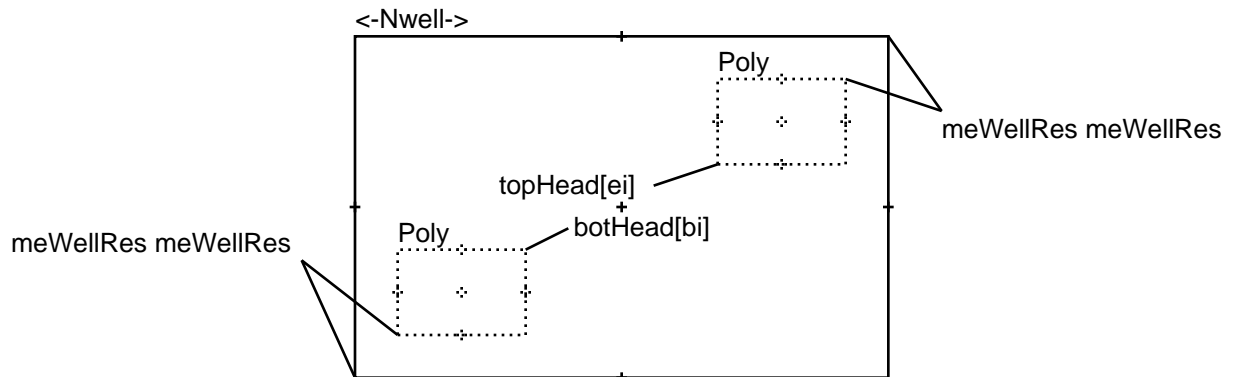
## res\_poly\_nw (macro)

pcell macro res\_poly\_nw

2: include

```
meWellRes = dpt->meWellRes
```

10: include



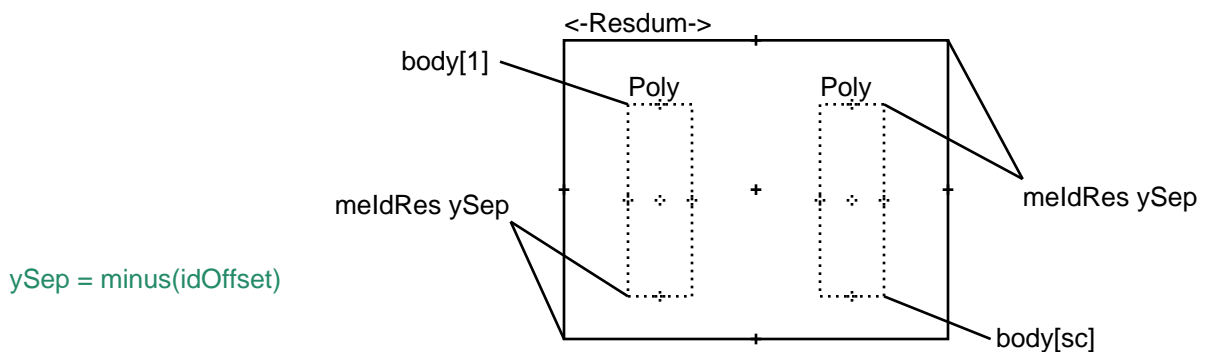
## res\_poly\_id (macro)

pcell macro res\_poly\_id

2: include

```
meldRes = dpt->meldRes || 0
```

10: include



## res\_poly\_sb (macro)

pcell macro res\_poly\_sb

2: include

```
meSbRes = dpt->meSbRes || 0      msSbRes = dpt->msSbRes
msSbCont = dpt->msSbCont || 0    maSb = dpt->maSb
```

10: include

1: include

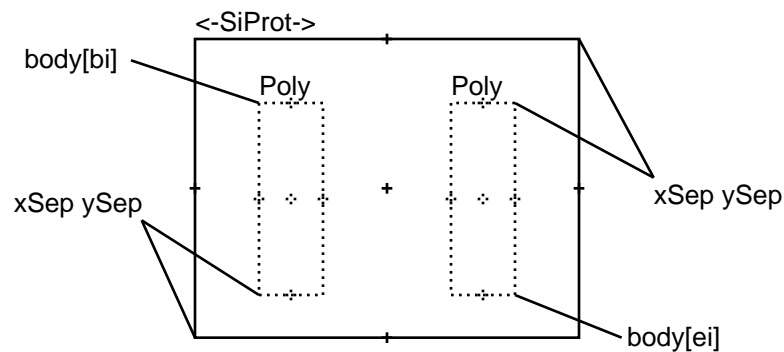
```
sbOffset = msSbCont - meResCont
sbOffset = if( dogbone && msSbRes then max( sbOffset msSbRes )
              else sbOffset )

xSep = meSbRes
ySep = minus(sbOffset)
```

2: if maSb

```
xSep = PasCeiling( ( ( ( maSb / ( alBody - (2.0 * sbOffset) ) ) -
                      ( ( ei - bi ) * ( awBody + asBody ) ) / 2.0 ) grid t )
xSep = max( xSep meSbRes)
```

3: include



**res\_diff (macro)**

pcell macro res\_diff

1: include

```

dpt = PasGetDeviceProps( cv )
scale = dpt->scale
epsilon = grid / 10.0
mwRes = dpt->mwRes
mwCont = dpt->mwCont
msCont = dpt->msCont
meResCont = dpt->meResCont
meeResCont = dpt->meeResCont || meResCont
msldCont = dpt->msldCont || 0
msldRes = dpt->msldRes

msRes = dpt->msRes
mwM1 = dpt->mwM1
msM1 = dpt->msM1
meM1Cont = dpt->meM1Cont
meeM1Cont = dpt->meeM1Cont || meM1Cont

```

3: include

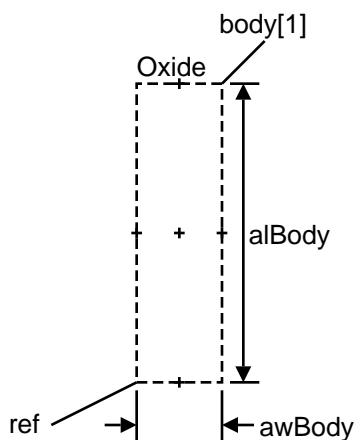
```

alBody = sl / scale
awBody = w / scale
sc = max(1 segments)
awHead = max( awBody mwRes (mwCont + (2.0 * meeResCont)) )
cntSpan = ( max(1 cntRows ) * ( mwCont + msCont ) ) - msCont
alHead = max( mwRes (cntSpan + (2.0 * meResCont)) )
asHead = msRes
asBody = asHead + awHead - awBody
dogbone = awBody + epsilon < awHead
xSepHB = if( dogbone then PasCeiling( ( awHead - awBody ) / 2.0 ) grid t ) else 0.0 )
bi = if( leftDummy then 0 else 1 )
ei = if( rightDummy then sc + 1 else sc )
xSepHM = meeResCont - meeM1Cont
ySepHM = meResCont - meM1Cont
alM1 = awHead - (2.0 * xSepHM)
awM1 = cntSpan + (2.0 * meM1Cont)
idOffset = msldCont - meResCont
idOffset = if( dogbone && msldRes then max( idOffset msldRes )
              else idOffset )
alBody = alBody + (2.0 * idOffset)
aeeResM1 = xSepHM

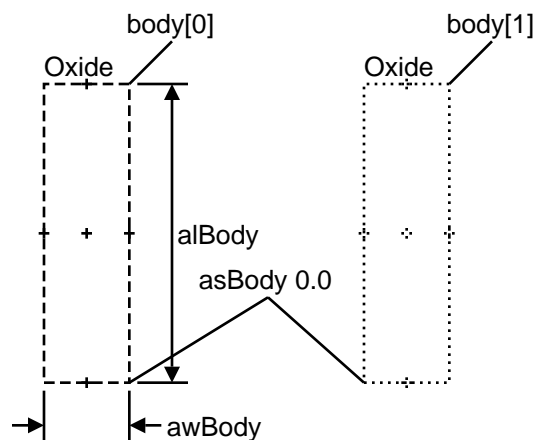
```

pcell macro res\_diff

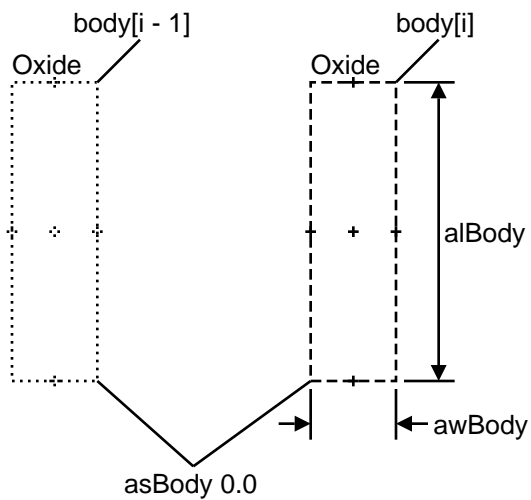
5: include



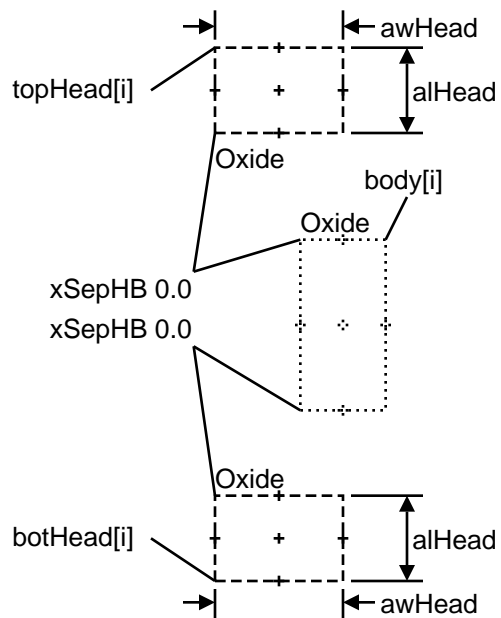
6: if leftDummy



7: for i 2 ei

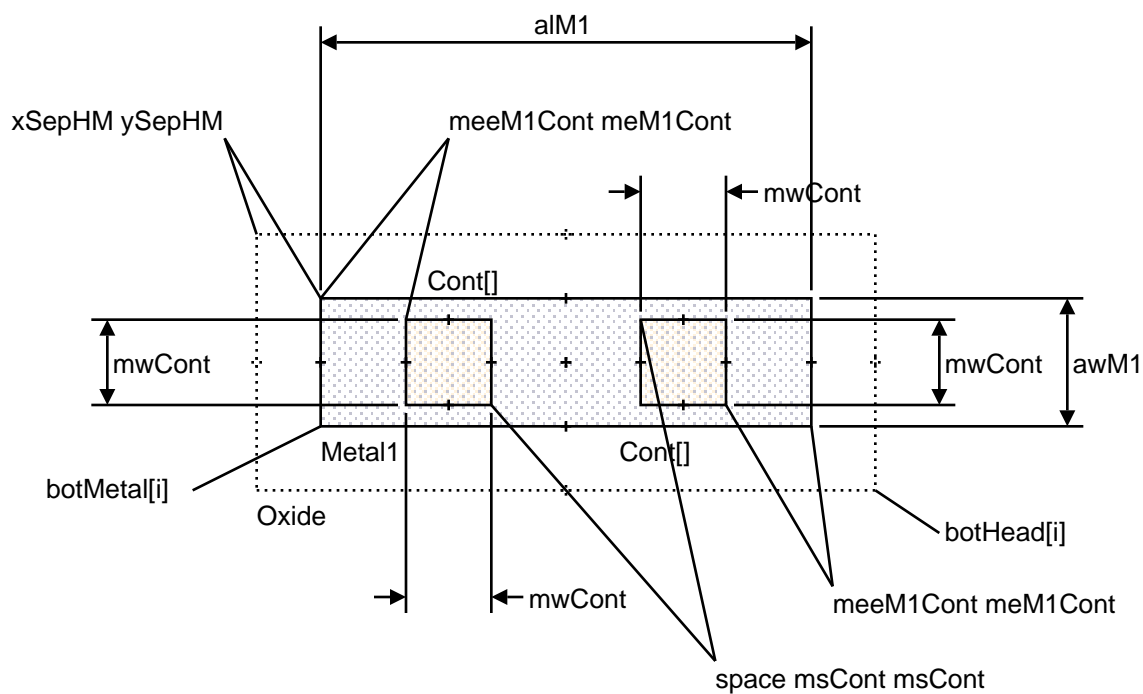


8: for i bi ei



9: for i bi ei

contact





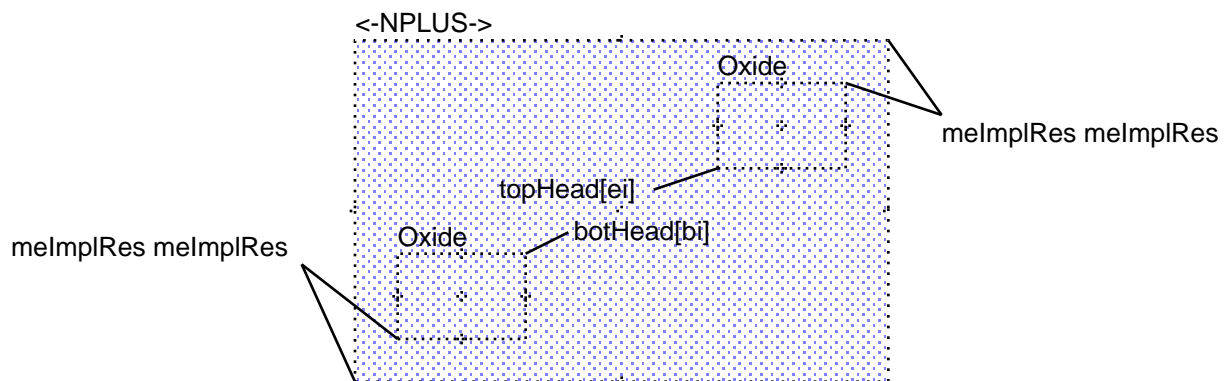
## res\_diff\_nplus (macro)

pcell macro res\_diff\_nplus

2: include

```
meImplRes = dpt->meImplRes
```

10: include



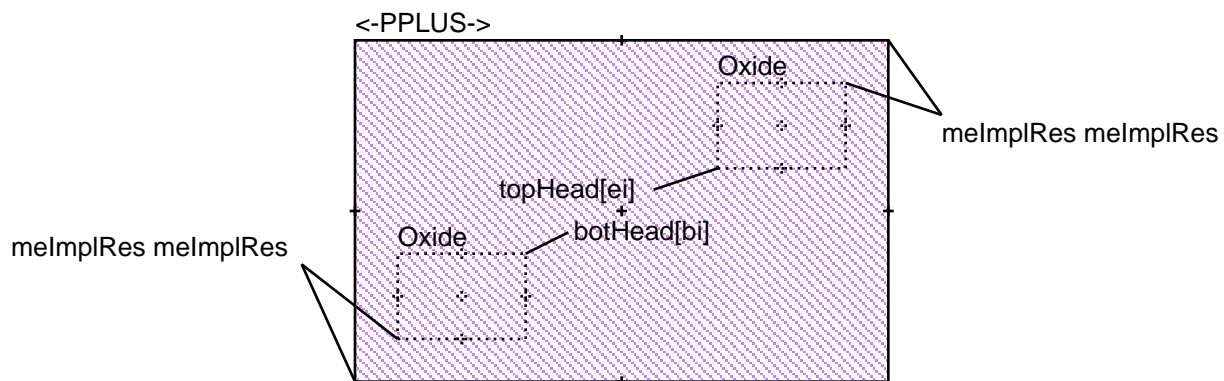
## res\_diff\_pplus (macro)

pcell macro res\_diff\_pplus

2: include

```
meImplRes = dpt->meImplRes
```

10: include



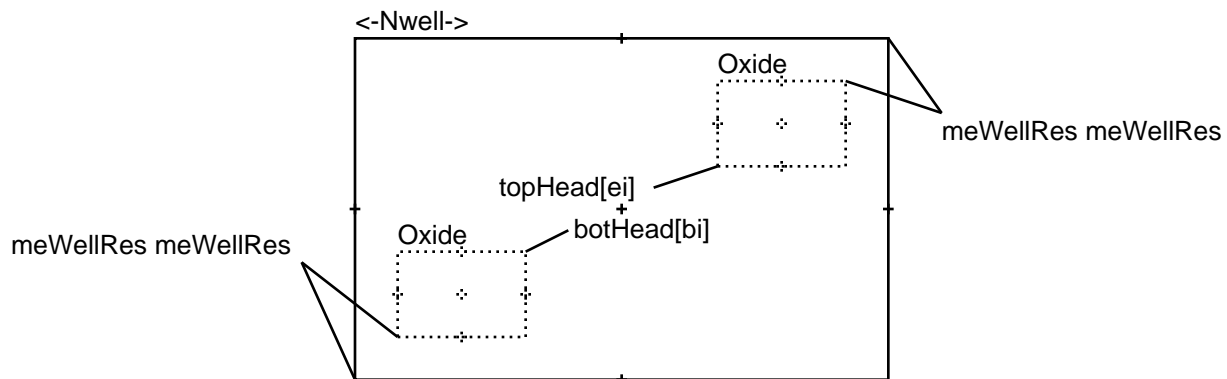
## res\_diff\_nw (macro)

pcell macro res\_diff\_nw

2: include

```
meWellRes = dpt->meWellRes
```

10: include



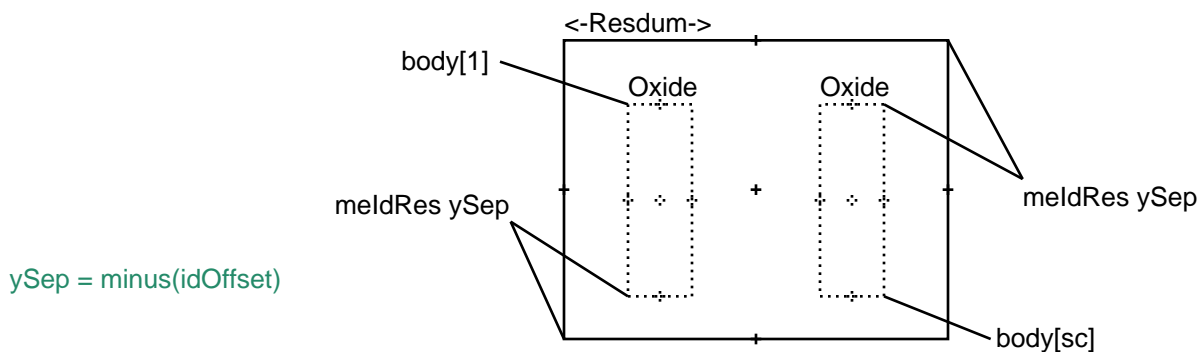
## res\_diff\_id (macro)

pcell macro res\_diff\_id

2: include

```
meldRes = dpt->meldRes || 0
```

10: include



## res\_nwell (macro)

pcell macro res\_nwell

1: include

```
dpt = PasGetDeviceProps( cv )
scale = dpt->scale
epsilon = grid / 10.0
mwRes = dpt->mwRes
mwDiff = dpt->mwDiff
meResDiff = dpt->meResDiff
mwCont = dpt->mwCont
msCont = dpt->msCont
meDiffCont = dpt->meDiffCont
meeDiffCont = dpt->meeDiffCont || meDiffCont
meImplDiff = dpt->meImplDiff

msRes = dpt->msRes
msDiff = dpt->msDiff || 0
meeResDiff = dpt->meeResDiff || meResDiff
mwM1 = dpt->mwM1
msM1 = dpt->msM1
meM1Cont = dpt->meM1Cont
meeM1Cont = dpt->meeM1Cont || meM1Cont
```

3: include

three

1: include

```
alBody = sl / scale
awBody = w / scale
sc = max(1 segments)
tmp = 2.0 * meeResDiff
awDiff = max( ( awBody - tmp ) ( mwRes - tmp ) ( mwCont + ( 2.0 * meeDiffCont ) ) )
awHead = awDiff + tmp
asHead = max( msRes ( msDiff - tmp ) )
asBody = asHead + awHead - awBody
dogbone = awBody + epsilon < awHead
xSepHB = if( dogbone then PasCeiling( ( (awHead - awBody) / 2.0 ) grid t ) else 0.0 )
bi = if( leftDummy then 0 else 1 )
ei = if( rightDummy then sc + 1 else sc )
xSepDM = meeDiffCont - meeM1Cont
ySepDM = meDiffCont - meM1Cont
alM1 = awDiff - (2.0 * xSepDM)
idOffset = meImplDiff - meResDiff
alBody = alBody + (2.0 * idOffset)
aeeResM1 = xSepDM + meeResDiff
```

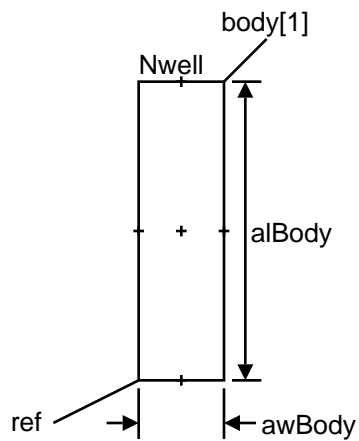
pcell macro res\_nwell

three

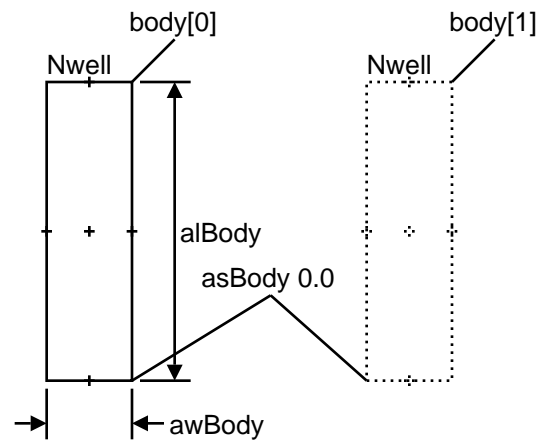
2: include

```
tmp = 2.0 * meResDiff
cntSpan = ( max(1 cntRows ) * ( mwCont + msCont ) ) - msCont
alDiff = max( ( mwRes - tmp ) (cntSpan + (2.0 * meDiffCont)) )
alHead = alDiff + tmp
awM1 = cntSpan + (2.0 * meM1Cont)
```

5: include

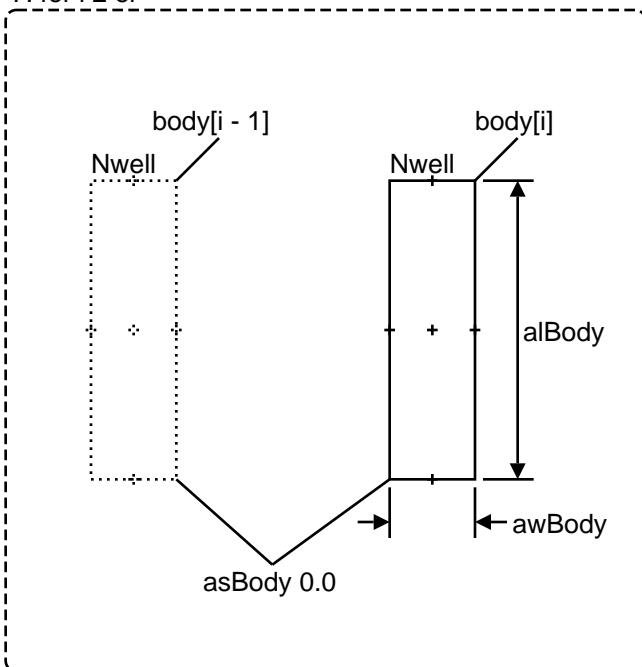


6: if leftDummy

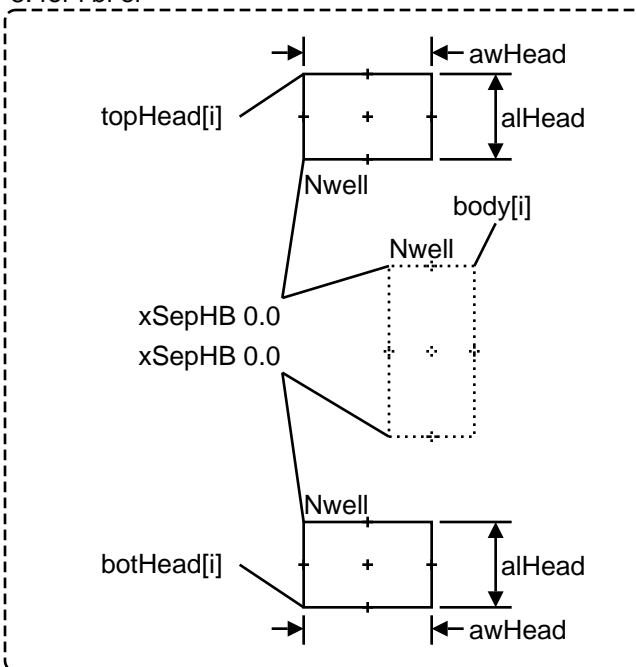


pcell macro res\_nwell

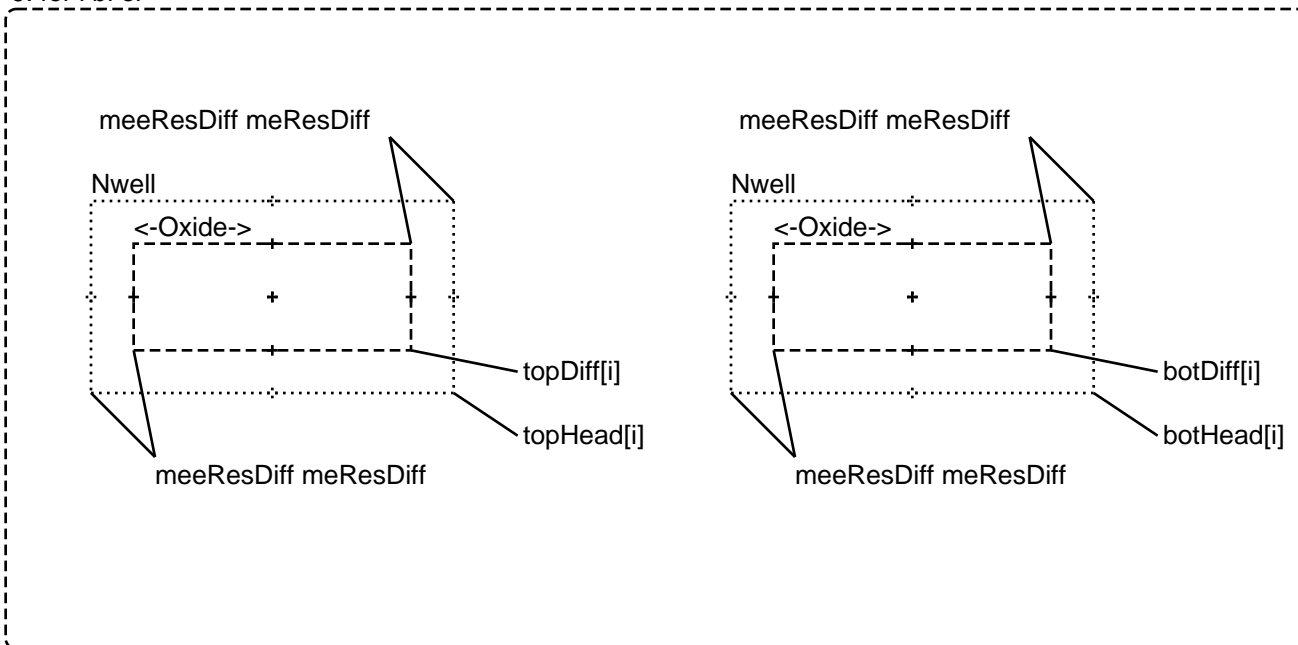
7: for i 2 ei



8: for i bi ei

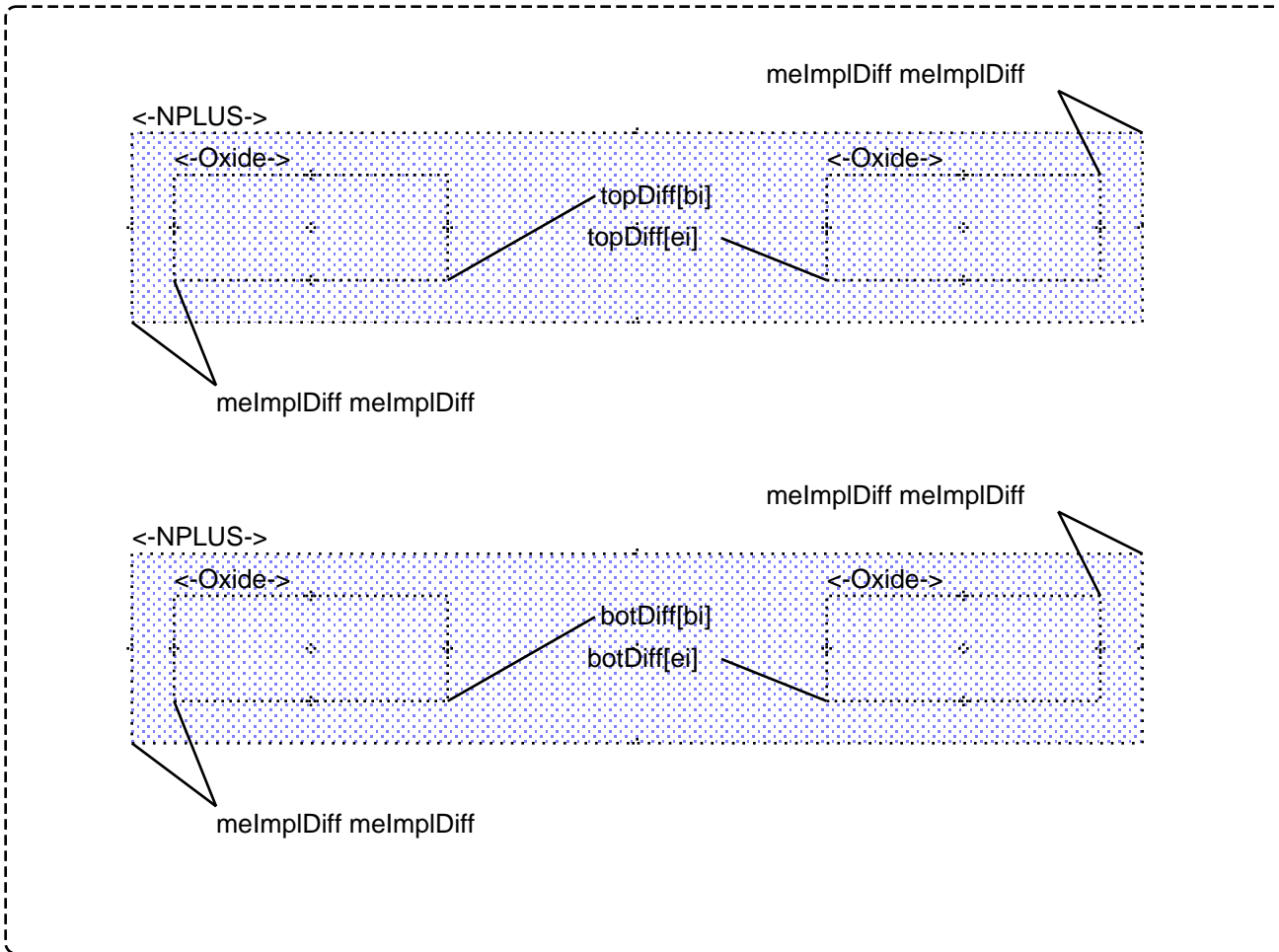


9: for i bi ei



pcell macro res\_nwell

10: include

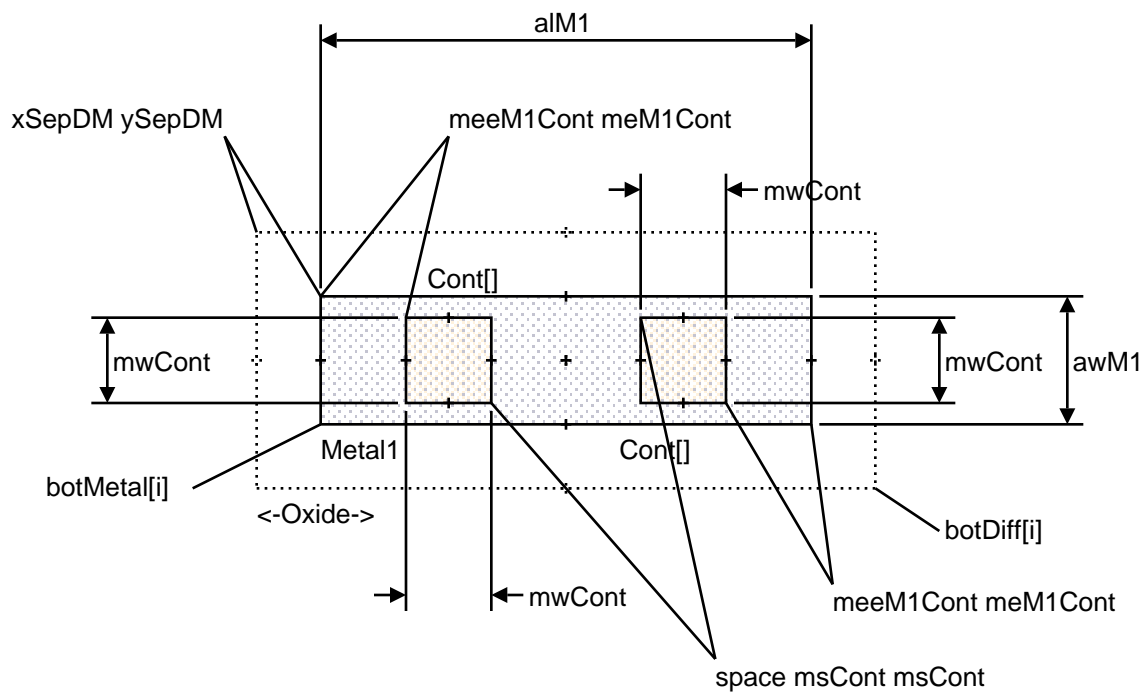
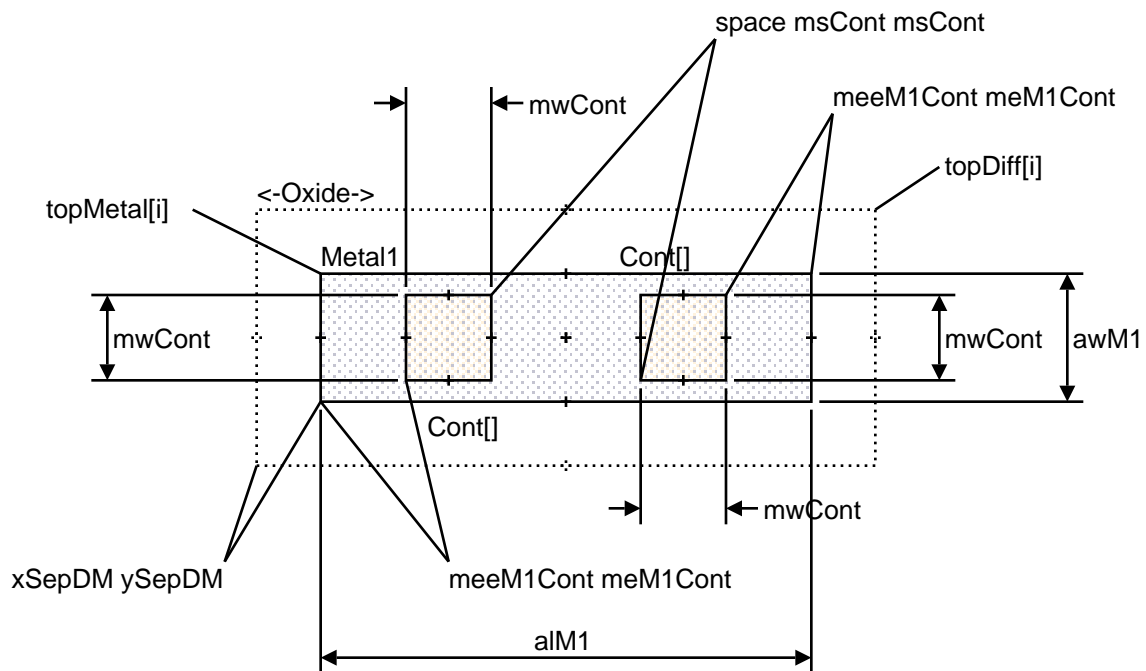


pcell macro res\_nwell

11: for i bi ei

if ((i>0 && leqp(i sc)) || (i==0 && cntOnDummy) || (i>sc && cntOnDummy))

contact



## res\_nwell\_id (macro)

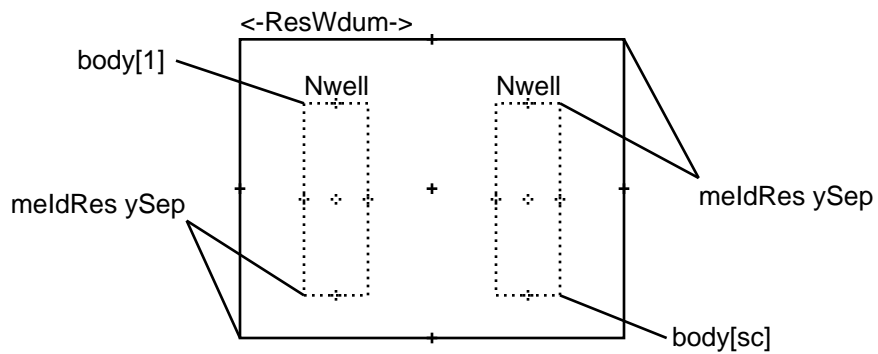
pcell macro res\_nwell\_id

2: include

```
meldRes = dpt->meldRes || 0
```

10: include

```
ySep = minus(idOffset)
```





# res\_connect (macro)

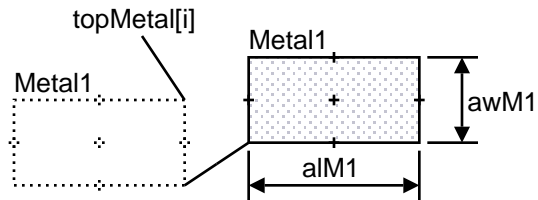
pcell macro res\_connect

15: if (connection != "Parallel")

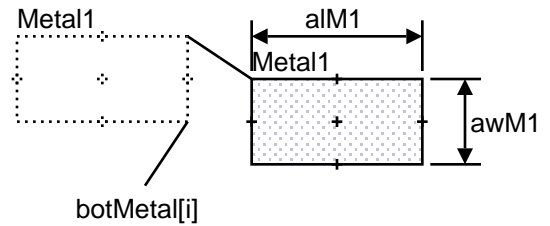
$$a1M1 = asHead + (2.0 * aeeResM1)$$

for i 1 (sc - 1)

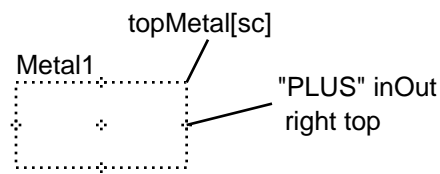
if oddp(i)



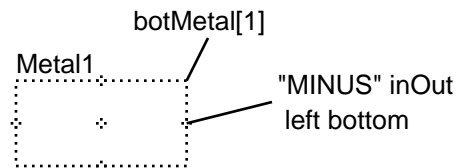
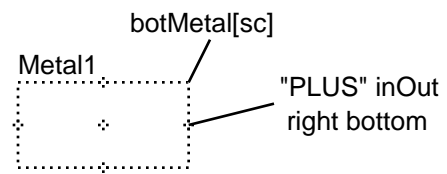
if evenp(i)



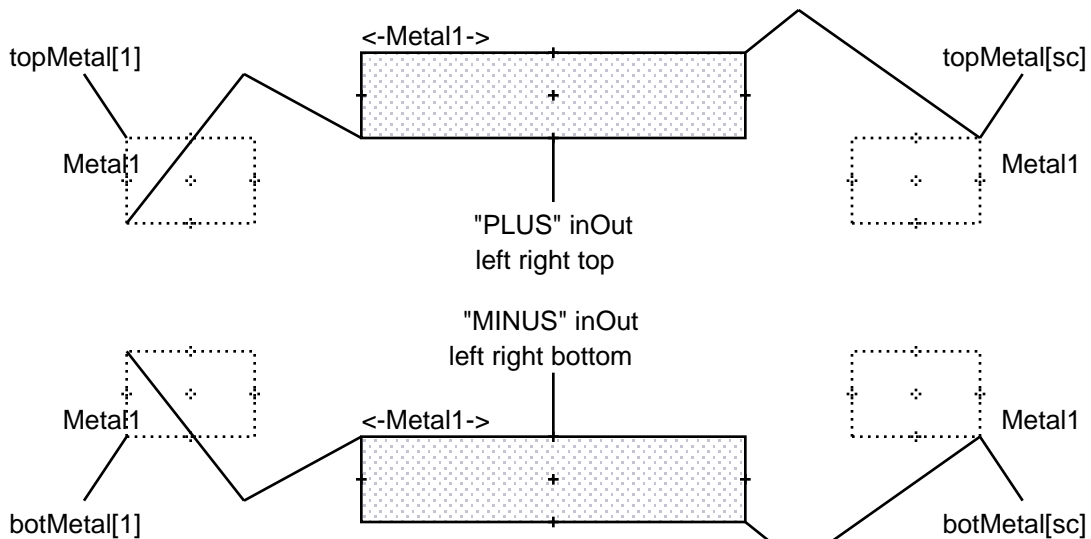
if oddp(sc)



if evenp(sc)



15: if (connection == "Parallel")



**res\_m (macro)**

pcell macro res\_m

1: include

```

dpt = PasGetDeviceProps( cv )
scale = dpt->scale
epsilon = grid / 10.0
mwRes = dpt->mwRes
mwCont = dpt->mwCont
msCont = dpt->msCont
meResCont = dpt->meResCont
meeResCont = dpt->meeResCont || meResCont
msldCont = dpt->msldCont || 0
msldRes = dpt->msldRes

msRes = dpt->msRes
mwM1 = dpt->mwM1
msM1 = dpt->msM1
meM1Cont = dpt->meM1Cont
meeM1Cont = dpt->meeM1Cont || meM1Cont

```

3: include

```

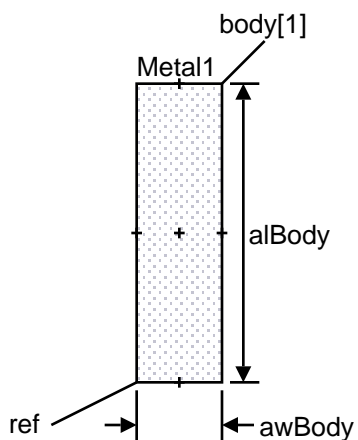
alBody = sl / scale
awBody = w / scale
sc = max(1 segments)
awHead = max( awBody mwRes (mwCont + (2.0 * meeResCont)) )
cntSpan = ( max(1 cntRows ) * ( mwCont + msCont ) ) - msCont
alHead = max( mwRes (cntSpan + (2.0 * meResCont)) )
asHead = msRes
asBody = asHead + awHead - awBody
dogbone = awBody + epsilon < awHead
xSepHB = if( dogbone then PasCeiling( ( awHead - awBody ) / 2.0 ) grid t ) else 0.0 )
bi = if( leftDummy then 0 else 1 )
ei = if( rightDummy then sc + 1 else sc )
xSepHM = meeResCont - meeM1Cont
ySepHM = meResCont - meM1Cont
alM1 = awHead - (2.0 * xSepHM)
awM1 = cntSpan + (2.0 * meM1Cont)
idOffset = msldCont - meResCont
idOffset = if( dogbone && msldRes then max( idOffset msldRes )
              else idOffset )
alBody = alBody + (2.0 * idOffset)
aeeResM1 = xSepHM

```

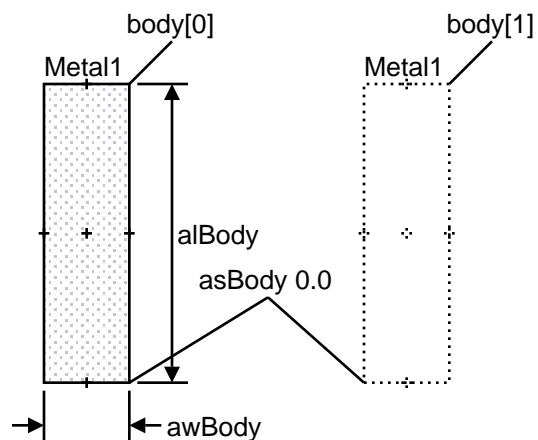
# res\_m1 (macro)

pcell macro res\_m1

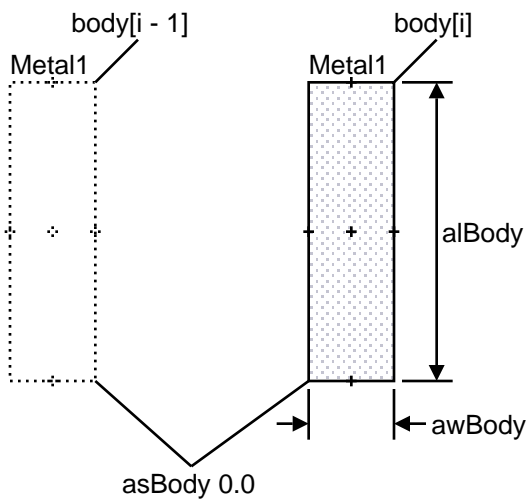
5: include



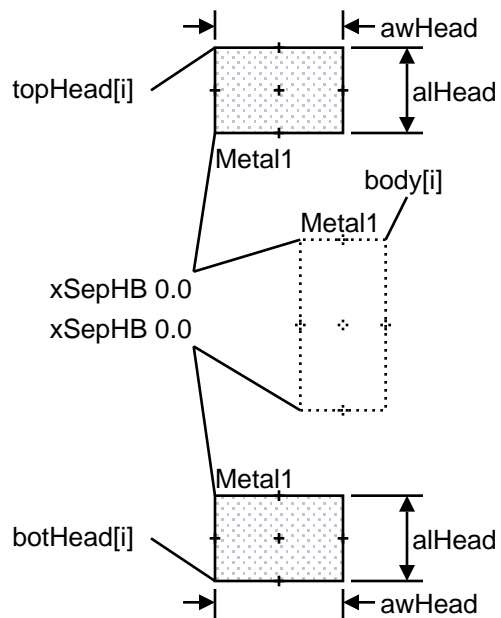
6: if leftDummy



7: for i 2 ei



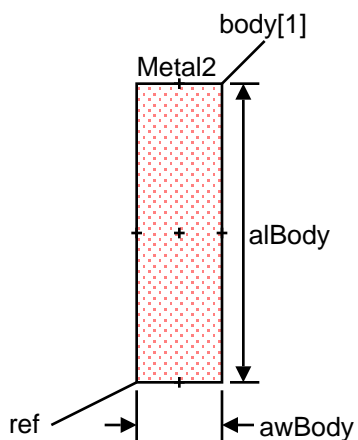
8: for i bi ei



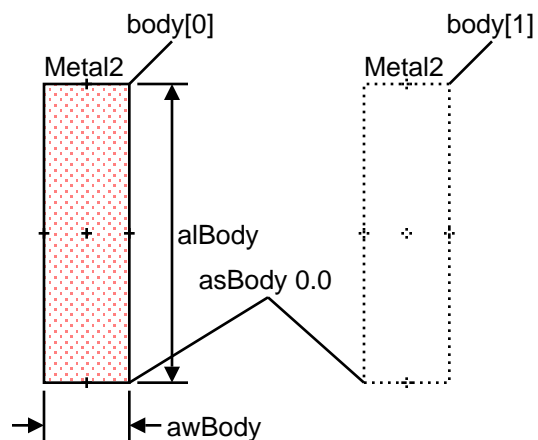
# res\_m2 (macro)

pcell macro res\_m2

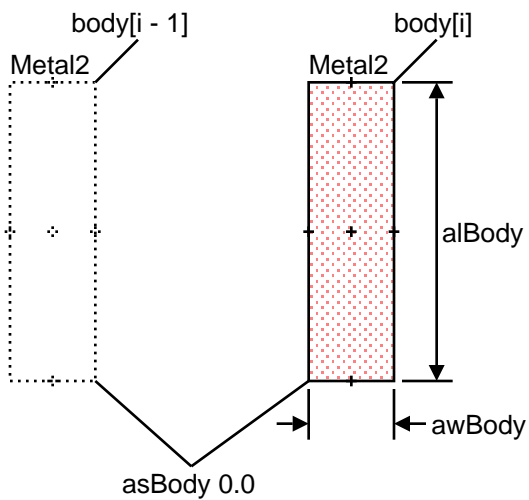
5: include



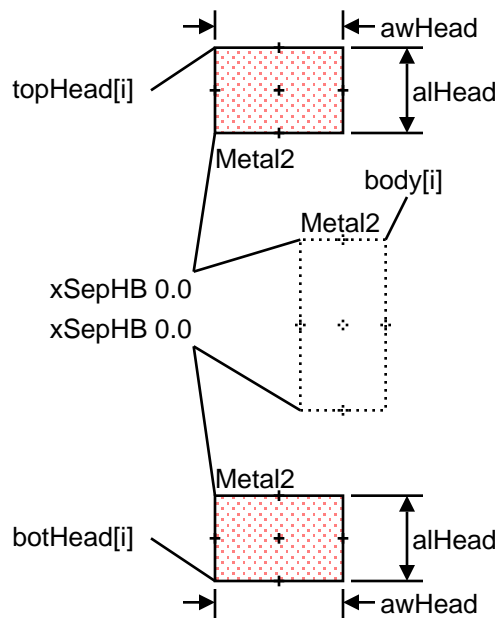
6: if leftDummy



7: for i 2 ei



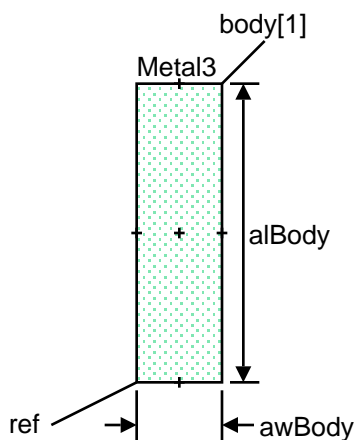
8: for i bi ei



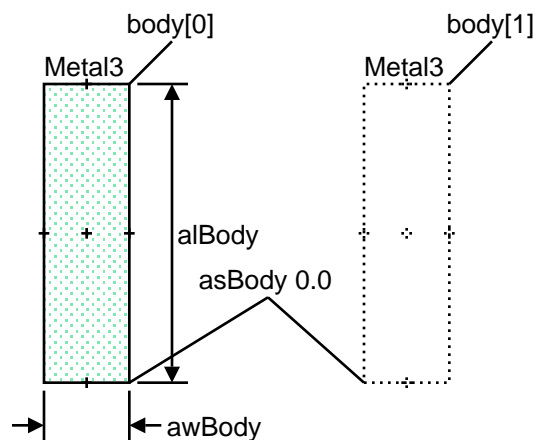
# res\_m3 (macro)

pcell macro res\_m3

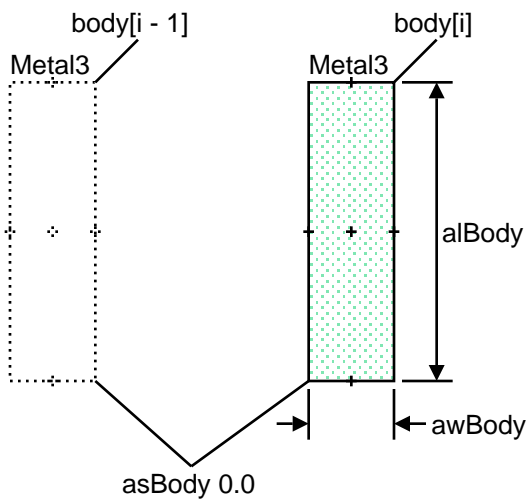
5: include



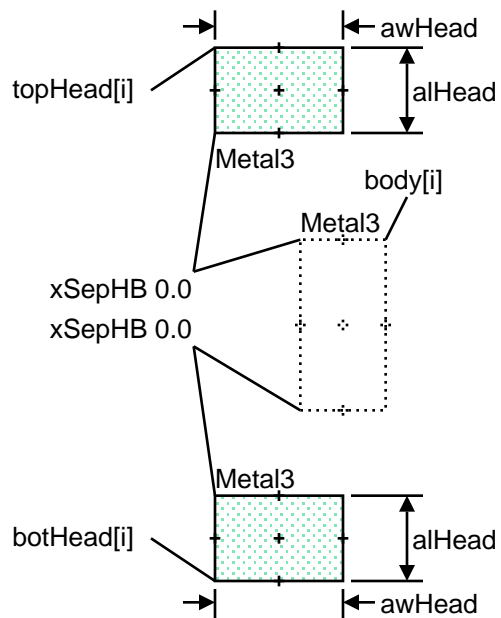
6: if leftDummy



7: for i 2 ei



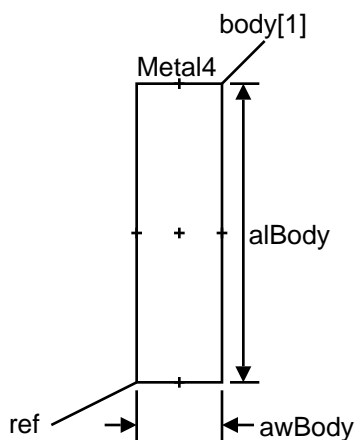
8: for i bi ei



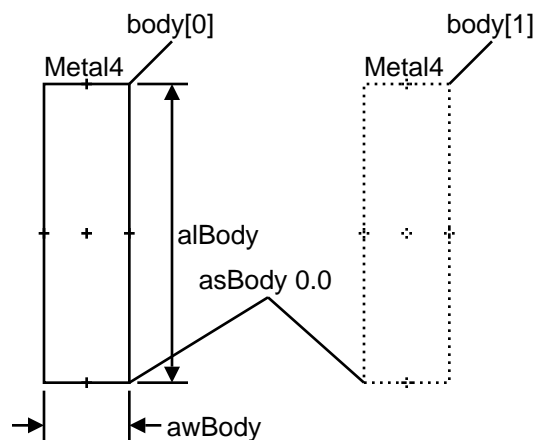
# res\_m4 (macro)

pcell macro res\_m4

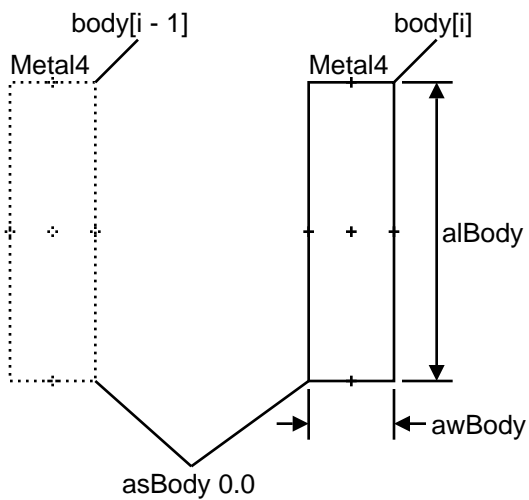
5: include



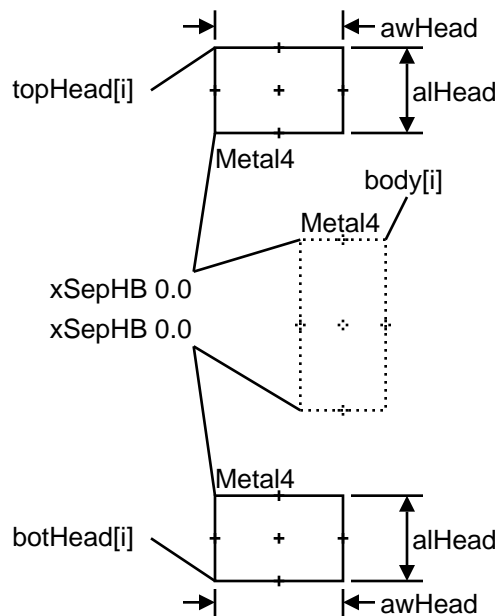
6: if leftDummy



7: for i 2 ei



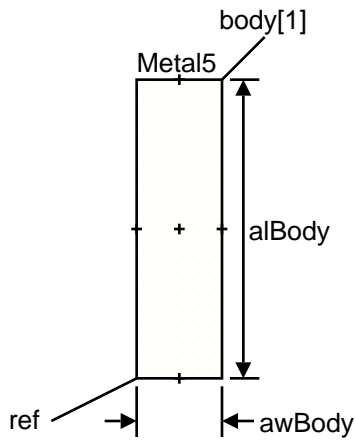
8: for i bi ei



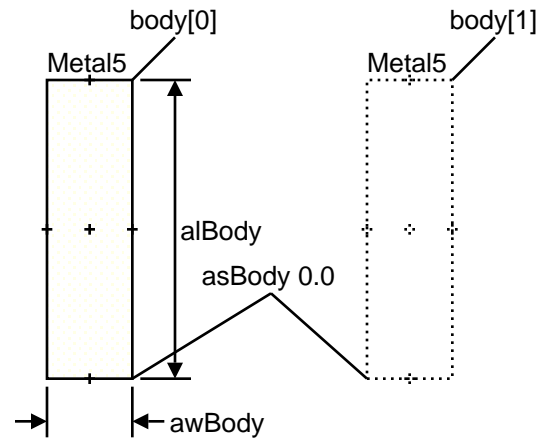
# res\_m5 (macro)

pcell macro res\_m5

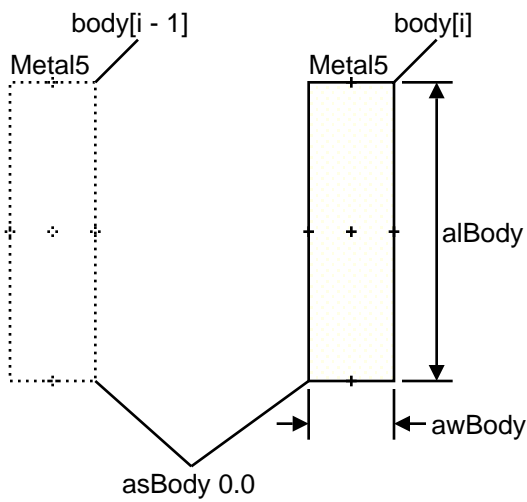
5: include



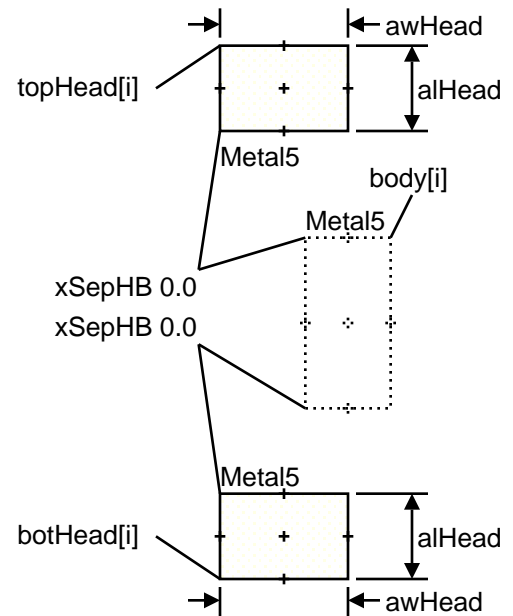
6: if leftDummy



7: for i 2 ei



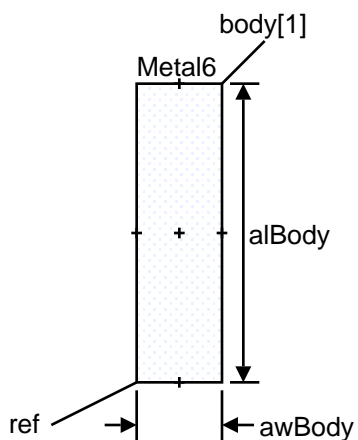
8: for i bi ei



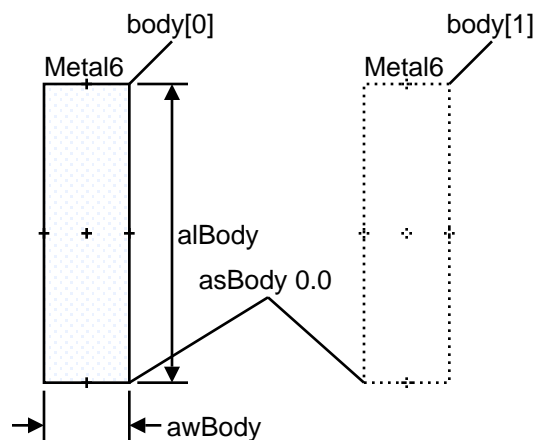
# res\_m6 (macro)

pcell macro res\_m6

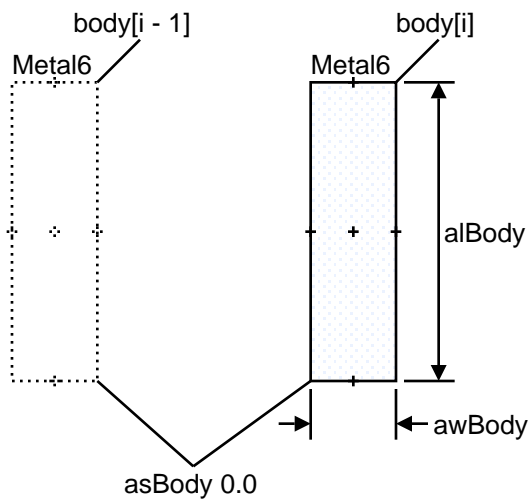
5: include



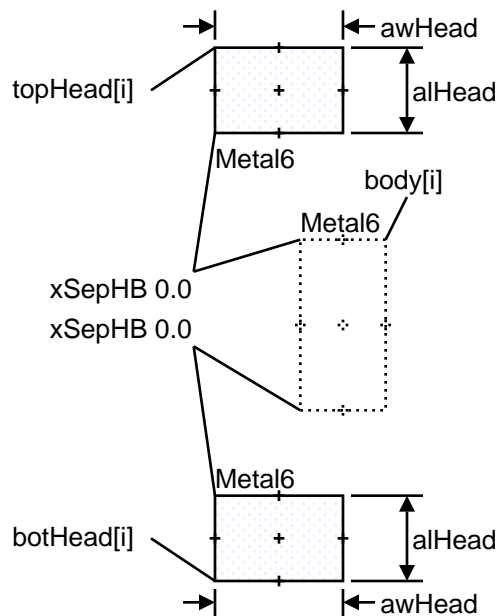
6: if leftDummy



7: for i 2 ei



8: for i bi ei





## res\_m1\_id (macro)

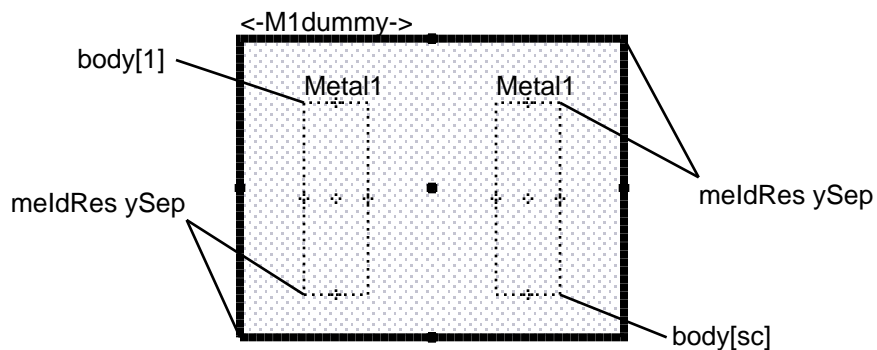
pcell macro res\_m1\_id

2: include

```
meldRes = dpt->meldRes || 0
```

10: include

```
ySep = minus(idOffset)
```



## res\_m2\_id (macro)

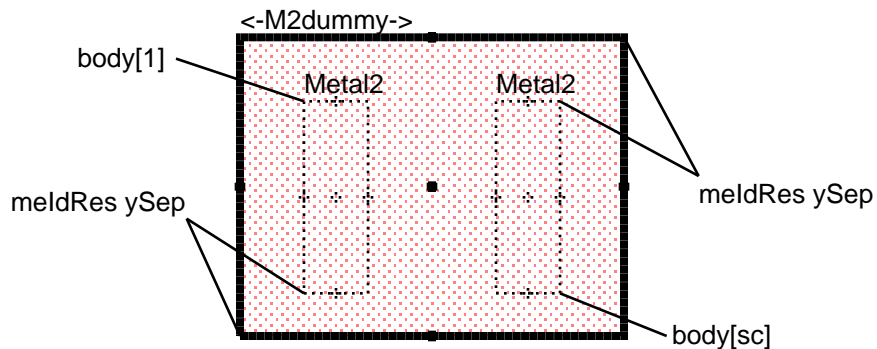
pcell macro res\_m2\_id

2: include

```
meldRes = dpt->meldRes || 0
```

10: include

```
ySep = minus(idOffset)
```



## res\_m3\_id (macro)

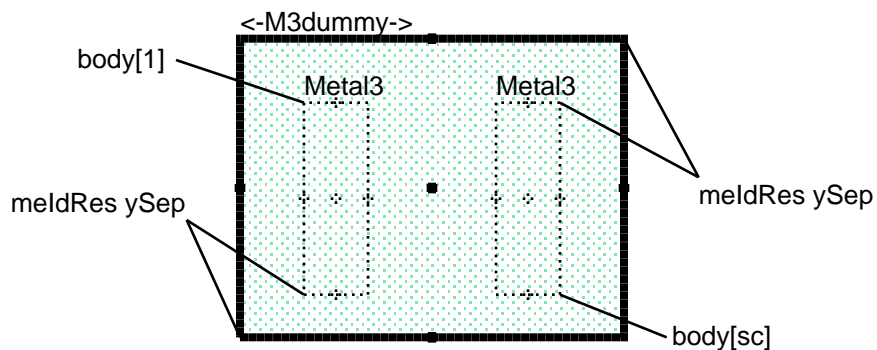
pcell macro res\_m3\_id

2: include

```
meldRes = dpt->meldRes || 0
```

10: include

```
ySep = minus(idOffset)
```



## res\_m4\_id (macro)

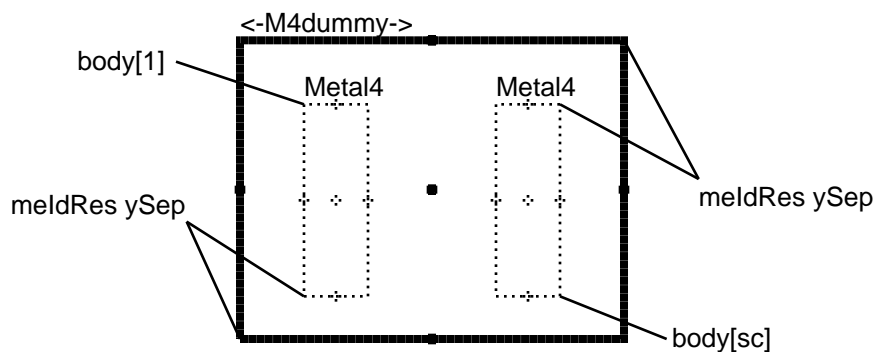
pcell macro res\_m4\_id

2: include

```
meldRes = dpt->meldRes || 0
```

10: include

```
ySep = minus(idOffset)
```



## res\_m5\_id (macro)

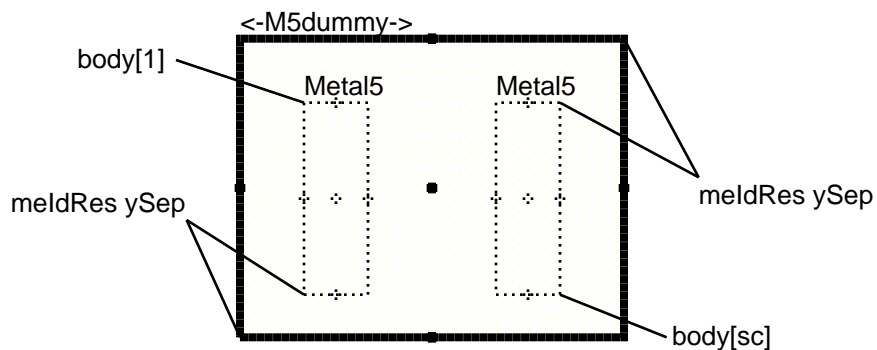
pcell macro res\_m5\_id

2: include

```
meldRes = dpt->meldRes || 0
```

10: include

```
ySep = minus(idOffset)
```



## res\_m6\_id (macro)

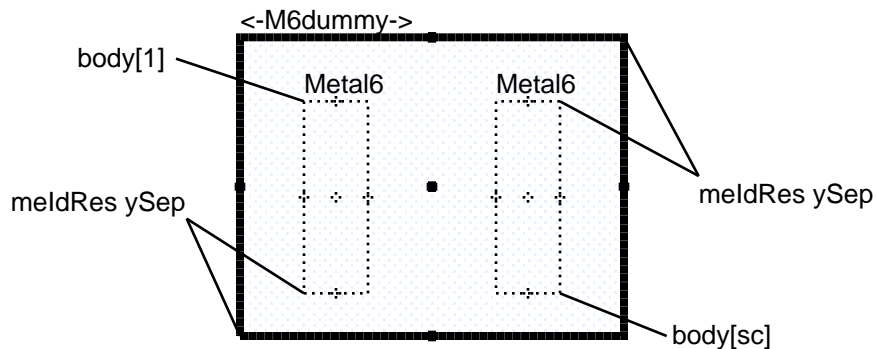
pcell macro res\_m6\_id

2: include

```
meldRes = dpt->meldRes || 0
```

10: include

```
ySep = minus(idOffset)
```



# m1res\_connect (macro)

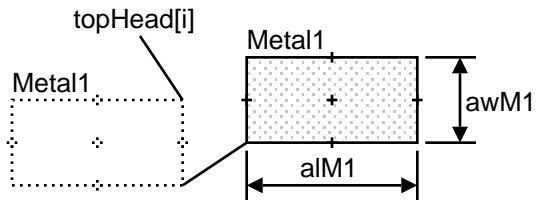
pcell macro m1res\_connect

15: if (connection != "Parallel")

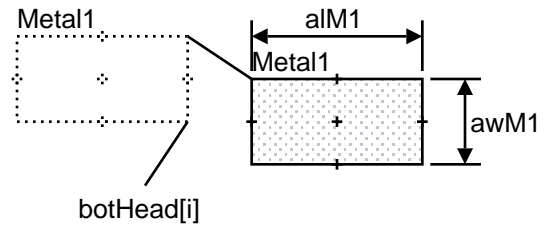
$$a1M1 = asHead + (2.0 * aeeResM1)$$

for i 1 (sc - 1)

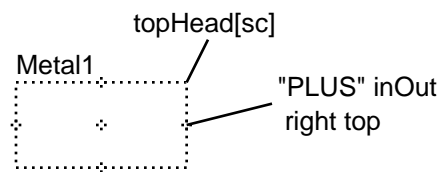
if oddp(i)



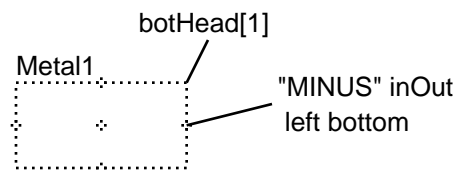
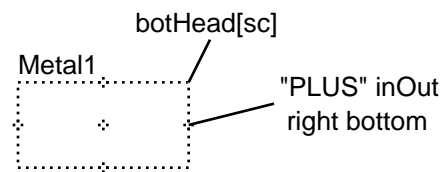
if evenp(i)



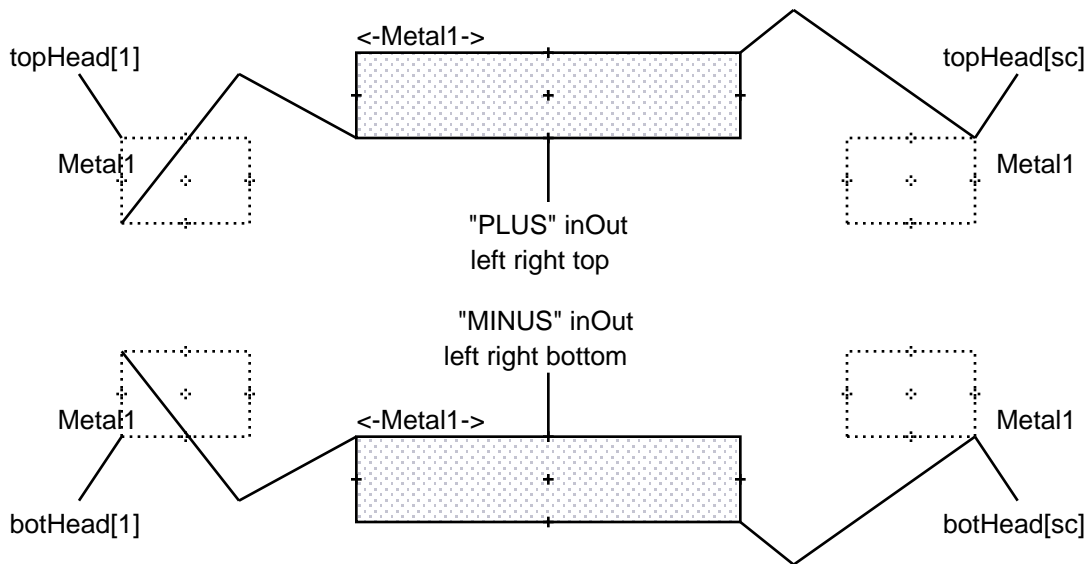
if oddp(sc)



if evenp(sc)



15: if (connection == "Parallel")



# m2res\_connect (macro)

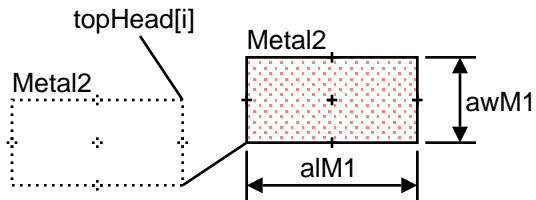
pcell macro m2res\_connect

15: if (connection != "Parallel")

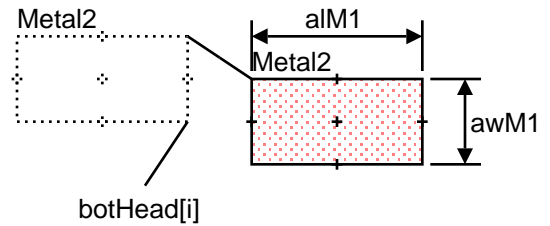
$$aIM1 = asHead + (2.0 * aeeResM1)$$

for i 1 (sc - 1)

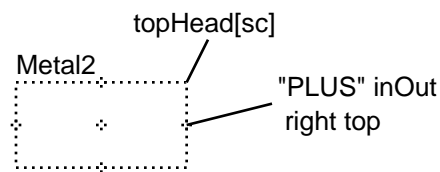
if oddp(i)



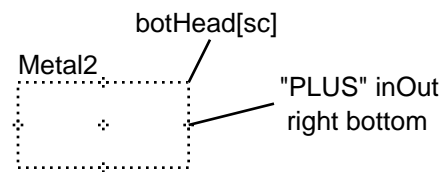
if evenp(i)



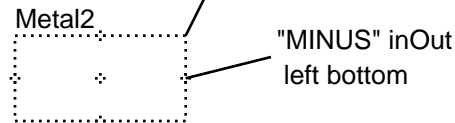
if oddp(sc)



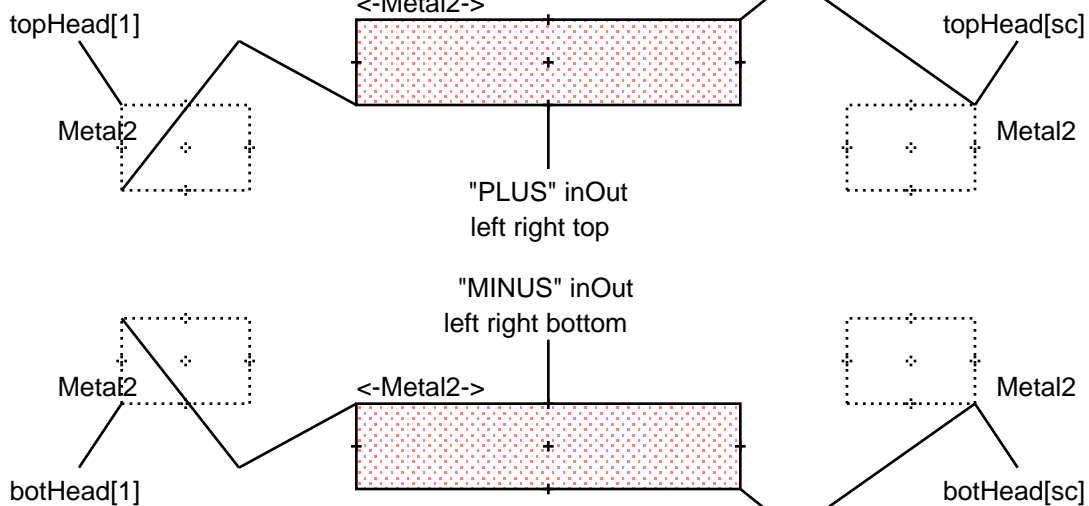
if evenp(sc)



botHead[1]



15: if (connection == "Parallel")



# m3res\_connect (macro)

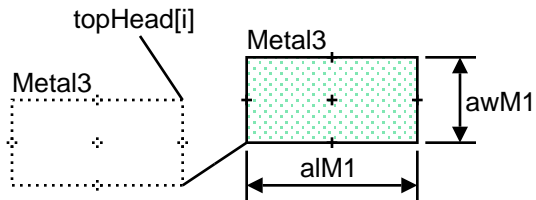
pcell macro m3res\_connect

15: if (connection != "Parallel")

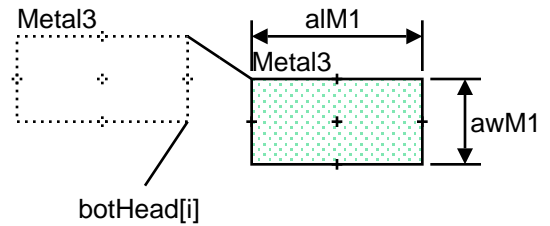
$$aIM1 = asHead + (2.0 * aeeResM1)$$

for i 1 (sc - 1)

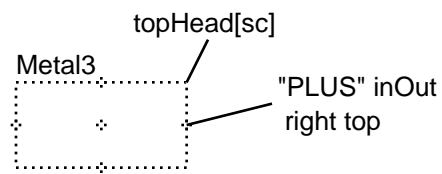
if oddp(i)



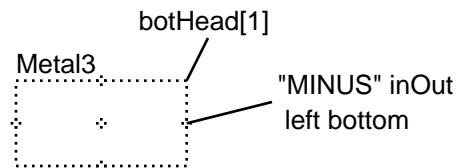
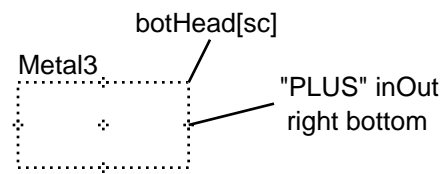
if evenp(i)



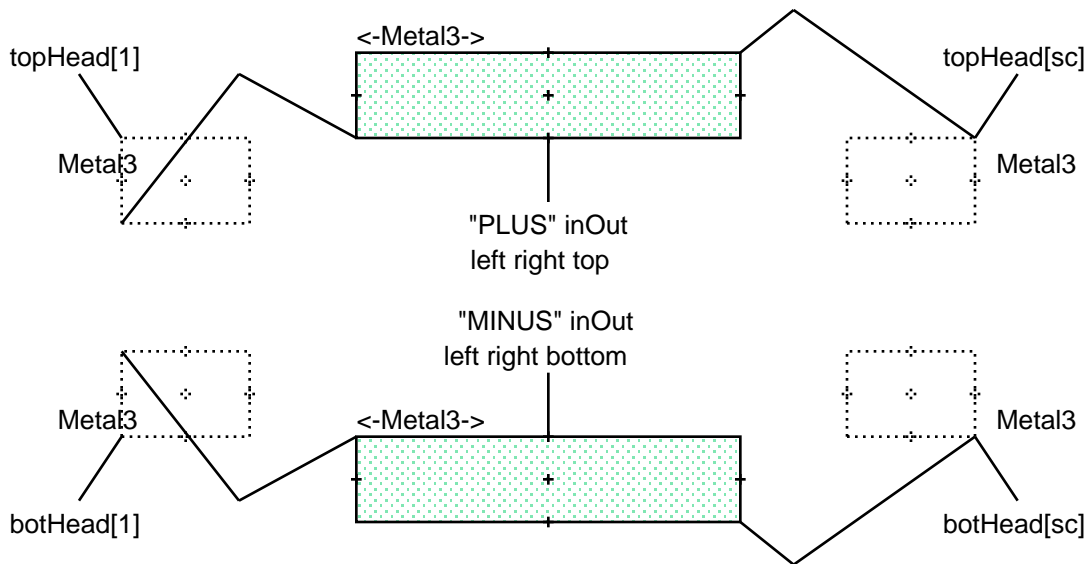
if oddp(sc)



if evenp(sc)



15: if (connection == "Parallel")



# m4res\_connect (macro)

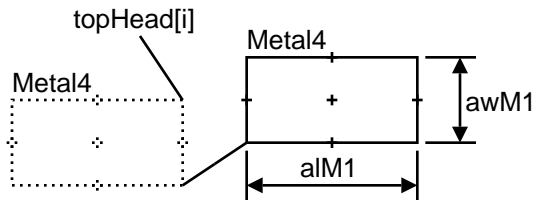
pcell macro m4res\_connect

15: if (connection != "Parallel")

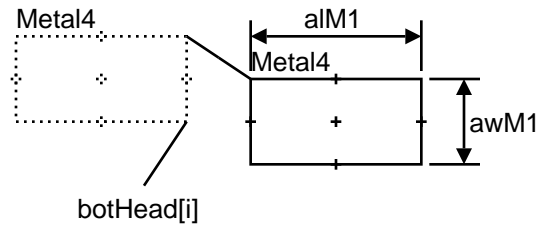
$$aIM1 = asHead + (2.0 * aeeResM1)$$

for i 1 (sc - 1)

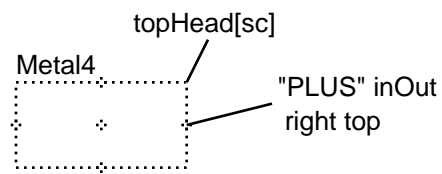
if oddp(i)



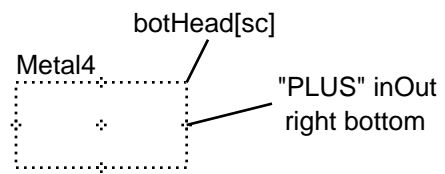
if evenp(i)



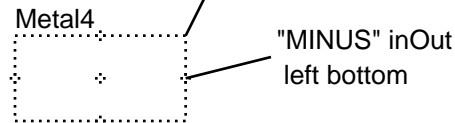
if oddp(sc)



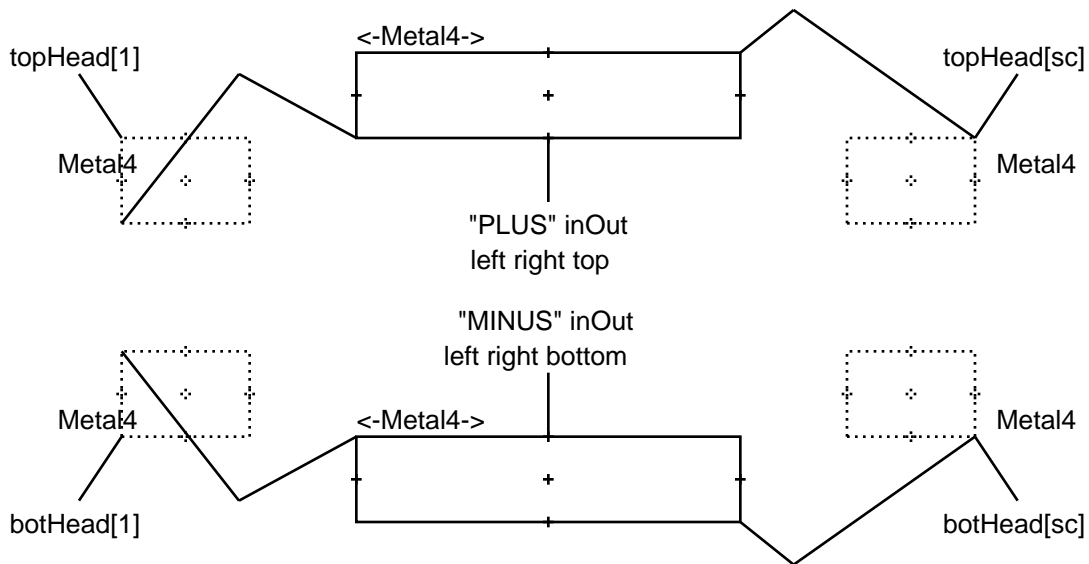
if evenp(sc)



botHead[1]



15: if (connection == "Parallel")



# m5res\_connect (macro)

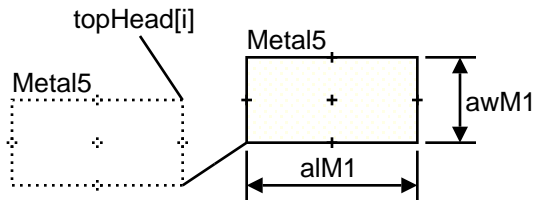
pcell macro m5res\_connect

15: if (connection != "Parallel")

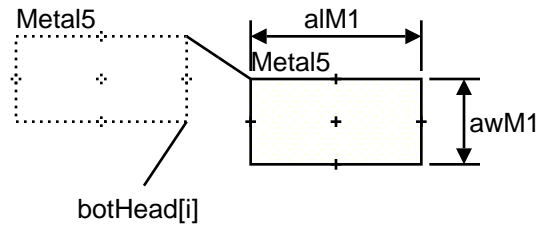
$$a1M1 = asHead + (2.0 * aeeResM1)$$

for i 1 (sc - 1)

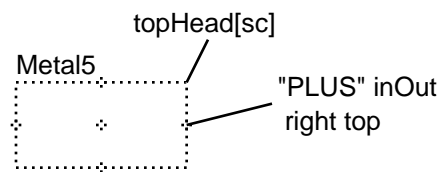
if oddp(i)



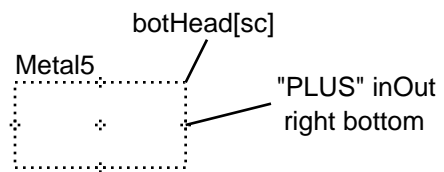
if evenp(i)



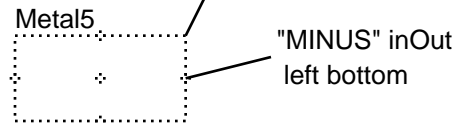
if oddp(sc)



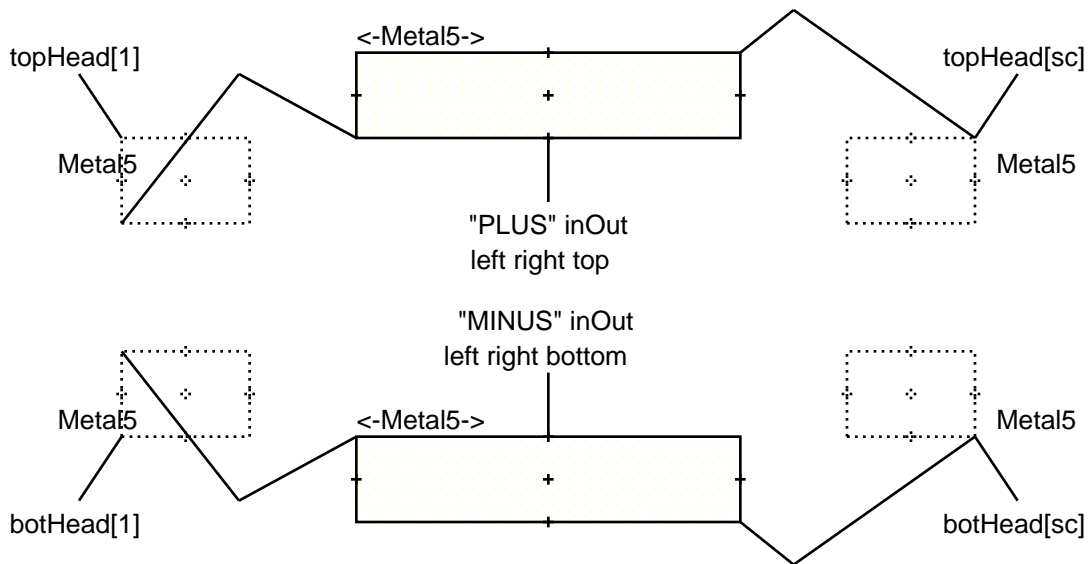
if evenp(sc)



botHead[1]



15: if (connection == "Parallel")





# m6res\_connect (macro)

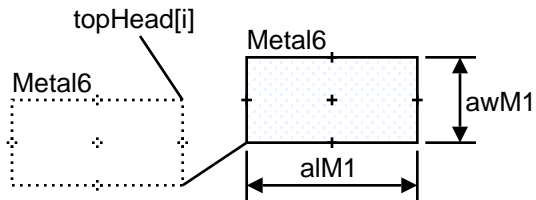
pcell macro m6res\_connect

15: if (connection != "Parallel")

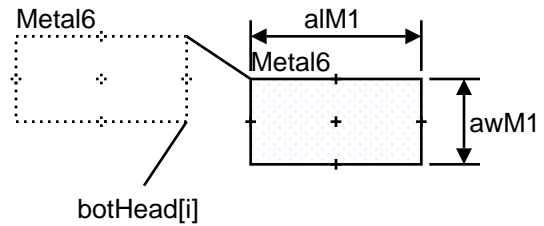
$$aIM1 = asHead + (2.0 * aeeResM1)$$

for i 1 (sc - 1)

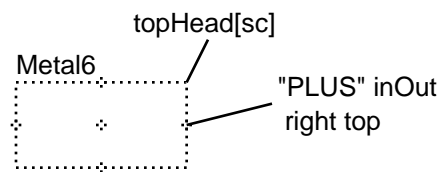
if oddp(i)



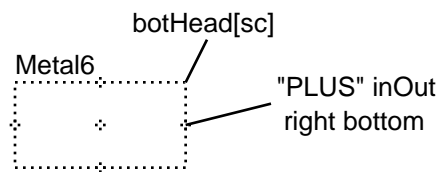
if evenp(i)



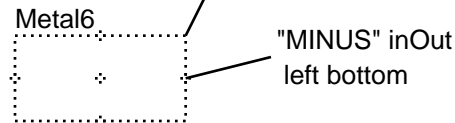
if oddp(sc)



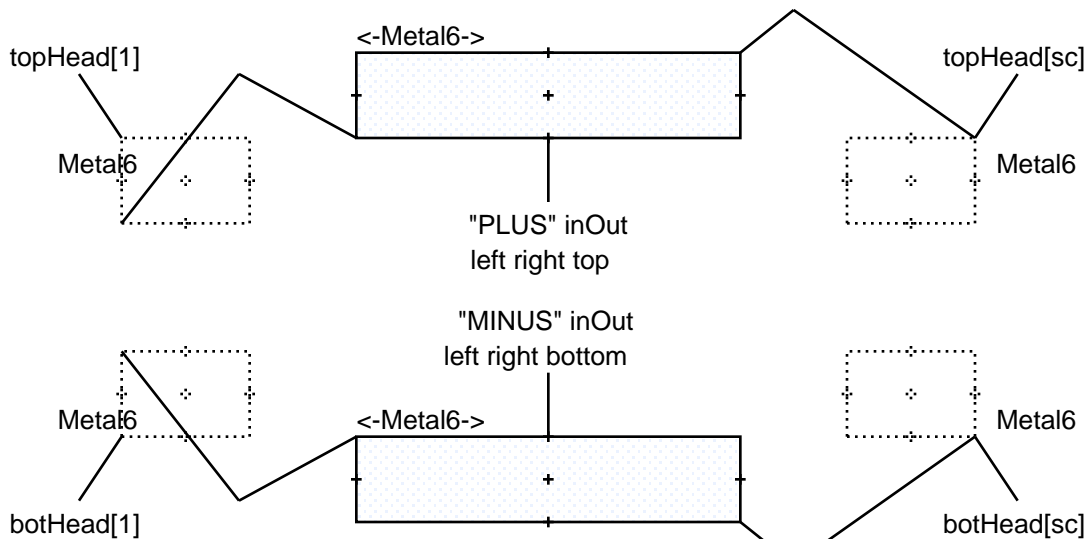
if evenp(sc)



botHead[1]



15: if (connection == "Parallel")



## PCell Devices

### Poly Resistor

pcell gpdk180 polyres

```
include macro res_multiSeg
include macro res_poly
include macro res_poly_id
```

### High Poly Resistor

pcell gpdk180 polyhres

```
include macro res_multiSeg
include macro res_poly
include macro res_poly_id
include macro res_poly_sb
```

### N+ Diff Resistor

pcell gpdk180 nplusres

```
include macro res_multiSeg
include macro res_diff
include macro res_diff_nplus
include macro res_diff_id
```

### P+ Diff Resistor

pcell gpdk180 pplusres

```
include macro res_multiSeg
include macro res_diff
include macro res_diff_pplus
include macro res_diff_id
include macro res_diff_nw
```

### NWell Resistor

pcell gpdk180 nwellres

```
include macro res_multiSeg
include macro res_nwell
include macro res_nwell_id
```

### Metal Resistor

pcell gpdk180 mxres

```
include macro mres_multiSeg
include macro $mac0
include macro res_m
include macro $mac1
include macro $mac2
```

Pcell Macro Table

\$cell	\$mac0	\$mac1	\$mac2
m1res	m1res_connect	res_m1	res_m1_id
m2res	m2res_connect	res_m2	res_m2_id
m3res	m3res_connect	res_m3	res_m3_id
m4res	m4res_connect	res_m4	res_m4_id
m5res	m5res_connect	res_m5	res_m5_id
m6res	m6res_connect	res_m6	res_m6_id

## **Library CAP Definitions**

## CDF Definitions

## mimcap

CDF gpd180 mimcap

CDF device params

<b>grid</b>	\$grid	private	mfg grid
<b>scale</b>	1.0e-6	private	dimensional scale factor
<b>model</b>	"mimcap"	private	device model name
<b>category</b>	"cap"	private	dfl library manager category

include macro cap

CDF device params

<b>mwCap</b>	4.0	private	min CAP width
<b>xwCap</b>	30.0	private	max CAP width
<b>dwCap</b>	4.0	private	default CAP width
<b>mlCap</b>	4.0	private	min CAP length
<b>xlCap</b>	30.0	private	max CAP length
<b>dlCap</b>	4.0	private	default CAP length
<b>ca</b>	0.001	private	area capacitance
<b>cf</b>	1.0e-10	private	fringe capacitance

CDF siminfo

<b>simulator</b>	auCdl
<b>componentName</b>	"\$cell"
<b>instParameters</b>	(AREA M)
<b>propMapping</b>	(nil AREA area M m)
<b>modelName</b>	"\$cell"

CDF siminfo

<b>simulator</b>	auLvs
<b>componentName</b>	"\$cell"
<b>instParameters</b>	(area m)

CDF siminfo

<b>simulator</b>	spectre
<b>instParameters</b>	(area perim m scale trise ic tc1 tc2)
<b>componentName</b>	"\$cell"

CDF cellview

<b>symbol</b>	pas_std	cap2	symbol	
<b>spectre</b>	pas_std	cap2	symbol	
<b>auCdl</b>	pas_std	cap2	symbol	
<b>auLvs</b>	pas_std	cap2	symbol	
<b>ads</b>	pas_std	cap2	symbol	
<b>ivpcell</b>	\$lib	\$cell	symbol	5

## CDF Macro Definitions

## Capacitor

CDF macro cap

CDF parameters

<b>name</b>	"model"
<b>prompt</b>	"Model name"
<b>defValue</b>	gpdK180_capValue('model ?id cdfId ?returnString t)
<b>type</b>	"string"
<b>display</b>	"gpdK180_capDisplay('model)"
<b>editable</b>	"nil"
<b>parseAsCEL</b>	"yes"

CDF parameters

<b>name</b>	"calcParam"
<b>prompt</b>	"Calculate Parameter"
<b>type</b>	"cyclic"
<b>choices</b>	list("capacitance" "length" "width")
<b>defValue</b>	"capacitance"
<b>display</b>	"t"

CDF parameters

<b>name</b>	"c"
<b>prompt</b>	"Capacitance"
<b>units</b>	"capacitance"
<b>defValue</b>	gpdK180_capValue('c ?id cdfId ?returnString t)
<b>type</b>	"string"
<b>display</b>	"t"
<b>editable</b>	"cdfgData->calcParam->value!=\"capacitance\""
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"
<b>callback</b>	"gpdK180_capCB('c)"

CDF parameters

<b>name</b>	"w"
<b>prompt</b>	"Width"
<b>units</b>	"lengthMetric"
<b>defValue</b>	gpdK180_capValue('w ?id cdfId ?returnString t)
<b>type</b>	"string"
<b>display</b>	"t"
<b>editable</b>	"cdfgData->calcParam->value!=\"width\""
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"
<b>callback</b>	"gpdK180_capCB('w)"

## CDF macro cap

## CDF parameters

<b>name</b>	"l"
<b>prompt</b>	"Length"
<b>units</b>	"lengthMetric"
<b>defValue</b>	gpdK180_capValue('l' ?id cdfld ?returnString t)
<b>type</b>	"string"
<b>display</b>	"t"
<b>editable</b>	"cdfgData->calcParam->value!=\"length\""
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"
<b>callback</b>	"gpdK180_capCB('l')"

## CDF parameters

<b>name</b>	"m"
<b>prompt</b>	"Multiplier"
<b>defValue</b>	"1"
<b>type</b>	"string"
<b>display</b>	"t"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"showSimParams"
<b>prompt</b>	"Show Sim Parameters"
<b>defValue</b>	nil
<b>type</b>	"boolean"
<b>display</b>	"gpdK180_capDisplay('showSimParams')"

## CDF parameters

<b>name</b>	"ca"
<b>prompt</b>	"Area capacitance (F/M^2)"
<b>defValue</b>	gpdK180_capValue('ca' ?id cdfld)
<b>type</b>	"float"
<b>display</b>	"gpdK180_capDisplay('ca')"
<b>editable</b>	"nil"

## CDF parameters

<b>name</b>	"cf"
<b>prompt</b>	"Fringe capacitance (F/M)"
<b>defValue</b>	gpdK180_capValue('cf' ?id cdfld)
<b>type</b>	"float"
<b>display</b>	"gpdK180_capDisplay('cf')"
<b>editable</b>	"nil"

## CDF parameters

<b>name</b>	"area"
<b>prompt</b>	"Area"
<b>defValue</b>	"iPar(\"w\")*iPar(\"l\")"
<b>type</b>	"string"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"
<b>editable</b>	"nil"

## CDF parameters

<b>name</b>	"perim"
<b>prompt</b>	"Perim"
<b>defValue</b>	"2*iPar(\"w\")+2*iPar(\"l\")"
<b>type</b>	"string"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"
<b>editable</b>	"nil"

## CDF macro cap

CDF parameters

<b>name</b>	"ic"
<b>prompt</b>	"Initial condition"
<b>units</b>	"voltage"
<b>defValue</b>	""
<b>type</b>	"string"
<b>display</b>	"gpd180_capDisplay('ic')"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

CDF parameters

<b>name</b>	"scale"
<b>prompt</b>	"Scale factor"
<b>defValue</b>	""
<b>type</b>	"string"
<b>display</b>	"gpd180_capDisplay('scale')"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

CDF parameters

<b>name</b>	"trise"
<b>prompt</b>	"Temp rise from ambient"
<b>defValue</b>	""
<b>type</b>	"string"
<b>display</b>	"gpd180_capDisplay('trise')"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

CDF parameters

<b>name</b>	"tc1"
<b>prompt</b>	"Temperature coefficient 1"
<b>defValue</b>	""
<b>type</b>	"string"
<b>display</b>	"gpd180_capDisplay('tc1')"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

CDF parameters

<b>name</b>	"tc2"
<b>prompt</b>	"Temperature coefficient 2"
<b>defValue</b>	""
<b>type</b>	"string"
<b>display</b>	"gpd180_capDisplay('tc2')"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

CDF parameters

<b>name</b>	"dtemp"
<b>prompt</b>	"Temperature difference"
<b>defValue</b>	""
<b>type</b>	"string"
<b>display</b>	"gpd180_capDisplay('dtemp')"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

CDF parameters

<b>name</b>	"polyCoef"
<b>prompt</b>	"Number of Polynomial Coeffs"
<b>defValue</b>	0
<b>type</b>	"int"
<b>display</b>	"gpd180_capDisplay('polyCoef')"

## CDF macro cap

## CDF parameters

<b>name</b>	"c1"
<b>prompt</b>	"Poly Coeff 1"
<b>defValue</b>	""
<b>type</b>	"string"
<b>use</b>	"cdfgData->polyCoef->value >= 1"
<b>display</b>	"gpd180_capDisplay('c1')"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"c2"
<b>prompt</b>	"Poly Coeff 2"
<b>defValue</b>	""
<b>type</b>	"string"
<b>use</b>	"cdfgData->polyCoef->value >= 2"
<b>display</b>	"gpd180_capDisplay('c2')"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"c3"
<b>prompt</b>	"Poly Coeff 3"
<b>defValue</b>	""
<b>type</b>	"string"
<b>use</b>	"cdfgData->polyCoef->value >= 3"
<b>display</b>	"gpd180_capDisplay('c3')"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"



## CDF macro cap

## CDF parameters

<b>name</b>	"c4"
<b>prompt</b>	"Poly Coeff 4"
<b>defValue</b>	""
<b>type</b>	"string"
<b>use</b>	"cdfgData->polyCoef->value >= 4"
<b>display</b>	"gpd180_capDisplay('c4')"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"c5"
<b>prompt</b>	"Poly Coeff 5"
<b>defValue</b>	""
<b>type</b>	"string"
<b>use</b>	"cdfgData->polyCoef->value >= 5"
<b>display</b>	"gpd180_capDisplay('c5')"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"c6"
<b>prompt</b>	"Poly Coeff 6"
<b>defValue</b>	""
<b>type</b>	"string"
<b>use</b>	"cdfgData->polyCoef->value >= 6"
<b>display</b>	"gpd180_capDisplay('c6')"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF macro cap

## CDF parameters

<b>name</b>	"c7"
<b>prompt</b>	"Poly Coeff 7"
<b>defValue</b>	""
<b>type</b>	"string"
<b>use</b>	"cdfgData->polyCoef->value >= 7"
<b>display</b>	"gpd180_capDisplay('c7')"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"c8"
<b>prompt</b>	"Poly Coeff 8"
<b>defValue</b>	""
<b>type</b>	"string"
<b>use</b>	"cdfgData->polyCoef->value >= 8"
<b>display</b>	"gpd180_capDisplay('c8')"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"c9"
<b>prompt</b>	"Poly Coeff 9"
<b>defValue</b>	""
<b>type</b>	"string"
<b>use</b>	"cdfgData->polyCoef->value >= 9"
<b>display</b>	"gpd180_capDisplay('c9')"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF macro cap

## CDF parameters

<b>name</b>	"c10"
<b>prompt</b>	"Poly Coeff 10"
<b>defValue</b>	""
<b>type</b>	"string"
<b>use</b>	"cdfgData->polyCoef->value >= 10"
<b>display</b>	"gpd180_capDisplay('c10')"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"c11"
<b>prompt</b>	"Poly Coeff 11"
<b>defValue</b>	""
<b>type</b>	"string"
<b>use</b>	"cdfgData->polyCoef->value >= 11"
<b>display</b>	"gpd180_capDisplay('c11')"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"c12"
<b>prompt</b>	"Poly Coeff 12"
<b>defValue</b>	""
<b>type</b>	"string"
<b>use</b>	"cdfgData->polyCoef->value >= 12"
<b>display</b>	"gpd180_capDisplay('c12')"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF macro cap

## CDF parameters

<b>name</b>	"c13"
<b>prompt</b>	"Poly Coeff 13"
<b>defValue</b>	""
<b>type</b>	"string"
<b>use</b>	"cdfgData->polyCoef->value >= 13"
<b>display</b>	"gpd180_capDisplay('c13')"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"c14"
<b>prompt</b>	"Poly Coeff 14"
<b>defValue</b>	""
<b>type</b>	"string"
<b>use</b>	"cdfgData->polyCoef->value >= 14"
<b>display</b>	"gpd180_capDisplay('c14')"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"c15"
<b>prompt</b>	"Poly Coeff 15"
<b>defValue</b>	""
<b>type</b>	"string"
<b>use</b>	"cdfgData->polyCoef->value >= 15"
<b>display</b>	"gpd180_capDisplay('c15')"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF macro cap

## CDF parameters

<b>name</b>	"c16"
<b>prompt</b>	"Poly Coeff 16"
<b>defValue</b>	""
<b>type</b>	"string"
<b>use</b>	"cdfgData->polyCoef->value >= 16"
<b>display</b>	"gpd180_capDisplay('c16')"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"c17"
<b>prompt</b>	"Poly Coeff 17"
<b>defValue</b>	""
<b>type</b>	"string"
<b>use</b>	"cdfgData->polyCoef->value >= 17"
<b>display</b>	"gpd180_capDisplay('c17')"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"c18"
<b>prompt</b>	"Poly Coeff 18"
<b>defValue</b>	""
<b>type</b>	"string"
<b>use</b>	"cdfgData->polyCoef->value >= 18"
<b>display</b>	"gpd180_capDisplay('c18')"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF macro cap

## CDF parameters

<b>name</b>	"c19"
<b>prompt</b>	"Poly Coeff 19"
<b>defValue</b>	""
<b>type</b>	"string"
<b>use</b>	"cdfgData->polyCoef->value >= 19"
<b>display</b>	"gpd180_capDisplay('c19')"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"c20"
<b>prompt</b>	"Poly Coeff 20"
<b>defValue</b>	""
<b>type</b>	"string"
<b>use</b>	"cdfgData->polyCoef->value >= 20"
<b>display</b>	"gpd180_capDisplay('c20')"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"source"
<b>prompt</b>	"Source component"
<b>type</b>	"boolean"
<b>display</b>	"gpd180_capDisplay('source')"

## CDF parameters

<b>name</b>	"qmode"
<b>prompt</b>	"Quality type"
<b>defValue</b>	"ideal"
<b>choices</b>	'("ideal" "Constant Conductance" "Square Root" "Constant Q")'
<b>type</b>	"cyclic"
<b>display</b>	"gpd180_capDisplay('qmode')"

## CDF parameters

<b>name</b>	"q"
<b>prompt</b>	"Quality"
<b>defValue</b>	""
<b>type</b>	"string"
<b>display</b>	"gpd180_capDisplay('q')"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF macro cap

## CDF parameters

<b>name</b>	"freq"
<b>prompt</b>	"Frequency"
<b>units</b>	"frequency"
<b>defValue</b>	""
<b>type</b>	"string"
<b>display</b>	"gpd180_capDisplay('freq')"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"pasUpdateParamList"
<b>prompt</b>	"CDF Param Update List"
<b>defValue</b>	"c w l"
<b>type</b>	"string"
<b>display</b>	"nil"

## CDF siminfo

<b>simulator</b>	auCdl
<b>netlistProcedure</b>	ansCdlCompPrim
<b>instParameters</b>	(C M)
<b>termOrder</b>	(PLUS MINUS)
<b>propMapping</b>	(nil C c M m)
<b>namePrefix</b>	"C"

## CDF siminfo

<b>simulator</b>	auLvs
<b>propMapping</b>	nil
<b>netlistProcedure</b>	ansLvsCompPrim
<b>instParameters</b>	(c m)
<b>termOrder</b>	(PLUS MINUS)
<b>permuteRule</b>	"(p PLUS MINUS)"
<b>namePrefix</b>	"C"

## CDF siminfo

<b>simulator</b>	ams
<b>instParameters</b>	(c w l m scale trise ic tc1 tc2)
<b>termOrder</b>	(PLUS MINUS)
<b>isPrimitive</b>	t

## CDF macro cap

## CDF siminfo

<b>simulator</b>	spectre
<b>propMapping</b>	nil
<b>namePrefix</b>	"C"
<b>otherParameters</b>	(model)
<b>instParameters</b>	(c w l m scale trise ic tc1 tc2)
<b>termOrder</b>	(PLUS MINUS)
<b>termMapping</b>	(nil PLUS \:1 MINUS "(FUNCTION minus(root(\"PLUS\")))")
<b>componentName</b>	capacitor

## CDF siminfo

<b>simulator</b>	ads
<b>netlistProcedure</b>	ADSSimCompPrim
<b>otherParameters</b>	(model)
<b>instParameters</b>	(Area Periph _M scale Trise ic tc1 tc2)
<b>componentName</b>	nil
<b>termOrder</b>	(PLUS MINUS)
<b>termMapping</b>	(nil PLUS ":P1" MINUS ":minus.P1")
<b>propMapping</b>	(nil Area area Periph perim _M m Trise trise)
<b>typeMapping</b>	nil
<b>uselib</b>	nil



## CDF macro cap

## CDF properties

<b>formInitProc</b>	"PasCdfFormInit"
<b>doneProc</b>	""
<b>buttonFieldWidth</b>	340
<b>fieldHeight</b>	35
<b>fieldWidth</b>	350
<b>promptWidth</b>	175
<b>instDisplayMode</b>	"instName"
<b>instNameType</b>	"schematic"
<b>netNameType</b>	"schematic"
<b>paramDisplayMode</b>	"parameter"
<b>paramEvaluate</b>	"nil nil nil t nil"
<b>paramSimType</b>	"DC"
<b>termDisplayMode</b>	"netName"
<b>termSimType</b>	"DC"
<b>opPointLabelSet</b>	"i"
<b>paramLabelSet</b>	"-model c m"

**Callbacks****mimcap****CDF Callback**

```

procedure(gpdk180_capCB(param) ;_Feb 13 03 quoc 3858
let((procName cdfld libName cellName dpt grid scale sGrid epsilon
  ca cf minW maxW minL maxL todo paramId cap width length
  ael_C ael_W ael_L)

procName = "gpdk180_capCB"
cdfld = cdfgData
caseq( concat(cdfld->type)
  ((cellData baseCellData userCellData)
  libName = cdfld->id->lib->name
  cellName = cdfld->id->name
  )
  (instData
  libName = cdfld->id->master->libName
  cellName = cdfld->id->master->cellName
  )
  (t
  PasAttention()
  error("%s: unexpected CDF data type - %L" procName cdfld->type)
  )
) ; ** caseq concat **

dpt = PasGetPdkParams(libName cellName)
grid = dpt->grid
scale = dpt->scale
sGrid = grid*scale
epsilon = sGrid/10.0
ca = gpdk180_capValue('ca ?dpt dpt)
cf = gpdk180_capValue('cf ?dpt dpt)
minW = gpdk180_capValue('w ?dpt dpt ?type 'min)
maxW = gpdk180_capValue('w ?dpt dpt ?type 'max)
minL = gpdk180_capValue('l ?dpt dpt ?type 'min)
maxL = gpdk180_capValue('l ?dpt dpt ?type 'max)

;; process param argument
todo = list(nil)
caseq( param
  (c
  paramId = cdfld->c
  if( cdfld->calcParam->value == "width" then
    todo = cons('w todo)
  else
    todo = cons('l todo)
  )
  )
  (w
  paramId = cdfld->w...

```

# Pcell

## mimcap

pcell gpdk180 mimcap

PCell formal params

<b>l</b>	float	cdf
<b>w</b>	float	cdf

M3V2enc = {11D}

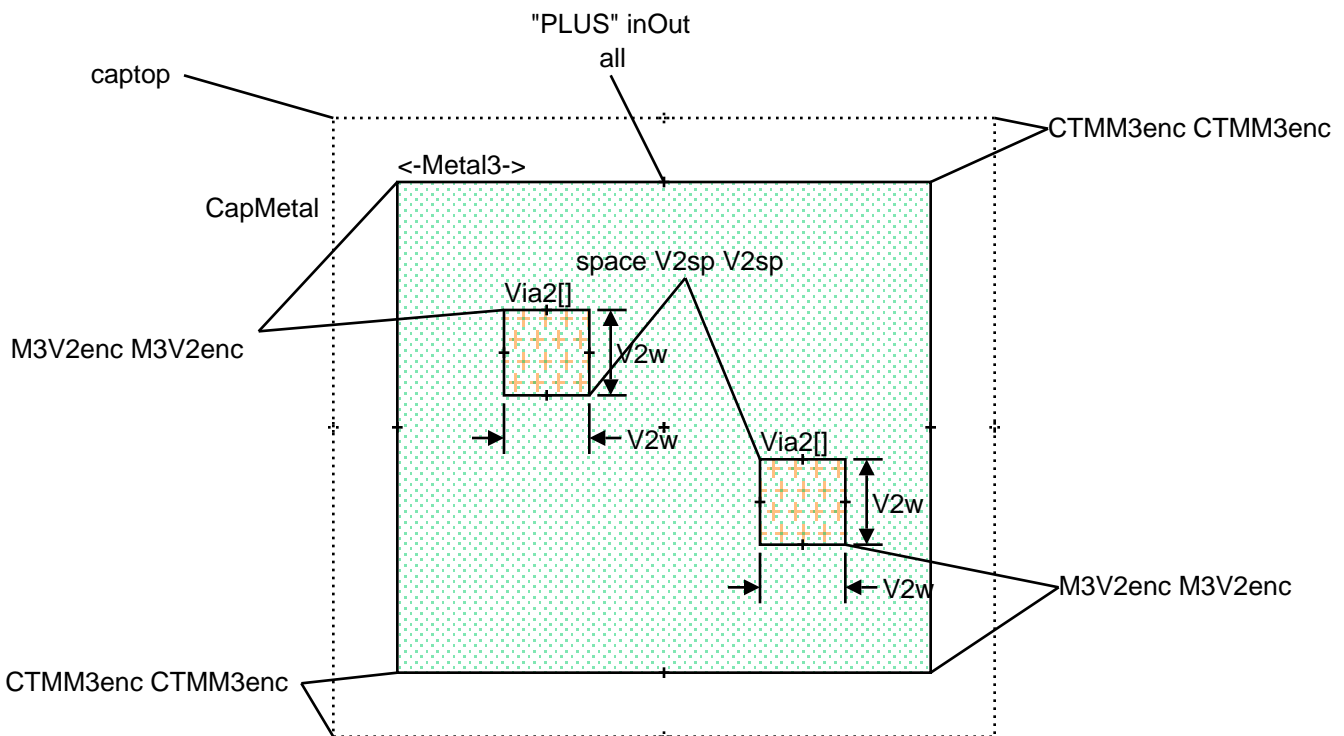
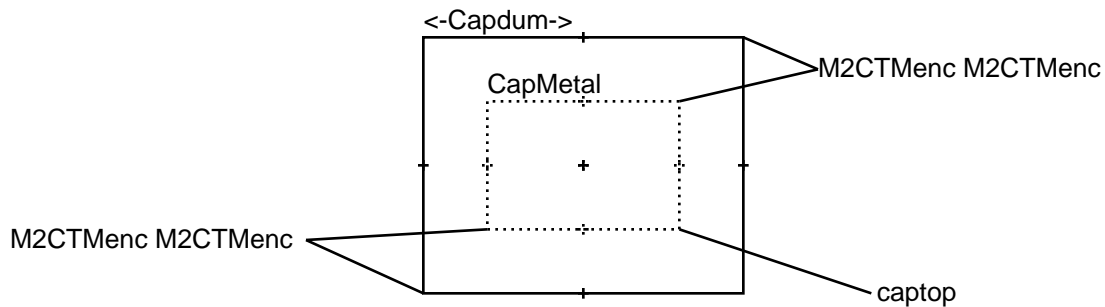
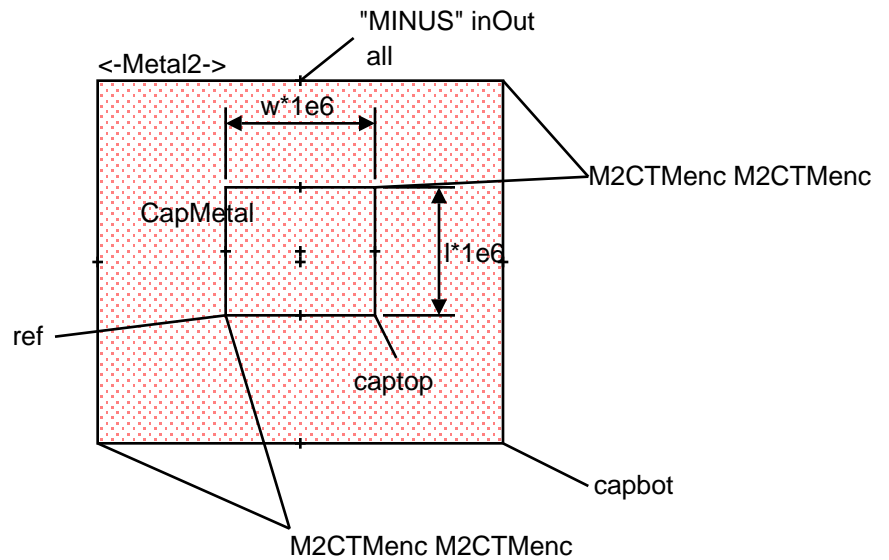
M2V2enc = {10C}

M2CTMenc = {12B}

CTMM3enc = {12D}

V2w = {10A}

V2sp = {10B}



## **OLD CDF/PCELL/CALLBACK Definitions**

## Junction Varactors

## Junction Varactor Pcells

### xjvar\_w40 pcell

pcell gpdk180 xjvar\_w40

```
pcw = cdfParseFloatString(w) * 1e6
pcl = cdfParseFloatString(l) * 1e6
pcnf = fix(cdfParseFloatString(nf))
```

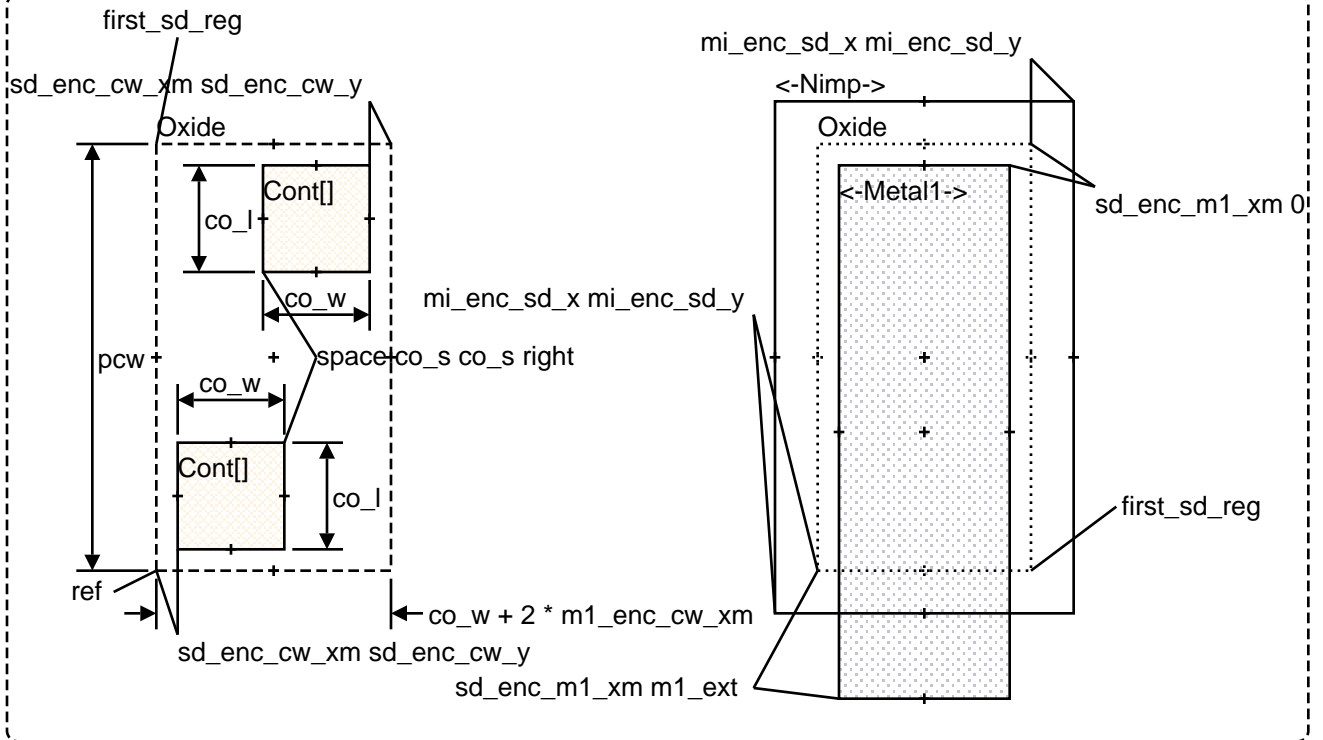
PCell formal params

<b>w</b>	string	cdf
<b>l</b>	string	cdf
<b>nf</b>	string	cdf

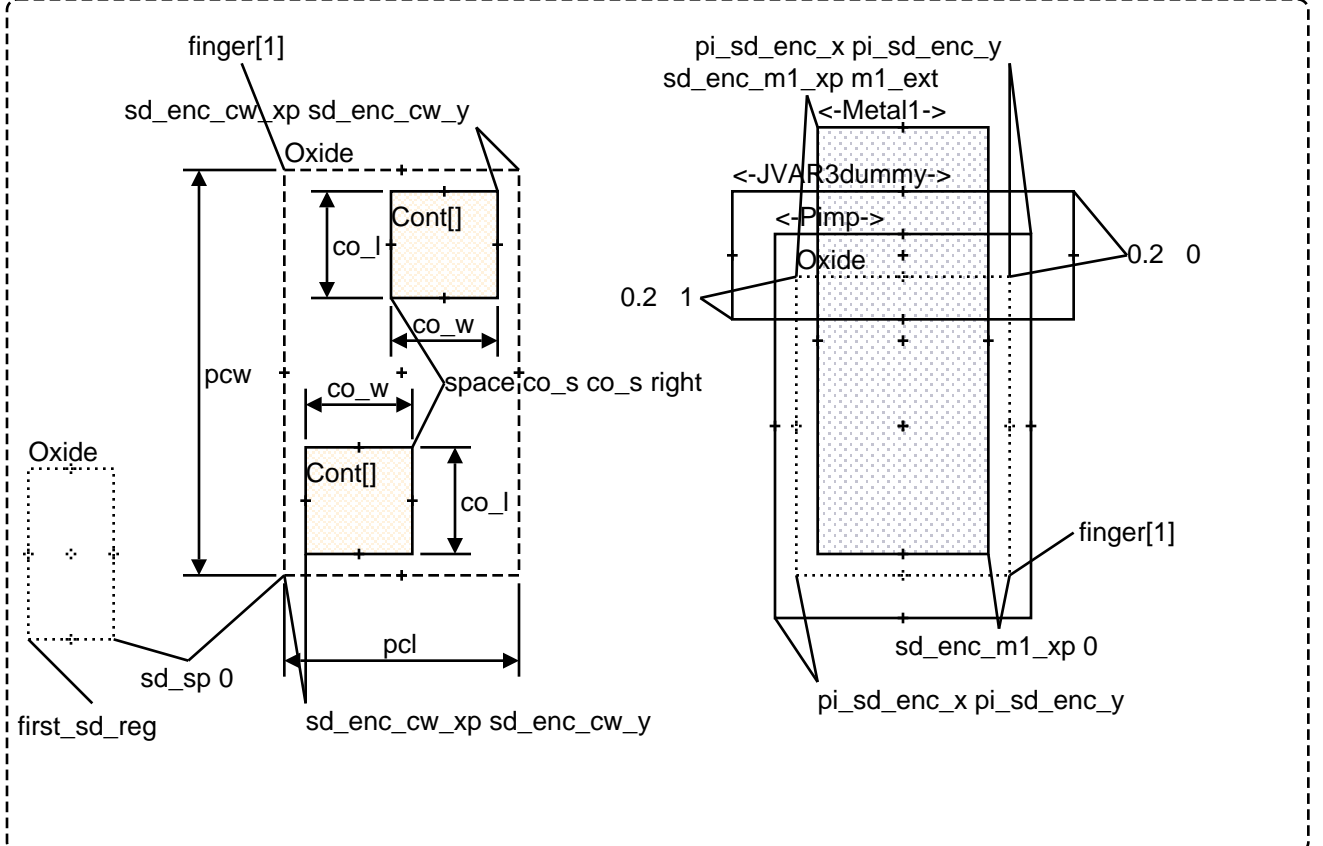
```
dpt = PasGetDeviceProps( cv )
co_w = dpt->CwW
co_l = dpt->CwW
co_s = dpt->CwSp
sd_enc_cw_xm = dpt->M1CwEncXm
sd_enc_cw_xp = gpdk180_PDKsnapToGrid((pcl - co_w) / 2 grid)
sd_enc_cw_x = dpt->SDCwEncY
sd_enc_cw_y = dpt->SDCwEncY
m1_ext = dpt->M1Ext
m1_enc_cw_xm = dpt->M1CwEncXm
m1_enc_cw_xp = dpt->M1CwEncXp
mi_enc_sd_x = dpt->MISDEncX
mi_enc_sd_y = dpt->MISDEncY
sd_enc_m1_xm = sd_enc_cw_xm - m1_enc_cw_xm
sd_enc_m1_xp = sd_enc_cw_xp - m1_enc_cw_xp
pi_sd_enc_x = dpt->PISDEncX
pi_sd_enc_y = dpt->PISDEncY
sd_sp = dpt->SDSp
nw_enc_sd_x = dpt->NWSEncX
nw_enc_sd_y = dpt->NWSEncY
ri_enc_sd_x = dpt->RISDEncX
ri_enc_sd_y = dpt->RISDEncY
ri2_enc_sd_x = dpt->RI2SEncX
ri2_enc_sd_y = dpt->RI2SEncY
m1_sw = dpt->M1SW
msPwNw = 0.5
mwPwGr = 0.6
```

pcell gpdk180 xjvar\_w40

1: include

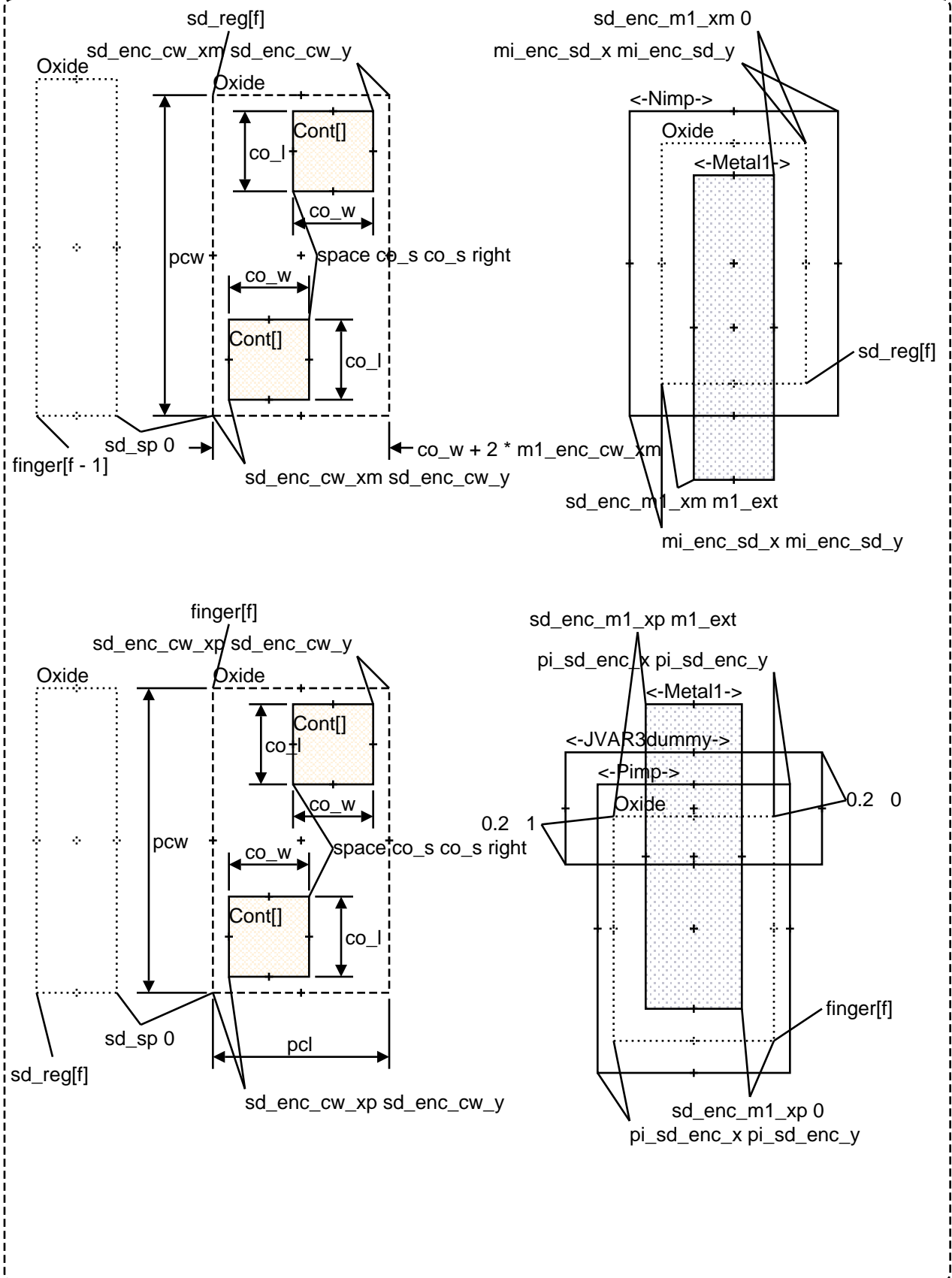


2: include



pcell gpdk180 xjvar\_w40

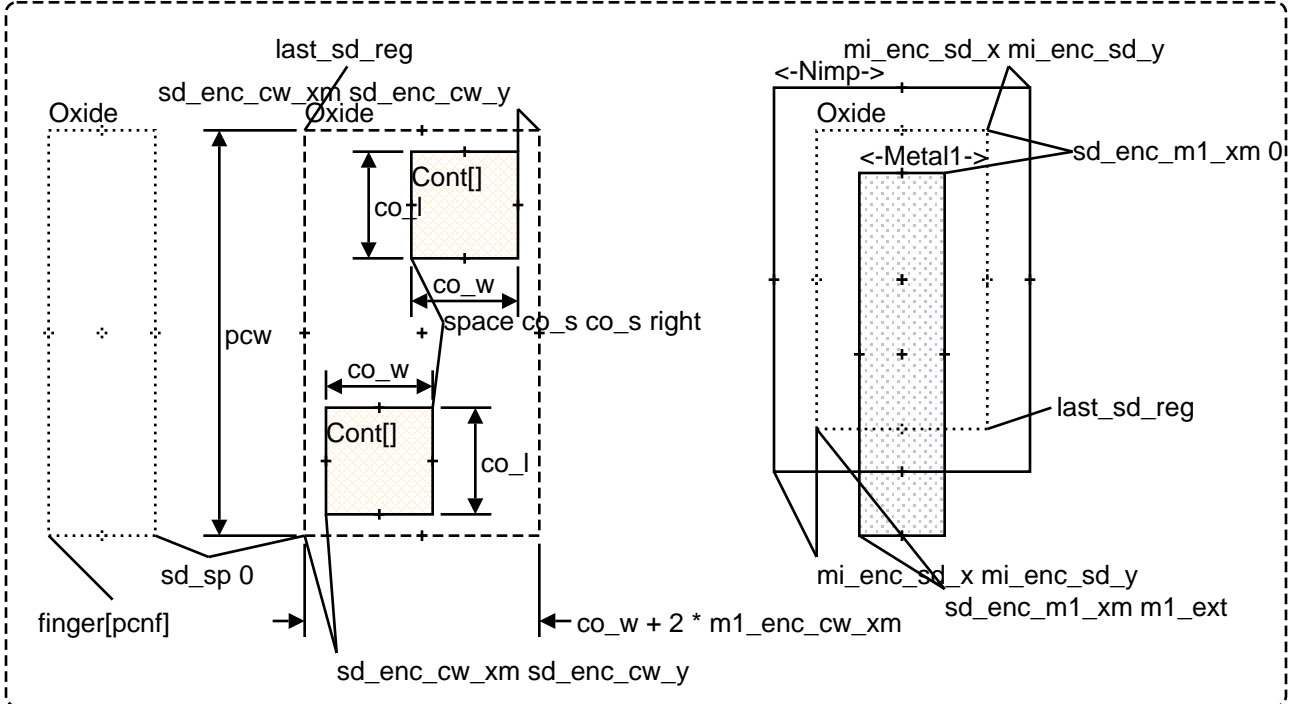
3: for f 2 pcnf



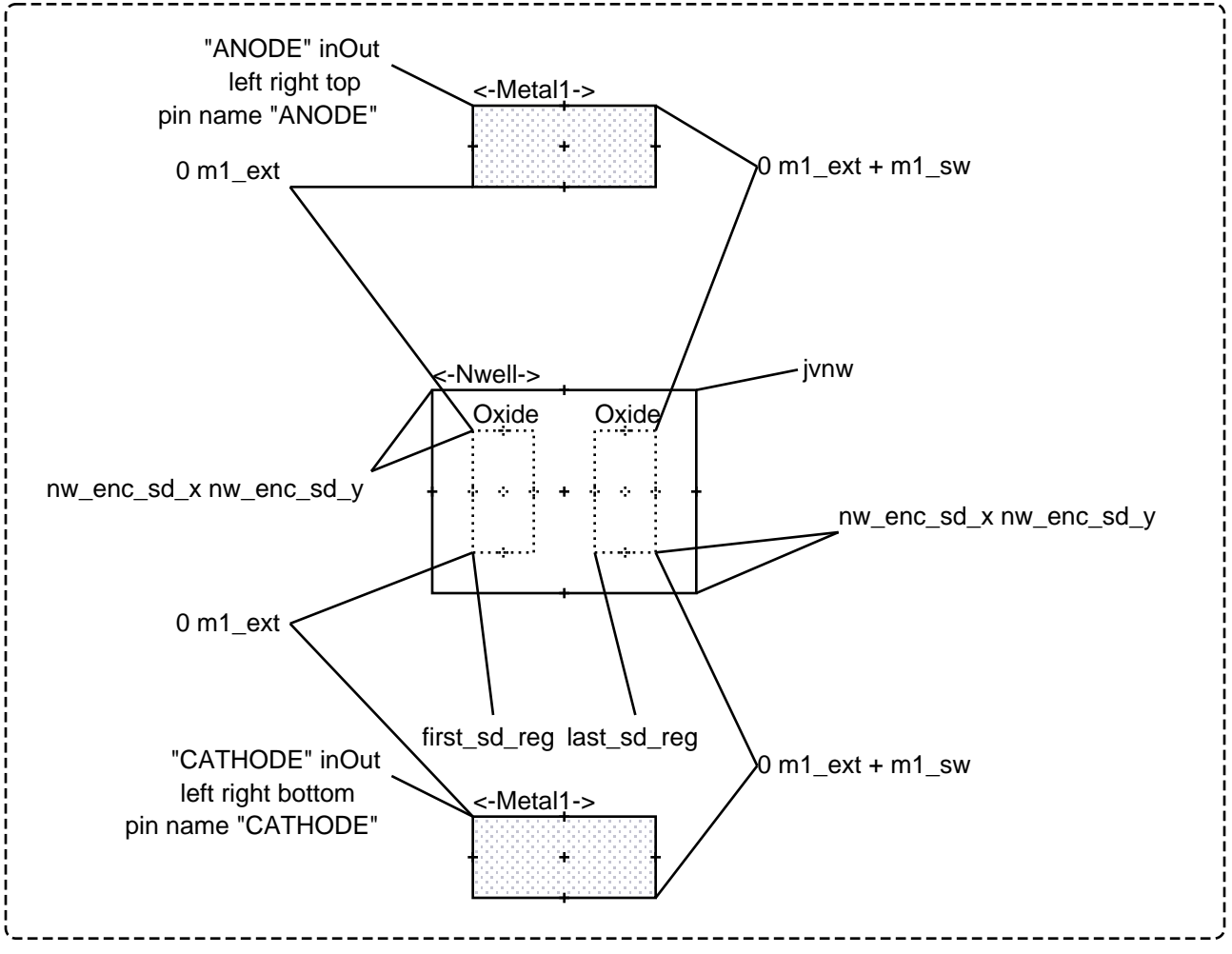


pcell gpd180 xjvar\_w40

4: include

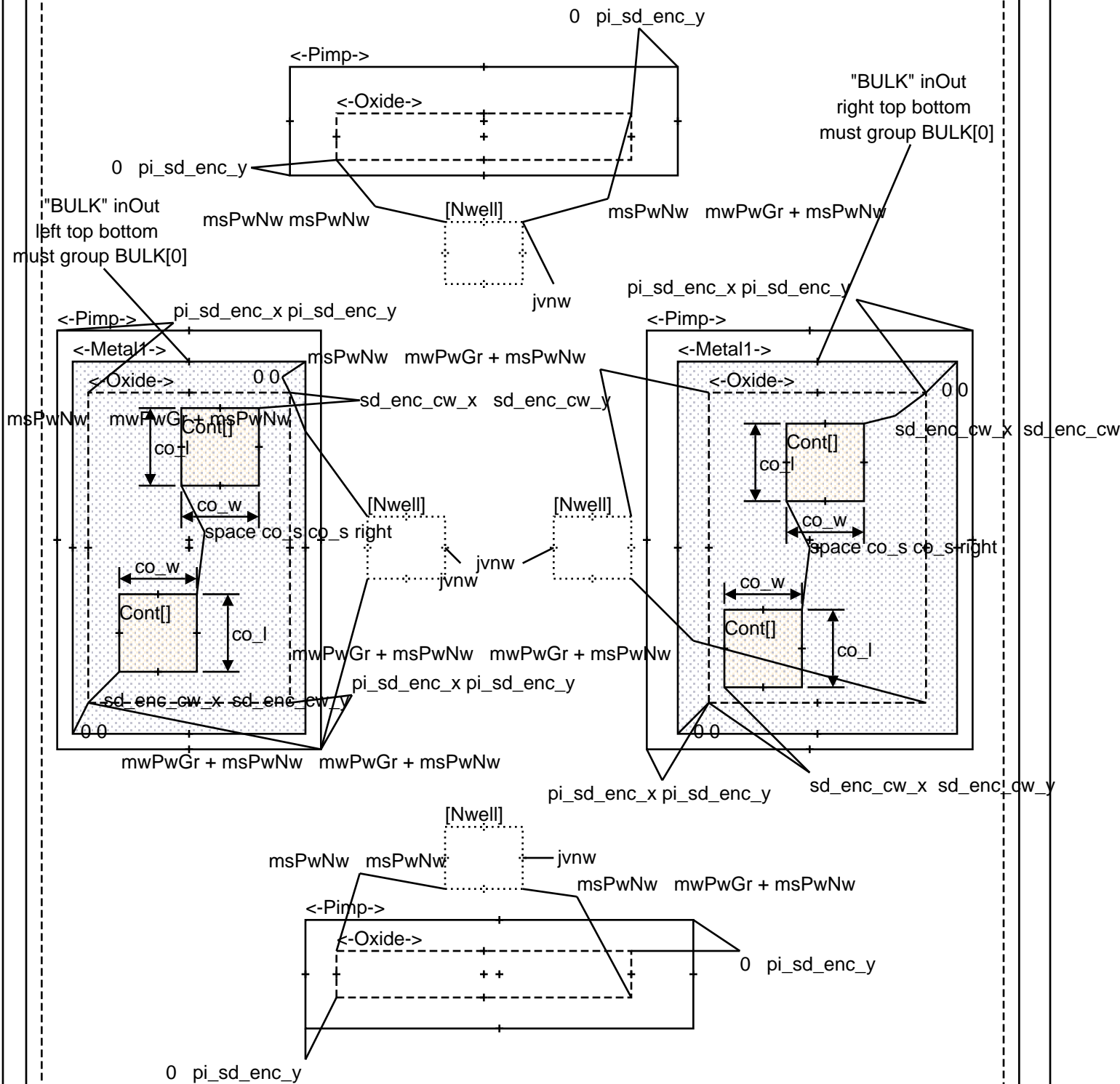


5: include



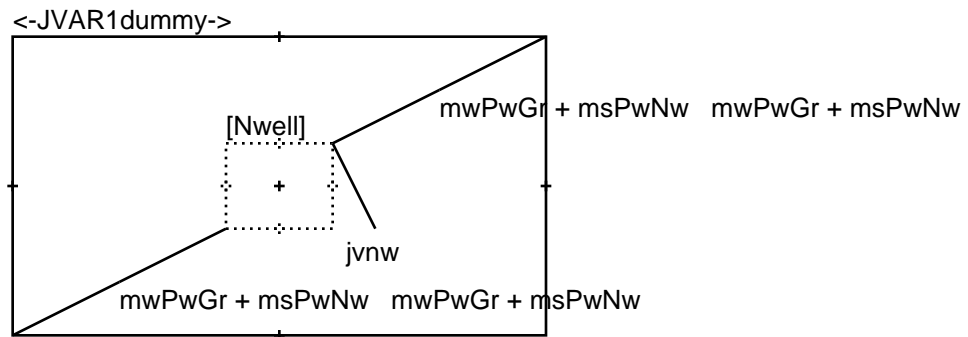
pcell gpd180 xjvar\_w40

6: include



pcell gpd180 xjvar\_w40

7: include



## xjvar\_nf36 pcell

pcell gpdk180 xjvar\_nf36

```
pcw = cdfParseFloatString(w) * 1e6
pcl = cdfParseFloatString(l) * 1e6
pcnf = fix(cdfParseFloatString(nf))
```

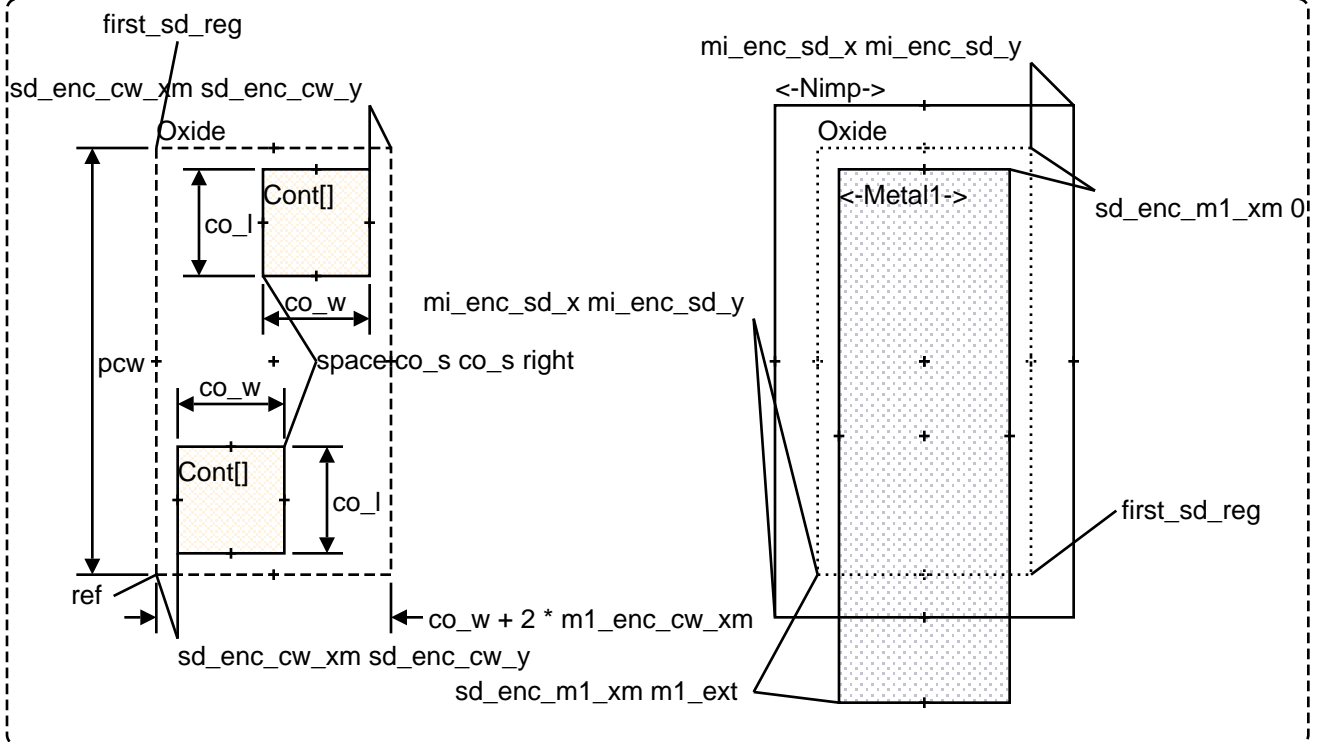
PCell formal params

<b>w</b>	string	cdf
<b>l</b>	string	cdf
<b>nf</b>	string	cdf

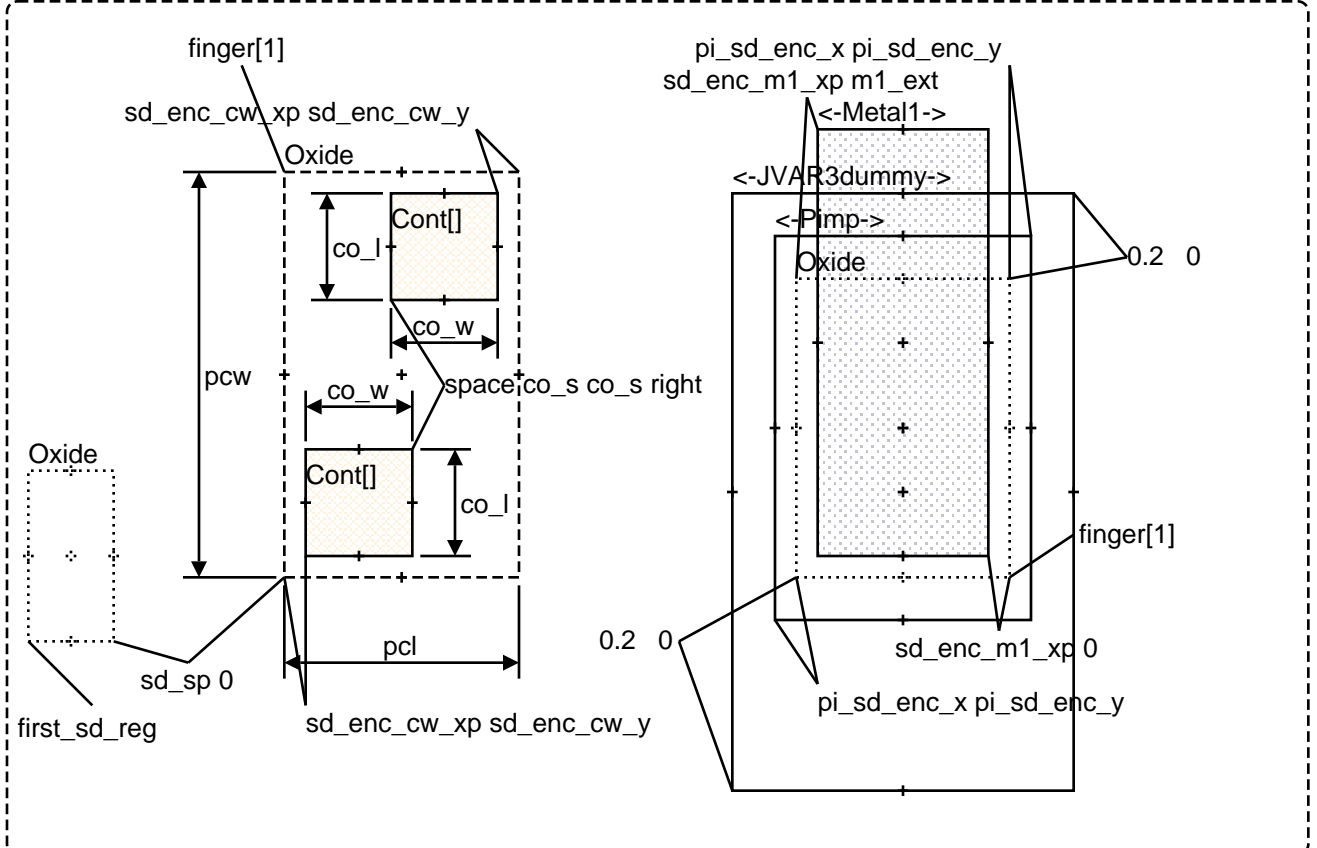
```
dpt = PasGetDeviceProps( cv )
co_w = dpt->CwW
co_l = dpt->CwW
co_s = dpt->CwSp
sd_enc_cw_xm = dpt->M1CwEncXm
sd_enc_cw_xp = gpdk180_PDKsnapToGrid((pcl - co_w) / 2 grid)
sd_enc_cw_x = dpt->SDCwEncY
sd_enc_cw_y = dpt->SDCwEncY
m1_ext = dpt->M1Ext
m1_enc_cw_xm = dpt->M1CwEncXm
m1_enc_cw_xp = dpt->M1CwEncXp
mi_enc_sd_x = dpt->MISDEncX
mi_enc_sd_y = dpt->MISDEncY
sd_enc_m1_xm = sd_enc_cw_xm - m1_enc_cw_xm
sd_enc_m1_xp = sd_enc_cw_xp - m1_enc_cw_xp
pi_sd_enc_x = dpt->PISDEncX
pi_sd_enc_y = dpt->PISDEncY
sd_sp = dpt->SDSp
nw_enc_sd_x = dpt->NWSEncX
nw_enc_sd_y = dpt->NWSEncY
ri_enc_sd_x = dpt->RISDEncX
ri_enc_sd_y = dpt->RISDEncY
ri2_enc_sd_x = dpt->RI2SEncX
ri2_enc_sd_y = dpt->RI2SEncY
m1_sw = dpt->M1SW
msPwNw = 0.5
mwPwGr = 0.6
```

pcell gpdk180 xjvar\_nf36

1: include

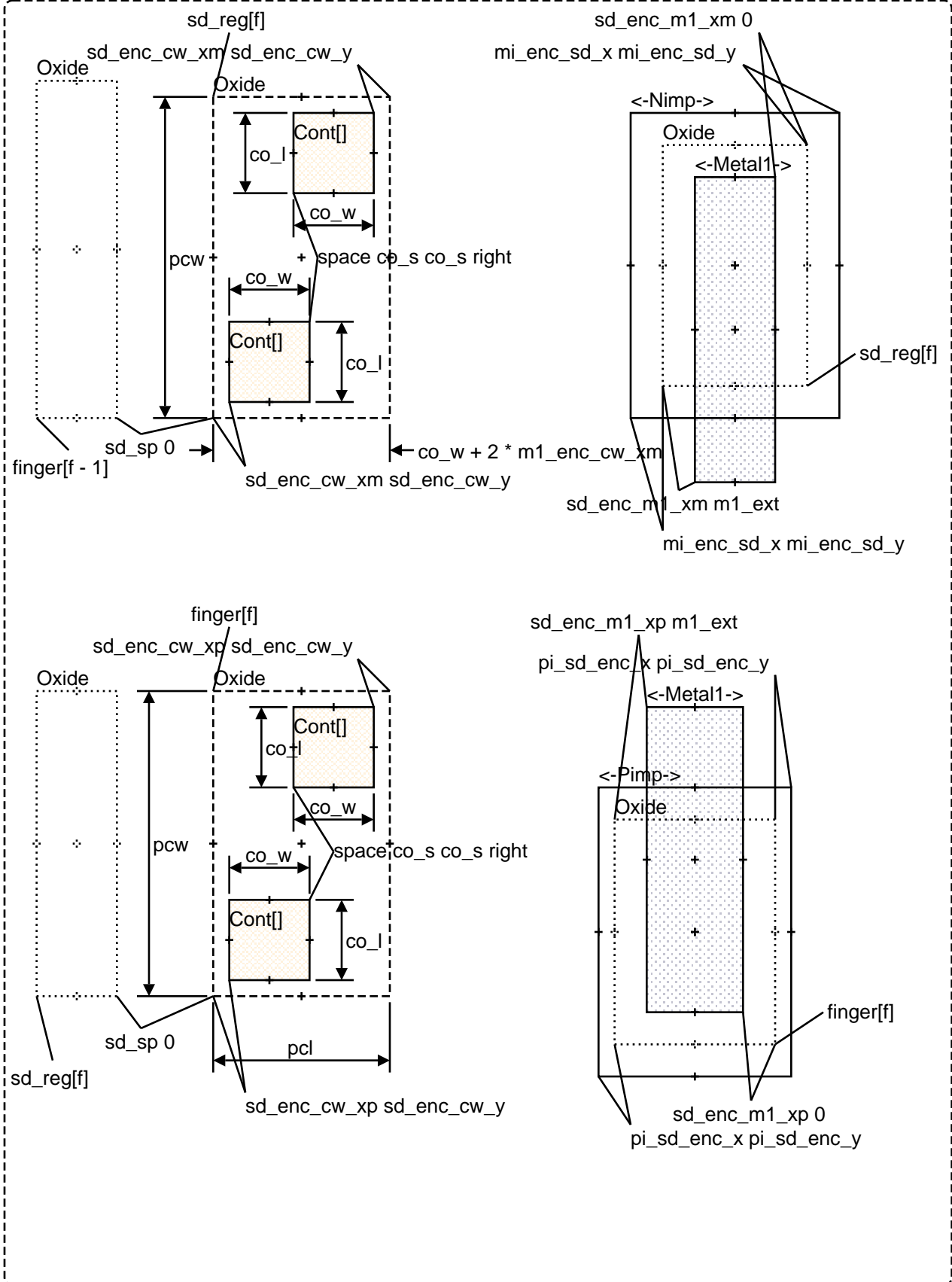


2: include



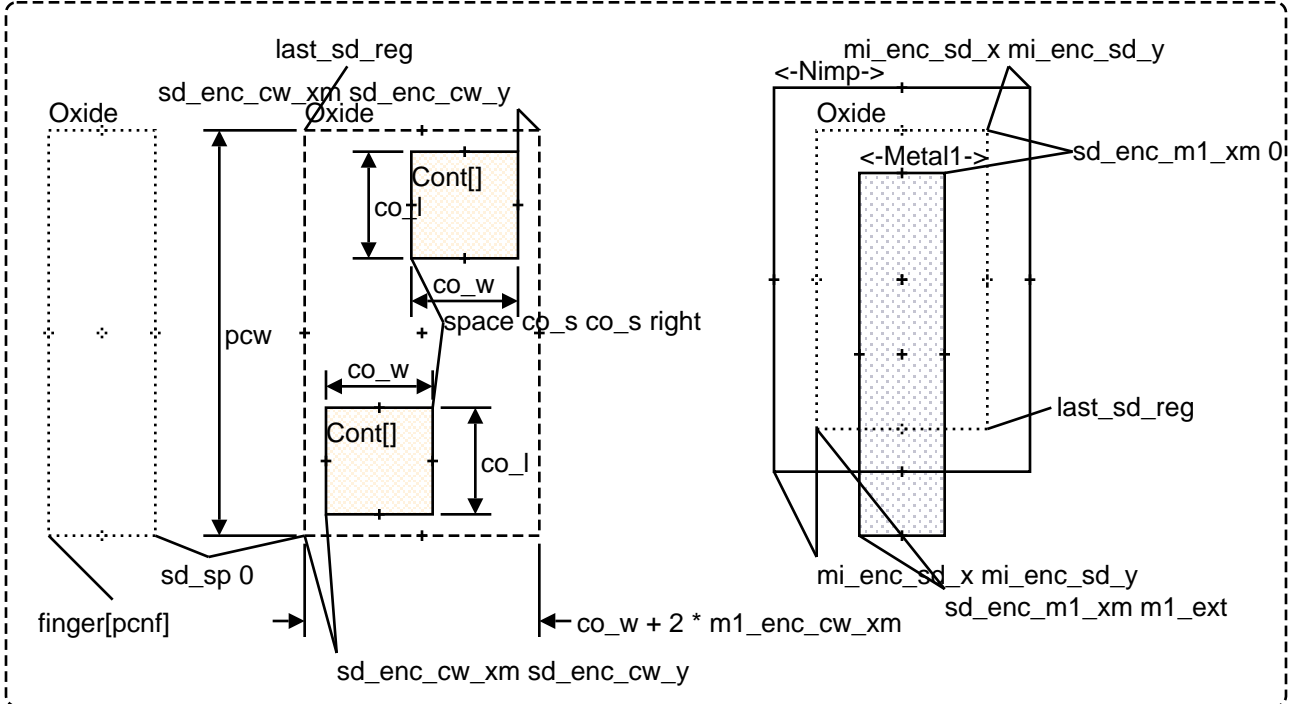
pcell gpdk180 xjvar\_nf36

3: for f 2 pcnf

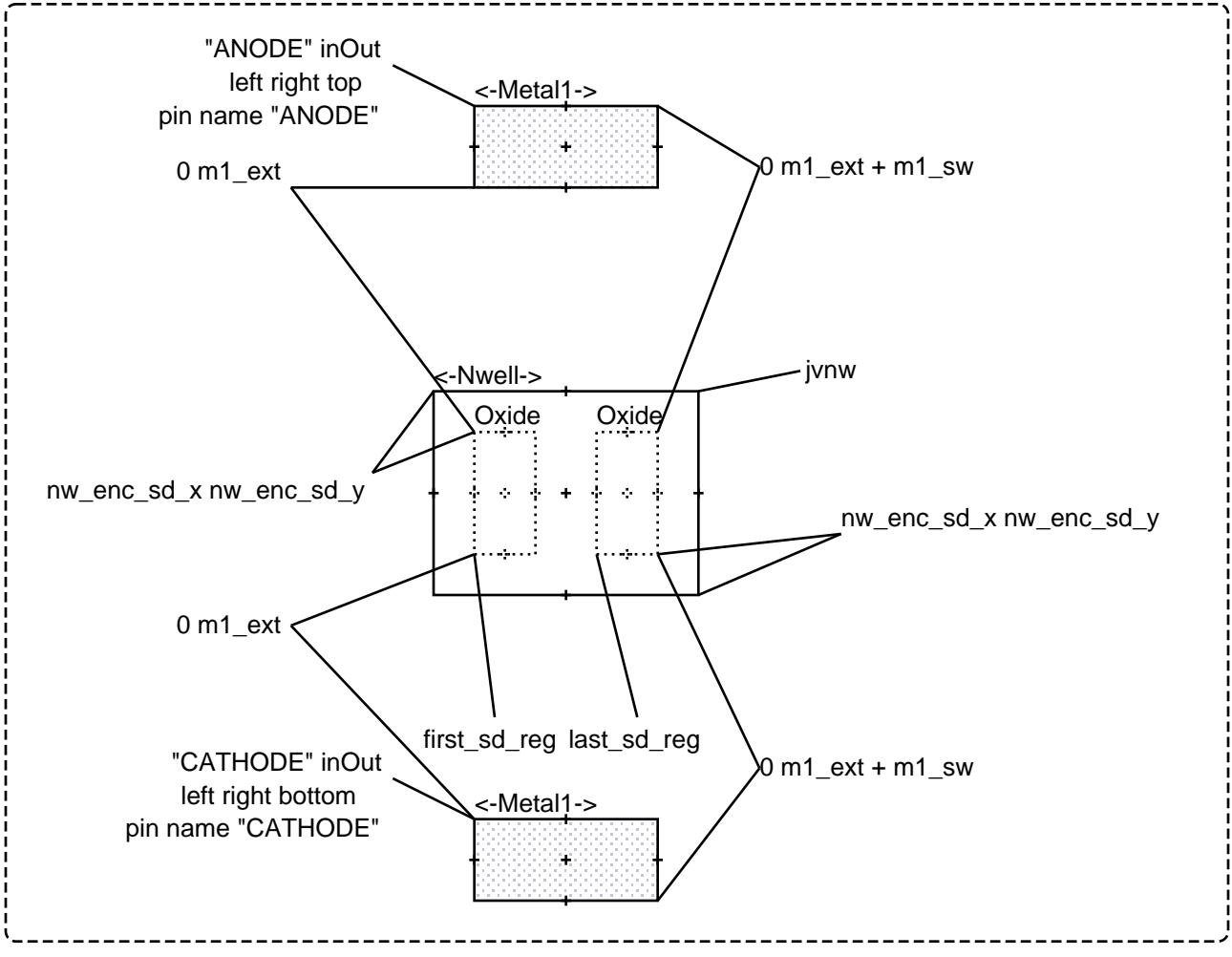


pcell gpd180 xjvar\_nf36

4: include

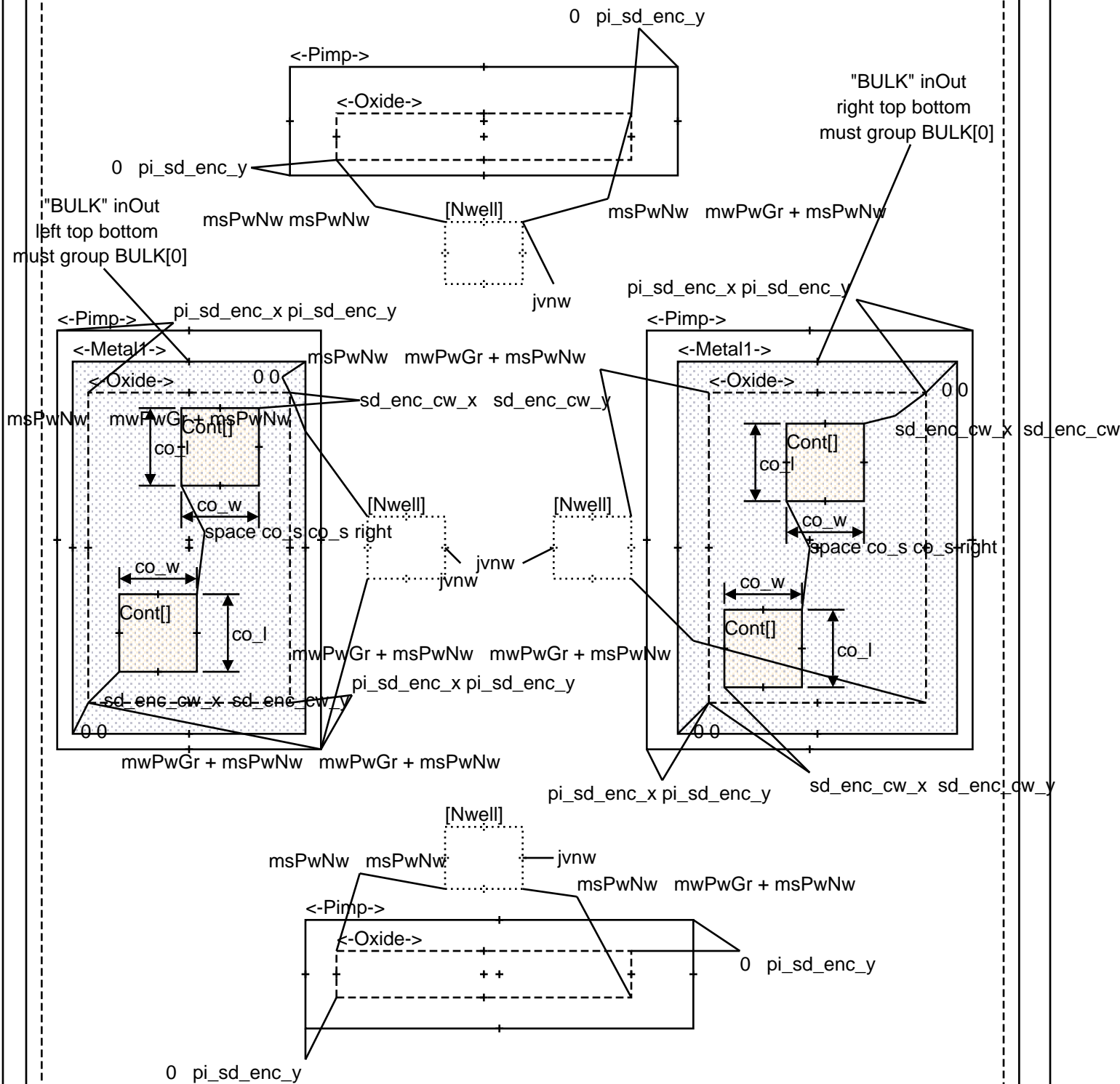


5: include



pcell gpd180 xjvar\_nf36

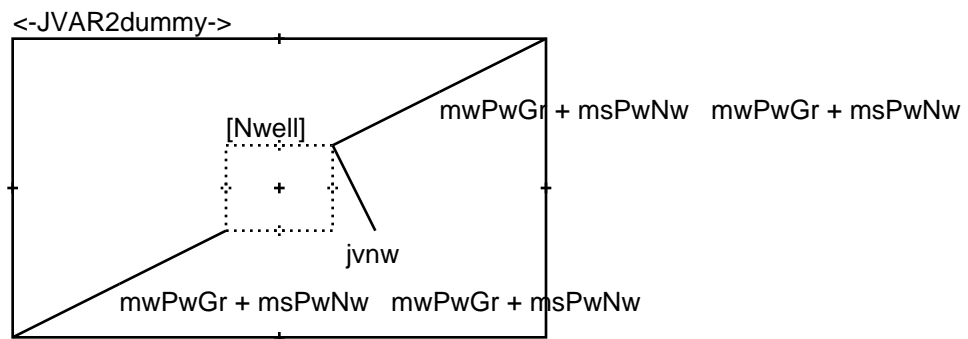
6: include





pcell gpd180 xjvar\_nf36

7: include



## Junction Varactor CDF

## xjvar\_w40 cdf

CDF gpd180 xjvar\_w40

## CDF device params

CwSp	0.2	private
CwW	0.2	private
M1CwEncXm	0.2	private
M1CwEncXp	0.2	private
M1CwEncY	0.2	private
M1Ext	0.48	private
M1SW	1.54	private
MISDEncX	0.2	private
MISDEncY	0.2	private
MaxF	50	public
MaxW	60e-6	public
MinF	1	public
MinW	20e-6	public

## CDF parameters

name	"w"
prompt	"Width (M)"
defValue	"4e-05"
type	"string"
editable	"nil"
parseAsNumber	"yes"
parseAsCEL	"yes"

## CDF parameters

name	"l"
prompt	"Length (M)"
defValue	"6e-07"
type	"string"
editable	"nil"
parseAsNumber	"yes"
parseAsCEL	"yes"

## CDF parameters

name	"m"
prompt	"Multiplier"
defValue	"1"
type	"string"
editable	"deGetEditCellView()->cellViewType != \"maskLayout\""
callback	"gpd180_jvCB()"
parseAsNumber	"yes"
parseAsCEL	"yes"

## CDF device params

NWSEncX	0.5	private
NWSEncY	0.5	private
PIMISp	0.0	private
PISDEncX	0.2	private
PISDEncY	0.2	private
RI2SEncX	0.2	private
RI2SEncY	0.2	private
RISDEncX	0.2	private
RISDEncY	0.2	private
SDCwEncX	0.2	private
SDCwEncY	0.2	private
SDSp	0.3	private
category	"varactors"	private

## CDF parameters

name	"nf"
prompt	"Number of Fingers"
defValue	"4"
type	"string"
callback	"gpd180_jvCB()"
parseAsNumber	"yes"
parseAsCEL	"yes"

## CDF parameters

name	"model"
prompt	"Model Name"
defValue	"xjvar_w40"
type	"string"
editable	"nil"
parseAsCEL	"yes"

CDF gpdk180 xjvar\_w40

CDF siminfo

<b>simulator</b>	ams
<b>propMapping</b>	(nil)
<b>instParameters</b>	(model nf m)
<b>otherParameters</b>	(model)
<b>termOrder</b>	(ANODE CATHODE BULK)
<b>isPrimitive</b>	t

CDF siminfo

<b>simulator</b>	spectre
<b>instParameters</b>	(nf m)
<b>otherParameters</b>	(model)
<b>termMapping</b>	(nil ANODE ":1" CATHODE ":2" BULK ":3")
<b>termOrder</b>	(ANODE CATHODE BULK)
<b>componentName</b>	subcircuit

CDF cellview

<b>auCdl</b>	baseline_gpdk	jvar	symbol	
<b>auLvs</b>	baseline_gpdk	jvar	symbol	
<b>spectre</b>	baseline_gpdk	jvar	symbol	
<b>symbol</b>	baseline_gpdk	jvar	symbol	
<b>ivpcell</b>	\$lib	\$cell	symbol	5

CDF siminfo

<b>simulator</b>	auCdl
<b>namePrefix</b>	"X"
<b>termOrder</b>	(ANODE CATHODE BULK)
<b>instParameters</b>	(model nf m)
<b>netlistProcedure</b>	_ansCdlSubcktCall

CDF properties

<b>formInitProc</b>	""
<b>doneProc</b>	""
<b>buttonFieldWidth</b>	340
<b>fieldHeight</b>	35
<b>fieldWidth</b>	350
<b>promptWidth</b>	175
<b>paramLabelSet</b>	"-model nf m"

CDF siminfo

<b>simulator</b>	auLvs
<b>namePrefix</b>	D
<b>termOrder</b>	(ANODE CATHODE BULK)
<b>componentName</b>	"xjvar_w40"
<b>instParameters</b>	(nf m)
<b>netlistProcedure</b>	ansLvsCompPrim

**xjvar\_nf36 cdf**

CDF gpd180 xjvar\_nf36

## CDF device params

<b>CwSp</b>	0.2	private
<b>CwW</b>	0.2	private
<b>M1CwEncXm</b>	0.2	private
<b>M1CwEncXp</b>	0.2	private
<b>M1CwEncY</b>	0.2	private
<b>M1Ext</b>	0.48	private
<b>M1SW</b>	1.54	private
<b>MISDEncX</b>	0.2	private
<b>MISDEncY</b>	0.2	private
<b>MaxF</b>	50	public
<b>MaxW</b>	60e-6	public
<b>MinF</b>	1	public
<b>MinW</b>	20e-6	public

## CDF parameters

<b>name</b>	"w"
<b>prompt</b>	"Width (M)"
<b>defValue</b>	"2e-05"
<b>type</b>	"string"
<b>callback</b>	"gpd180_jvCB()"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"l"
<b>prompt</b>	"Length (M)"
<b>defValue</b>	"6e-07"
<b>type</b>	"string"
<b>editable</b>	"nil"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"m"
<b>prompt</b>	"Multiplier"
<b>defValue</b>	"1"
<b>type</b>	"string"
<b>editable</b>	"deGetEditCellView()->cellViewType != \"maskLayout\""
<b>callback</b>	"gpd180_jvCB()"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF device params

<b>NWSEncX</b>	0.5	private
<b>NWSEncY</b>	0.5	private
<b>PIMISp</b>	0.0	private
<b>PISDEncX</b>	0.2	private
<b>PISDEncY</b>	0.2	private
<b>RI2SEncX</b>	0.2	private
<b>RI2SEncY</b>	0.2	private
<b>RISDEncX</b>	0.2	private
<b>RISDEncY</b>	0.2	private
<b>SDCwEncX</b>	0.2	private
<b>SDCwEncY</b>	0.2	private
<b>SDSp</b>	0.3	private
<b>category</b>	"varactors"	private

## CDF parameters

<b>name</b>	"nf"
<b>prompt</b>	"Number of Fingers"
<b>defValue</b>	"36"
<b>type</b>	"string"
<b>editable</b>	"nil"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"model"
<b>prompt</b>	"Model Name"
<b>defValue</b>	"xjvar_nf36"
<b>type</b>	"string"
<b>editable</b>	"nil"
<b>parseAsCEL</b>	"yes"

## CDF gpdk180 xjvar\_nf36

## CDF siminfo

<b>simulator</b>	ams
<b>propMapping</b>	(nil)
<b>instParameters</b>	(model w m)
<b>otherParameters</b>	(model)
<b>termOrder</b>	(ANODE CATHODE BULK)
<b>isPrimitive</b>	t

## CDF siminfo

<b>simulator</b>	spectre
<b>instParameters</b>	(w m)
<b>otherParameters</b>	(model)
<b>termMapping</b>	(nil ANODE ":1" CATHODE ":2" BULK ":3")
<b>termOrder</b>	(ANODE CATHODE BULK)
<b>componentName</b>	subcircuit

## CDF cellview

<b>auCdl</b>	baseline_gpdk	jvar	symbol	
<b>auLvs</b>	baseline_gpdk	jvar	symbol	
<b>spectre</b>	baseline_gpdk	jvar	symbol	
<b>symbol</b>	baseline_gpdk	jvar	symbol	
<b>ivpcell</b>	\$lib	\$cell	symbol	5

## CDF siminfo

<b>simulator</b>	auCdl
<b>namePrefix</b>	"X"
<b>termOrder</b>	(ANODE CATHODE BULK)
<b>instParameters</b>	(model w m)
<b>netlistProcedure</b>	_ansCdlSubcktCall

## CDF properties

<b>formInitProc</b>	""
<b>doneProc</b>	""
<b>buttonFieldWidth</b>	340
<b>fieldHeight</b>	35
<b>fieldWidth</b>	350
<b>promptWidth</b>	175
<b>paramLabelSet</b>	"-model w m"

## CDF siminfo

<b>simulator</b>	auLvs
<b>namePrefix</b>	D
<b>termOrder</b>	(ANODE CATHODE BULK)
<b>componentName</b>	"xjvar_nf36"
<b>instParameters</b>	(w m)
<b>netlistProcedure</b>	ansLvsCompPrim

## Junction Varactor Callback

### CDF Callback

```

/*
 * File      : jv.cb
 *
 * Purpose   : CDF Callback for junction varactor
 *
 * Author(s) : Padman S
 *
 * Created   : 02/20/02
 *
 * Notes     :
 *
 * Modified  : 07/23/03  johnag  Updates for PAS
 *
 * Copyright (C) 2004 Cadence Design Systems, Inc.
 *           All Rights Reserved.
 */

procedure(gpdk180_jvCB()
  let((bagId libName cellName m nf minF maxF nfVar wVal wVar minW maxW)

  libName = or(cdfgData->id->libName cdfgData->id->lib->name )
  cellName = or(cdfgData->id->cellName cdfgData->id->name )
  bagId = PasGetPdkParams(libName cellName)

  minF = bagId->MinF
  maxF = bagId->MaxF
  minW = bagId->MinW
  maxW = bagId->MaxW

; check "m"
  if(rexMatchp("^[ ]*$" cdfgData->m->value) then
    artError("Multiplier value must be a positive integer - set to default")
    m = cdfParseFloatString(cdfgData->m->defValue)
  else
    m = cdfParseFloatString(cdfgData->m->value)
  )
  unless((m && floatp(m))
    artError("Multiplier value must be a number - setting to default")
    m = cdfParseFloatString(cdfgData->m->defValue)
  )
  if(m < 1 then
    artError("Multiplier must be a positive integer - setting to default")
    m = cdfParseFloatString(cdfgData->m->defValue)
  )
  cdfgData->m->value = sprintf(nil "%d" fix(m))

; check "nf"...
```

## Diode Macro

### CDF macro diode

#### CDF parameters

<b>name</b>	"model"
<b>prompt</b>	"Model name"
<b>defValue</b>	"diode"
<b>type</b>	"string"
<b>editable</b>	"nil"
<b>parseAsCEL</b>	"yes"

#### CDF parameters

<b>name</b>	"macro"
<b>prompt</b>	"Model name"
<b>defValue</b>	"iPar(\"model\")"
<b>type</b>	"string"
<b>display</b>	"nil"
<b>parseAsCEL</b>	"yes"

#### CDF parameters

<b>name</b>	"w"
<b>prompt</b>	"Width (M)"
<b>units</b>	"lengthMetric"
<b>defValue</b>	sprintf( nil "%g" PasGetPdkParams("\$lib" "\$cell")~>MinW)
<b>type</b>	"string"
<b>callback</b>	"gpd180_diodeCB()"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

#### CDF parameters

<b>name</b>	"l"
<b>prompt</b>	"Length (M)"
<b>units</b>	"lengthMetric"
<b>defValue</b>	sprintf( nil "%g" PasGetPdkParams("\$lib" "\$cell")~>MinL)
<b>type</b>	"string"
<b>callback</b>	"gpd180_diodeCB()"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

#### CDF parameters

<b>name</b>	"m"
<b>prompt</b>	"Multiplier"
<b>defValue</b>	"1"
<b>type</b>	"string"
<b>editable</b>	"deGetEditCellView()~>cellViewType != \"maskLayout\""
<b>callback</b>	"gpd180_diodeCB()"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

#### CDF device params

<b>category</b>	"diode"	private	Library Manager Category
-----------------	---------	---------	--------------------------

## CDF macro diode

## CDF parameters

<b>name</b>	"area"
<b>prompt</b>	"Device area"
<b>defValue</b>	sprintf( nil "%g" PasGetPdkParams("\$lib" "\$cell")~>MinW*PasGetPdkParams("\$lib" "\$cell")~>MinL)
<b>type</b>	"string"
<b>editable</b>	"nil"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"pj"
<b>prompt</b>	"Periphery of junction"
<b>defValue</b>	sprintf( nil "%g" 2*PasGetPdkParams("\$lib" "\$cell")~>MinW+2*PasGetPdkParams("\$lib" "\$cell")~>MinL)
<b>type</b>	"string"
<b>editable</b>	"nil"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF siminfo

<b>simulator</b>	auCdl
<b>instParameters</b>	(area)
<b>netlistProcedure</b>	ansCdlCompPrim
<b>componentName</b>	diode
<b>termOrder</b>	(PLUS MINUS)
<b>namePrefix</b>	"D"
<b>modelName</b>	"\$cell"

## CDF parameters

<b>name</b>	"calcParam"
<b>prompt</b>	"Calculate Parameter"
<b>type</b>	"cyclic"
<b>choices</b>	list("area" "width" "length")
<b>defValue</b>	"area"
<b>display</b>	"nil"

## CDF siminfo

<b>simulator</b>	auLvs
<b>namePrefix</b>	"D"
<b>termOrder</b>	(PLUS MINUS)
<b>componentName</b>	"\$cell"
<b>instParameters</b>	(area m)
<b>netlistProcedure</b>	ansLvsCompPrim

## CDF siminfo

<b>simulator</b>	spectre
<b>otherParameters</b>	(model)
<b>instParameters</b>	(area pj m)
<b>termOrder</b>	(PLUS MINUS)
<b>termMapping</b>	(nil PLUS \:1 MINUS \:2)
<b>componentName</b>	pdio

## CDF siminfo

<b>simulator</b>	ams
<b>otherParameters</b>	(model)
<b>instParameters</b>	(model area pj m)
<b>termOrder</b>	(PLUS MINUS)
<b>isPrimitive</b>	t



## CDF macro diode

## CDF siminfo

<b>simulator</b>	ads
<b>netlistProcedure</b>	ADSSimCompPrim
<b>otherParameters</b>	(model)
<b>instParameters</b>	(Area Periph _M)
<b>componentName</b>	nil
<b>termOrder</b>	(PLUS MINUS)
<b>termMapping</b>	(nil PLUS ":P1" MINUS ":P2")
<b>propMapping</b>	(nil Area area Periph pj _M m)
<b>typeMapping</b>	nil
<b>uselib</b>	nil

## CDF properties

<b>formInitProc</b>	""
<b>doneProc</b>	""
<b>buttonFieldWidth</b>	340
<b>fieldHeight</b>	35
<b>fieldWidth</b>	350
<b>promptWidth</b>	175
<b>modelLabelSet</b>	"is rs n"
<b>opPointLabelSet</b>	"i v region"
<b>paramLabelSet</b>	"-model area m"

# ndio

CDF gpdk180 ndio

include macro diode

CDF parameters

<b>name</b>	"model"
<b>defValue</b>	"\$cell"

CDF device params

<b>MinW</b>	0.6u	private	
<b>MaxW</b>	20.0u	private	
<b>MinL</b>	0.6u	private	
<b>MaxL</b>	20.0u	private	

CDF device params

<b>grid</b>	\$grid	private	MFG Grid
<b>scale</b>	1.0e-6	private	Dimension scale factor

CDF cellview

<b>symbol</b>	pas_std	diode2	symbol	
<b>spectre</b>	pas_std	diode2	symbol	
<b>auLvs</b>	pas_std	diode2	symbol	
<b>auCdl</b>	pas_std	diode2	symbol	
<b>ads</b>	pas_std	diode2	symbol	
<b>ivpcell</b>	gpdk180	\$cell	symbol	40

# pdio

CDF gpdk180 pdio

include macro diode

CDF parameters

<b>name</b>	"model"
<b>defValue</b>	"\$cell"

CDF device params

<b>MinW</b>	0.6u	private	
<b>MaxW</b>	20.0u	private	
<b>MinL</b>	0.6u	private	
<b>MaxL</b>	20.0u	private	

CDF device params

<b>grid</b>	\$grid	private	MFG Grid
<b>scale</b>	1.0e-6	private	Dimension scale factor

CDF cellview

<b>symbol</b>	pas_std	diode2	symbol	
<b>spectre</b>	pas_std	diode2	symbol	
<b>auLvs</b>	pas_std	diode2	symbol	
<b>auCdl</b>	pas_std	diode2	symbol	
<b>ads</b>	pas_std	diode2	symbol	
<b>ivpcell</b>	gpdk180	\$cell	symbol	40

## inductor

CDF gpdk180 ind

CDF parameters

<b>name</b>	"ind"
<b>prompt</b>	"Inductance (H)"
<b>defValue</b>	sprintf( nil "%g" PasGetPdkParams("\$lib" "\$cell")~>DefInd)
<b>type</b>	"string"
<b>editable</b>	"nil"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

CDF parameters

<b>name</b>	"nr"
<b>prompt</b>	"Number Of Turns"
<b>defValue</b>	sprintf( nil "%g" PasGetPdkParams("\$lib" "\$cell")~>DefNR)
<b>choices</b>	'("1.5" "2.5" "3.5" "4.5" "5.5" "6.5" "7.5" "8.5" "9.5")
<b>type</b>	"cyclic"
<b>callback</b>	"gpdk180_indCB()"

CDF device params

<b>DefInd</b>	5.95156e-10	private
<b>DefNR</b>	2.5	private
<b>MinR</b>	20.0u	private
<b>MaxR</b>	100.0u	private
<b>MinS</b>	2.0u	private
<b>MaxS</b>	10.0u	private
<b>MinW</b>	5.0u	private
<b>MaxW</b>	20.0u	private

CDF parameters

<b>name</b>	"rad"
<b>prompt</b>	"Inner Radius"
<b>units</b>	"lengthMetric"
<b>defValue</b>	sprintf( nil "%g" PasGetPdkParams("\$lib" "\$cell")~>MinR)
<b>type</b>	"string"
<b>callback</b>	"gpdk180_indCB()"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

CDF parameters

<b>name</b>	"width"
<b>prompt</b>	"Inductor Width"
<b>units</b>	"lengthMetric"
<b>defValue</b>	sprintf( nil "%g" PasGetPdkParams("\$lib" "\$cell")~>MinW)
<b>type</b>	"string"
<b>callback</b>	"gpdk180_indCB()"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

CDF parameters

<b>name</b>	"space"
<b>prompt</b>	"Inductor Space"
<b>units</b>	"lengthMetric"
<b>defValue</b>	sprintf( nil "%g" PasGetPdkParams("\$lib" "\$cell")~>MinS)
<b>type</b>	"string"
<b>callback</b>	"gpdk180_indCB()"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF gpd180 ind

## CDF parameters

<b>name</b>	"model"
<b>prompt</b>	"Model"
<b>defValue</b>	"inductor"
<b>type</b>	"string"
<b>display</b>	"nil"
<b>parseAsCEL</b>	"yes"

## CDF device params

<b>category</b>	"ind"	private	Library Manager Category
-----------------	-------	---------	--------------------------

## CDF device params

<b>grid</b>	\$grid	private	MFG Grid
<b>scale</b>	1.0e-6	private	Dimension scale factor

## CDF parameters

<b>name</b>	"m"
<b>prompt</b>	"multiplier"
<b>defValue</b>	"1"
<b>type</b>	"string"
<b>display</b>	"nil"
<b>editable</b>	"nil"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"macro"
<b>prompt</b>	"Model"
<b>defValue</b>	"iPar(\"model\")"
<b>type</b>	"string"
<b>display</b>	"nil"
<b>parseAsCEL</b>	"yes"

## CDF siminfo

<b>simulator</b>	auCdl
<b>modelName</b>	"\$cell"
<b>componentName</b>	ind
<b>namePrefix</b>	"I"
<b>propMapping</b>	(nil I ind)
<b>termOrder</b>	(PLUS MINUS)
<b>instParameters</b>	(I)
<b>netlistProcedure</b>	ansCdlCompPrim

## CDF siminfo

<b>simulator</b>	auLvs
<b>netlistProcedure</b>	ansLvsCompPrim
<b>instParameters</b>	(ind)
<b>componentName</b>	"\$cell"
<b>termOrder</b>	(PLUS MINUS)
<b>namePrefix</b>	"I"

## CDF siminfo

<b>simulator</b>	spectre
<b>propMapping</b>	(nil I ind)
<b>termMapping</b>	(nil PLUS \:1 MINUS \:2)
<b>instParameters</b>	(I)
<b>termOrder</b>	(PLUS MINUS)
<b>otherParameters</b>	(model)
<b>componentName</b>	ind

## CDF siminfo

<b>simulator</b>	ams
<b>isPrimitive</b>	t
<b>termOrder</b>	(PLUS MINUS)
<b>instParameters</b>	ind

CDF gpdk180 ind

CDF siminfo

<b>simulator</b>	ads
<b>netlistProcedure</b>	ADSSimCompPrim
<b>otherParameters</b>	(model)
<b>instParameters</b>	(L)
<b>componentName</b>	nil
<b>termOrder</b>	(PLUS MINUS)
<b>termMapping</b>	(nil PLUS ":P1" MINUS ":P2")
<b>propMapping</b>	(nil L ind)
<b>typeMapping</b>	nil
<b>uselib</b>	nil

CDF properties

<b>formInitProc</b>	""
<b>doneProc</b>	""
<b>buttonFieldWidth</b>	340
<b>fieldHeight</b>	35
<b>fieldWidth</b>	350
<b>promptWidth</b>	175
<b>paramLabelSet</b>	"-model nr rad"

CDF cellview

<b>symbol</b>	pas_std	ind2	symbol	
<b>spectre</b>	pas_std	ind2	symbol	
<b>auLvs</b>	pas_std	ind2	symbol	
<b>auCdl</b>	pas_std	ind2	symbol	
<b>ads</b>	pas_std	ind2	symbol	
<b>ivpcell</b>	gpdk180	ind	symbol	40

# vpnp

## CDF gpdk180 vpnp

### CDF parameters

<b>name</b>	"model"
<b>prompt</b>	"Model name"
<b>defValue</b>	"vpnp"
<b>type</b>	"string"
<b>editable</b>	"nil"
<b>parseAsCEL</b>	"yes"

### CDF parameters

<b>name</b>	"macro"
<b>prompt</b>	"Model name"
<b>defValue</b>	"iPar(\"model\")"
<b>type</b>	"string"
<b>display</b>	"nil"
<b>parseAsCEL</b>	"yes"

### CDF parameters

<b>name</b>	"EmitterSize"
<b>prompt</b>	"Emitter Size"
<b>defValue</b>	"1.3x1.3"
<b>choices</b>	'("1.3x1.3" "5.0x5.0")
<b>type</b>	"cyclic"
<b>callback</b>	"gpdk180_vpnpCB()"

### CDF parameters

<b>name</b>	"area"
<b>prompt</b>	"Area"
<b>defValue</b>	"1.69"
<b>type</b>	"string"
<b>editable</b>	"nil"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

### CDF device params

<b>grid</b>	\$grid	private	MFG Grid
<b>scale</b>	1.0e-6	private	Dimension scale factor (not used)

### CDF parameters

<b>name</b>	"m"
<b>prompt</b>	"Multiplier"
<b>defValue</b>	"1"
<b>type</b>	"string"
<b>editable</b>	"deGetEditCellView()->cellViewType != \"maskLayout\""
<b>callback</b>	"gpdk180_vpnpCB()"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

### CDF device params

<b>category</b>	"bip"	private	Library Manager Category
-----------------	-------	---------	--------------------------

CDF gpdk180 vpn

CDF siminfo

<b>simulator</b>	auCdl
<b>netlistProcedure</b>	ansCdlCompPrim
<b>propMapping</b>	(nil M m EA area)
<b>instParameters</b>	(EA M)
<b>namePrefix</b>	"Q"
<b>componentName</b>	vpn
<b>termOrder</b>	(C B E)
<b>modelName</b>	"\$cell"

CDF siminfo

<b>simulator</b>	auLvs
<b>netlistProcedure</b>	ansLvsCompPrim
<b>instParameters</b>	(area m)
<b>componentName</b>	"\$cell"
<b>termOrder</b>	(C B E)
<b>namePrefix</b>	"Q"

CDF siminfo

<b>simulator</b>	ams
<b>isPrimitive</b>	t
<b>termOrder</b>	(C B E)
<b>instParameters</b>	(area m)
<b>otherParameters</b>	(model)

CDF siminfo

<b>simulator</b>	spectre
<b>termMapping</b>	(nil C \:c B \:b E \:e)
<b>termOrder</b>	(C B E)
<b>instParameters</b>	(area m)
<b>otherParameters</b>	(model)



CDF gpdk180 vnpn

CDF siminfo

<b>simulator</b>	ads
<b>netlistProcedure</b>	ADSSimCompPrim
<b>otherParameters</b>	(model)
<b>instParameters</b>	(Area _M)
<b>componentName</b>	nil
<b>termOrder</b>	(C B E)
<b>termMapping</b>	(nil C ":P1" B ":P2" E ":P3")
<b>propMapping</b>	(nil _M m)
<b>typeMapping</b>	nil
<b>uselib</b>	nil

CDF properties

<b>formInitProc</b>	""
<b>doneProc</b>	""
<b>buttonFieldWidth</b>	340
<b>fieldHeight</b>	35
<b>fieldWidth</b>	350
<b>promptWidth</b>	175
<b>modelLabelSet</b>	"bf is vaf"
<b>opPointLabelSet</b>	"betadc ic vce"
<b>paramLabelSet</b>	"-model m area"

CDF cellview

<b>symbol</b>	pas_std	pnnp3	symbol	
<b>spectre</b>	pas_std	pnnp3	symbol	
<b>auLvs</b>	pas_std	pnnp3	symbol	
<b>auCdl</b>	pas_std	pnnp3	symbol	
<b>ads</b>	pas_std	pnnp3	symbol	
<b>ivpcell</b>	gpdk180	vnpn	symbol	40

## nnp

## CDF gpd180 npn

## CDF parameters

<b>name</b>	"model"
<b>prompt</b>	"Model name"
<b>defValue</b>	"nnp"
<b>type</b>	"string"
<b>editable</b>	"nil"
<b>parseAsCEL</b>	"yes"

## CDF device params

<b>MinE</b>	0.6	private	
<b>MaxE</b>	10.0	private	

## CDF device params

<b>grid</b>	\$grid	private	MFG Grid
<b>scale</b>	1.0e-6	private	Dimension scale factor (not used)

## CDF parameters

<b>name</b>	"Ewidth"
<b>prompt</b>	"Emitter width"
<b>defValue</b>	sprintf( nil "%g" PasGetPdkParams("\$lib" "\$cell")~>MinE)
<b>type</b>	"string"
<b>callback</b>	"gpd180_bjtCB()"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"area"
<b>prompt</b>	"Area"
<b>defValue</b>	sprintf( nil "%g" PasGetPdkParams("\$lib" "\$cell")~>MinE*PasGetPdkParams("\$lib" "\$cell")~>MinE)
<b>type</b>	"string"
<b>editable</b>	"nil"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"m"
<b>prompt</b>	"Multiplier"
<b>defValue</b>	"1"
<b>type</b>	"string"
<b>editable</b>	"deGetEditCellView()->cellViewType != \"maskLayout\""
<b>callback</b>	"gpd180_bjtCB()"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"scaleF"
<b>prompt</b>	"Scale Factor"
<b>defValue</b>	"1"
<b>type</b>	"string"
<b>editable</b>	"nil"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF device params

<b>category</b>	"bip"	private	Library Manager Category
-----------------	-------	---------	--------------------------

## CDF gpdk180 npn

## CDF parameters

<b>name</b>	"macro"
<b>prompt</b>	"Model name"
<b>defValue</b>	"iPar(\"model\")"
<b>type</b>	"string"
<b>display</b>	"nil"
<b>parseAsCEL</b>	"yes"

## CDF siminfo

<b>simulator</b>	auCdl
<b>modelName</b>	"\$cell"
<b>termOrder</b>	(C B E)
<b>componentName</b>	npn
<b>namePrefix</b>	"Q"
<b>instParameters</b>	(EA)
<b>propMapping</b>	(nil EA area)
<b>netlistProcedure</b>	ansCdlCompPrim

## CDF siminfo

<b>simulator</b>	auLvs
<b>netlistProcedure</b>	ansLvsCompPrim
<b>instParameters</b>	(area m)
<b>componentName</b>	"\$cell"
<b>termOrder</b>	(C B E)
<b>namePrefix</b>	"Q"

## CDF siminfo

<b>simulator</b>	spectre
<b>propMapping</b>	(nil area scaleF)
<b>termMapping</b>	(nil C \:c B \:b E \:e)
<b>termOrder</b>	(C B E)
<b>instParameters</b>	(area m)
<b>otherParameters</b>	(model)

## CDF siminfo

<b>simulator</b>	ams
<b>isPrimitive</b>	t
<b>termOrder</b>	(C B E)
<b>instParameters</b>	(area m)
<b>propMapping</b>	(nil area scaleF)
<b>otherParameters</b>	(model)

CDF gpdk180 npn

CDF siminfo

<b>simulator</b>	ads
<b>netlistProcedure</b>	ADSSimCompPrim
<b>otherParameters</b>	(model)
<b>instParameters</b>	(Area _M)
<b>componentName</b>	nil
<b>termOrder</b>	(C B E)
<b>termMapping</b>	(nil C ":P1" B ":P2" E ":P3")
<b>propMapping</b>	(nil Area scaleF _M m)
<b>typeMapping</b>	nil
<b>uselib</b>	nil

CDF properties

<b>formInitProc</b>	""
<b>doneProc</b>	""
<b>buttonFieldWidth</b>	340
<b>fieldHeight</b>	35
<b>fieldWidth</b>	350
<b>promptWidth</b>	175
<b>modelLabelSet</b>	"bf is vaf"
<b>opPointLabelSet</b>	"betadc ic vce"
<b>paramLabelSet</b>	"-model m area"

CDF cellview

<b>symbol</b>	pas_std	npn3	symbol	
<b>spectre</b>	pas_std	npn3	symbol	
<b>auLvs</b>	pas_std	npn3	symbol	
<b>auCdl</b>	pas_std	npn3	symbol	
<b>ads</b>	pas_std	npn3	symbol	
<b>ivpcell</b>	gpdk180	npn	symbol	40

## pnp

CDF gpd180 pnp

CDF parameters

<b>name</b>	"model"
<b>prompt</b>	"Model name"
<b>defValue</b>	"pnp"
<b>type</b>	"string"
<b>editable</b>	"nil"
<b>parseAsCEL</b>	"yes"

CDF device params

<b>MinE</b>	0.6	private
<b>MaxE</b>	10.0	private

CDF device params

<b>grid</b>	\$grid	private	MFG Grid
<b>scale</b>	1.0e-6	private	Dimension scale factor (not used)

CDF parameters

<b>name</b>	"Ewidth"
<b>prompt</b>	"Emitter width"
<b>defValue</b>	sprintf( nil "%g" PasGetPdkParams("\$lib" "\$cell")~>MinE)
<b>type</b>	"string"
<b>callback</b>	"gpd180_bjtCB()"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

CDF parameters

<b>name</b>	"area"
<b>prompt</b>	"Area"
<b>defValue</b>	sprintf( nil "%g" PasGetPdkParams("\$lib" "\$cell")~>MinE*PasGetPdkParams("\$lib" "\$cell")~>MinE)
<b>type</b>	"string"
<b>editable</b>	"nil"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

CDF parameters

<b>name</b>	"m"	CDF parameters	<b>name</b>	"scaleF"
<b>prompt</b>	"Multiplier"		<b>prompt</b>	"Scale Factor"
<b>defValue</b>	"1"		<b>defValue</b>	"1"
<b>type</b>	"string"		<b>type</b>	"string"
<b>editable</b>	"deGetEditCellView()->cellViewType != \"maskLayout\""		<b>editable</b>	"nil"
<b>callback</b>	"gpd180_bjtCB()"		<b>parseAsNumber</b>	"yes"
<b>parseAsNumber</b>	"yes"		<b>parseAsCEL</b>	"yes"
<b>parseAsCEL</b>	"yes"			

CDF device params

<b>category</b>	"bip"	private	Library Manager Category
-----------------	-------	---------	--------------------------

## CDF gpdk180 pnp

## CDF parameters

<b>name</b>	"macro"
<b>prompt</b>	"Model name"
<b>defValue</b>	"iPar(\"model\")"
<b>type</b>	"string"
<b>display</b>	"nil"
<b>parseAsCEL</b>	"yes"

## CDF siminfo

<b>simulator</b>	auLvs
<b>netlistProcedure</b>	ansLvsCompPrim
<b>instParameters</b>	(area m)
<b>componentName</b>	"\$cell"
<b>termOrder</b>	(C B E)
<b>namePrefix</b>	"Q"

## CDF siminfo

<b>simulator</b>	auCdl
<b>modelName</b>	"\$cell"
<b>termOrder</b>	(C B E)
<b>componentName</b>	pnp
<b>namePrefix</b>	"Q"
<b>instParameters</b>	(EA)
<b>propMapping</b>	(nil EA area)
<b>netlistProcedure</b>	ansCdlCompPrim

## CDF siminfo

<b>simulator</b>	spectre
<b>propMapping</b>	(nil area scaleF)
<b>termMapping</b>	(nil C \:c B \:b E \:e)
<b>termOrder</b>	(C B E)
<b>instParameters</b>	(area m)
<b>otherParameters</b>	(model)

## CDF siminfo

<b>simulator</b>	ams
<b>isPrimitive</b>	t
<b>termOrder</b>	(C B E)
<b>instParameters</b>	(area m)
<b>propMapping</b>	(nil area scaleF)
<b>otherParameters</b>	(model)

CDF gpdk180 pnp

CDF siminfo

<b>simulator</b>	ads
<b>netlistProcedure</b>	ADSSimCompPrim
<b>otherParameters</b>	(model)
<b>instParameters</b>	(Area _M)
<b>componentName</b>	nil
<b>termOrder</b>	(C B E)
<b>termMapping</b>	(nil C ":P1" B ":P2" E ":P3")
<b>propMapping</b>	(nil Area scaleF _M m)
<b>typeMapping</b>	nil
<b>uselib</b>	nil

CDF properties

<b>formInitProc</b>	""
<b>doneProc</b>	""
<b>buttonFieldWidth</b>	340
<b>fieldHeight</b>	35
<b>fieldWidth</b>	350
<b>promptWidth</b>	175
<b>modelLabelSet</b>	"bf is vaf"
<b>opPointLabelSet</b>	"betadc ic vce"
<b>paramLabelSet</b>	"-model m area"

CDF cellview

<b>symbol</b>	pas_std	pnp3	symbol	
<b>spectre</b>	pas_std	pnp3	symbol	
<b>auLvs</b>	pas_std	pnp3	symbol	
<b>auCdl</b>	pas_std	pnp3	symbol	
<b>ads</b>	pas_std	pnp3	symbol	
<b>ivpcell</b>	gpdk180	pnp	symbol	40

## moscap macro

CDF macro moscap

CDF device params

<b>dwGate</b>	2.0	private	Default channel width
<b>mwGate</b>	{2A}	private	Minimum channel Width
<b>xwGate</b>	50.0	private	Maximum channel Width
<b>mlGate</b>	{5A}	private	Minimum channel Length
<b>mxGate</b>	\$mos_mxGate	private	Minimum gate extension
<b>msGate</b>	\$mos_msGate	private	Minimum gate spacing
<b>meDiffGate</b>	\$mos_meDiffGate	private	Minimum diffusion enclosure of gate
<b>msGateCont</b>	\$mos_msGateCont	private	Minimum gate to contact spacing
<b>mwPoly</b>	\$mos_mwPoly	private	Minimum poly width on field
<b>msDiffPoly</b>	\$mos_msDiffPoly	private	Minimum diffusion to poly spacing
<b>mwCont</b>	\$mos_mwCont	private	Minimum contact width
<b>msCont</b>	\$mos_msCont	private	Minimum contact spacing
<b>masCont</b>	\$mos_masCont	private	Minimum contact array spacing
<b>meDiffCont</b>	\$mos_meDiffCont	private	Minimum diffusion enclosure of contact
<b>meeDiffCont</b>	\$mos_meeDiffCont	private	Minimum diffusion end enclosure of contact
<b>mePolyCont</b>	\$mos_mePolyCont	private	Minimum poly enclosure of contact
<b>meePolyCont</b>	\$mos_meePolyCont	private	Minimum poly end enclosure of contact
<b>mwm1</b>	\$mos_mwm1	private	Minimum metal 1 width
<b>msM1</b>	\$mos_msM1	private	Minimum metal 1 spacing
<b>meM1Cont</b>	\$mos_meM1Cont	private	Minimum metal 1 enclosure of contact
<b>meeM1Cont</b>	\$mos_meeM1Cont	private	Minimum metal 1 end enclosure of contact



## CDF macro moscap

## CDF parameters

<b>name</b>	"model"
<b>prompt</b>	"Model name"
<b>type</b>	"string"
<b>display</b>	"nil"
<b>editable</b>	"nil"
<b>parseAsCEL</b>	"yes"

## CDF device params

<b>DefC</b>	2.88f	private
<b>MinW</b>	0.6u	private
<b>MaxW</b>	100.0u	private
<b>MinL</b>	0.18u	private
<b>MaxL</b>	20.0u	private
<b>MaxF</b>	1000	private
<b>CapA</b>	0.008	private
<b>CapP</b>	0.0	private

## CDF parameters

<b>name</b>	"c"
<b>prompt</b>	"Capacitance (F)"
<b>units</b>	"capacitance"
<b>defValue</b>	sprintf( nil "%g" PasGetPdkParams("\$lib" "\$cell")~>DefC)
<b>type</b>	"string"
<b>editable</b>	"cdfgData->entryModeC->value != \"Length&Width\""
<b>callback</b>	"gpdK180_moscapCB()"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"totalC"
<b>prompt</b>	"Total Capacitance (F)"
<b>units</b>	"capacitance"
<b>defValue</b>	sprintf( nil "%g" PasGetPdkParams("\$lib" "\$cell")~>DefC)
<b>type</b>	"string"
<b>editable</b>	"nil"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"entryModeC"
<b>prompt</b>	"Capacitance Entry Mode"
<b>defValue</b>	"Capacitance"
<b>choices</b>	'("Capacitance" "Cap&Length" "Cap&Width" "Length&Width")
<b>type</b>	"cyclic"
<b>callback</b>	"gpdK180_moscapCB()"

## CDF macro moscap

## CDF parameters

<b>name</b>	"l"
<b>prompt</b>	"Length (m)"
<b>units</b>	"lengthMetric"
<b>defValue</b>	sprintf( nil "%g" PasGetPdkParams("\$lib" "\$cell")~>MinL)
<b>type</b>	"string"
<b>editable</b>	"cdfgData->entryModeC->value == \"Cap&Length\"    cdfgData->entryModeC->value == \"Length&Width\""
<b>callback</b>	"gpd180_moscapCB()"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"entryModeW"
<b>prompt</b>	"Width Entry Mode"
<b>defValue</b>	"TotalWidth"
<b>choices</b>	'("TotalWidth" "WidthPerFinger")
<b>type</b>	"cyclic"
<b>display</b>	"cdfgData->entryModeC->value == \"Cap&Width\"    cdfgData->entryModeC->value == \"Length&Width\""
<b>callback</b>	"gpd180_moscapCB()"

## CDF parameters

<b>name</b>	"w"
<b>prompt</b>	"Width (m)"
<b>units</b>	"lengthMetric"
<b>defValue</b>	sprintf( nil "%g" PasGetPdkParams("\$lib" "\$cell")~>MinW)
<b>type</b>	"string"
<b>editable</b>	"( cdfgData->entryModeC->value == \"Cap&Width\"    cdfgData->entryModeC->value == \"Length&Width\" ) && cdfgData->entryModeW->value == \"TotalWidth\""
<b>callback</b>	"gpd180_moscapCB()"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"fw"
<b>prompt</b>	"Width Per Finger (m)"
<b>units</b>	"lengthMetric"
<b>defValue</b>	sprintf( nil "%g" PasGetPdkParams("\$lib" "\$cell")~>MinW)
<b>type</b>	"string"
<b>editable</b>	"( cdfgData->entryModeC->value == \"Cap&Width\"    cdfgData->entryModeC->value == \"Length&Width\" ) && cdfgData->entryModeW->value == \"WidthPerFinger\""
<b>callback</b>	"gpd180_moscapCB()"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF macro moscap

## CDF parameters

<b>name</b>	"fingers"
<b>prompt</b>	"Number of Fingers"
<b>defValue</b>	"1"
<b>type</b>	"string"
<b>callback</b>	"gpdK180_moscapCB()"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"m"
<b>prompt</b>	"Multiplier"
<b>defValue</b>	"1"
<b>type</b>	"string"
<b>display</b>	"deGetEditCellView()->cellViewType != \"maskLayout\""
<b>editable</b>	"deGetEditCellView()->cellViewType != \"maskLayout\""
<b>callback</b>	"gpdK180_moscapCB()"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"simM"
<b>prompt</b>	"Simulation Multiplier"
<b>defValue</b>	"iPar(\"m\") * iPar(\"fingers\")"
<b>type</b>	"string"
<b>display</b>	"nil"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"macro"
<b>prompt</b>	"macro model name"
<b>defValue</b>	"iPar(\"model\")"
<b>type</b>	"string"
<b>display</b>	"nil"
<b>parseAsCEL</b>	"yes"

## CDF macro moscap

## CDF parameters

<b>name</b>	"connectGates"
<b>prompt</b>	"Gate Connection"
<b>defValue</b>	"None"
<b>choices</b>	'("None" "Top" "Bottom" "Both" "Alternate")
<b>type</b>	"radio"
<b>display</b>	"t"

## CDF parameters

<b>name</b>	"mtlCvg"
<b>prompt</b>	"Metal Coverage"
<b>defValue</b>	""
<b>type</b>	"string"
<b>display</b>	"nil"

## CDF parameters

<b>name</b>	"rightAbut"
<b>prompt</b>	"Right Abutment"
<b>defValue</b>	0
<b>type</b>	"int"
<b>display</b>	"nil"

## CDF parameters

<b>name</b>	"showTapParams"
<b>prompt</b>	"Show Tap Parameters"
<b>defValue</b>	nil
<b>type</b>	"boolean"
<b>display</b>	"nil"

## CDF parameters

<b>name</b>	"switchSD"
<b>prompt</b>	"Switch S/D"
<b>defValue</b>	nil
<b>type</b>	"boolean"
<b>display</b>	"nil"

## CDF parameters

<b>name</b>	"leftAbut"
<b>prompt</b>	"Left Abutment"
<b>defValue</b>	0
<b>type</b>	"int"
<b>display</b>	"nil"

## CDF parameters

<b>name</b>	"sdMtlWidth"
<b>prompt</b>	"S/D Metal Width"
<b>units</b>	"lengthMetric"
<b>defValue</b>	".4u"
<b>type</b>	"string"
<b>display</b>	"nil"
<b>parseAsNumber</b>	"yes"
<b>parseAsCEL</b>	"yes"

## CDF parameters

<b>name</b>	"tapCntRows"
<b>prompt</b>	"Tap Contact Rows"
<b>defValue</b>	1
<b>type</b>	"int"
<b>display</b>	"nil"

## CDF parameters

<b>name</b>	"leftTap"
<b>prompt</b>	"Left Tap"
<b>defValue</b>	t
<b>type</b>	"boolean"
<b>display</b>	"t"

## CDF parameters

<b>name</b>	"connectSD"
<b>prompt</b>	"S/D Connection"
<b>defValue</b>	"None"
<b>choices</b>	'("None" "Source" "Drain" "Both")
<b>type</b>	"radio"
<b>display</b>	"gpd180_mosDisplay('connectSD)"

## CDF parameters

<b>name</b>	"bottomTap"
<b>prompt</b>	"Bottom Tap"
<b>defValue</b>	nil
<b>type</b>	"boolean"
<b>display</b>	"t"

## CDF parameters

<b>name</b>	"tap"
<b>prompt</b>	"Bodytie Type"
<b>defValue</b>	"Detached"
<b>choices</b>	'("None" "Detached")
<b>type</b>	"cyclic"
<b>display</b>	"nil"

## CDF parameters

<b>name</b>	"rightTap"
<b>prompt</b>	"Right Tap"
<b>defValue</b>	nil
<b>type</b>	"boolean"
<b>display</b>	"t"

## CDF parameters

<b>name</b>	"tapExtension"
<b>prompt</b>	"Tap Extension"
<b>defValue</b>	""
<b>type</b>	"string"
<b>display</b>	"nil"

## CDF parameters

<b>name</b>	"topTap"
<b>prompt</b>	"Top Tap"
<b>defValue</b>	nil
<b>type</b>	"boolean"
<b>display</b>	"t"

## CDF macro moscap

## CDF siminfo

<b>simulator</b>	auLvs
<b>permuteRule</b>	""
<b>propMapping</b>	nil
<b>deviceTerminals</b>	""
<b>namePrefix</b>	"M"
<b>termOrder</b>	(D G S B)
<b>instParameters</b>	(w l m)
<b>netlistProcedure</b>	ansLvsCompPrim

## CDF siminfo

<b>simulator</b>	auCdl
<b>netlistProcedure</b>	ansCdlCompPrim
<b>instParameters</b>	(L W M)
<b>termOrder</b>	(D G S B)
<b>propMapping</b>	(nil L l W w M m)
<b>namePrefix</b>	"M"

## CDF siminfo

<b>simulator</b>	spectre
<b>opParamExprList</b>	((("cap" "OP(inst() \"cgg\")")))
<b>termMapping</b>	(nil D \:d G \:g S \:s B \:b)
<b>instParameters</b>	(w l m)
<b>termOrder</b>	(D G S B)
<b>propMapping</b>	(nil m simM w fw)
<b>otherParameters</b>	(model)

## CDF siminfo

<b>simulator</b>	ams
<b>otherParameters</b>	(model)
<b>instParameters</b>	(model w l m)
<b>termOrder</b>	(D G S B)
<b>propMapping</b>	(nil m simM w fw)
<b>isPrimitive</b>	t

## CDF siminfo

<b>simulator</b>	ads
<b>netlistProcedure</b>	ADSSimCompPrim
<b>otherParameters</b>	(model)
<b>instParameters</b>	(W L _M)
<b>componentName</b>	nil
<b>termOrder</b>	(D G S B)
<b>termMapping</b>	(nil D ":P1" G ":P2" S ":P3" B ":P4")
<b>propMapping</b>	(nil W fw L l _M simM)
<b>typeMapping</b>	nil
<b>uselib</b>	nil

## CDF properties

<b>formInitProc</b>	""
<b>doneProc</b>	""
<b>buttonFieldWidth</b>	340
<b>fieldHeight</b>	35
<b>fieldWidth</b>	350
<b>promptWidth</b>	175
<b>paramLabelSet</b>	"-totalC l w fingers"
<b>opPointLabelSet</b>	"ids vgs vds vth vdsat"
<b>modelLabelSet</b>	"vtho kf beta0"

## nmoscap

CDF gpd180 nmoscap

include macro moscap

CDF parameters

<b>name</b>	"model"
<b>defValue</b>	"nmos1"

CDF device params

<b>grid</b>	\$grid	private	MFG Grid
<b>scale</b>	1e-6	private	Dimension scale factor
<b>model</b>	"nmos1"	private	Device Model Name
<b>category</b>	"cap"	private	Library Manager Category

CDF device params

<b>meImplDiff</b>	\$nmos_meImplDiff	private	Minimum implant enclosure of diffusion
<b>meImplGate</b>	\$nmos_meImplGate	private	Minimum implant enclosure of gate
<b>meImplGate</b>	\$nmos_meImplGate	private	Minimum implant end enclosure of gate
<b>meImplPoly</b>	\$nmos_meImplPoly	private	Minimum implant enclosure of poly on field
<b>meNbIDiff</b>	.93	private	Minimum Nbl enclosure of diff
<b>meCapDiff</b>	.93	private	Minimum Cap enclosure of diff
<b>meNbITap</b>	.93	private	Minimum Nbl enclosure of Tap
<b>meCapTap</b>	.93	private	Minimum Cap enclosure of Tap

CDF device params

<b>msTapDiff</b>	\$tap_nmos_msTapDiff	private	Minimum tap diffusion to device diffusion spacing
<b>msTapImpl</b>	\$tap_nmos_msTapImpl	private	Minimum tap diffusion to device implant spacing
<b>meTimpTap</b>	\$tap_nmos_meTimpTap	private	Minimum tap implant enclosure of tap diffusion
<b>msTimpDiff</b>	\$tap_nmos_msTimpDiff	private	Minimum tap implant to device diffusion spacing
<b>msTimpImpl</b>	\$tap_nmos_msTimpImpl	private	Minimum tap implant to device implant spacing
<b>maTap</b>	\$tap_nmos_maTap	private	Minimum tap diffusion area
<b>maTimp</b>	\$tap_nmos_maTimp	private	Minimum tap implant area

CDF cellview

<b>symbol</b>	baseline_gpd180	nmoscap4n	symbol	
<b>spectre</b>	baseline_gpd180	nmoscap4n	symbol	
<b>auLvs</b>	baseline_gpd180	nmoscap4n	symbol	
<b>auCdl</b>	baseline_gpd180	nmoscap4n	symbol	
<b>ads</b>	baseline_gpd180	nmoscap4n	symbol	
<b>ivpcell</b>	gpd180	nmoscap	symbol	40

CDF siminfo

<b>simulator</b>	auCdl
<b>componentName</b>	nmoscap
<b>modelName</b>	"\$cell"

CDF siminfo

<b>simulator</b>	auLvs
<b>componentName</b>	nmoscap

## nmoscap3

CDF gpd180 nmoscap3

include macro moscap

CDF parameters

<b>name</b>	"model"
<b>defValue</b>	"nmos1"

CDF device params

<b>grid</b>	\$grid	private	MFG Grid
<b>scale</b>	1e-6	private	Dimension scale factor
<b>model</b>	"nmos1"	private	Device Model Name
<b>category</b>	"cap"	private	Library Manager Category

CDF device params

<b>meImplDiff</b>	\$nmos_meImplDiff	private	Minimum implant enclosure of diffusion
<b>meImplGate</b>	\$nmos_meImplGate	private	Minimum implant enclosure of gate
<b>meImplGate</b>	\$nmos_meImplGate	private	Minimum implant end enclosure of gate
<b>meImplPoly</b>	\$nmos_meImplPoly	private	Minimum implant enclosure of poly on field
<b>meNbIDiff</b>	.93	private	Minimum Nbl enclosure of diff
<b>meCapDiff</b>	.93	private	Minimum Cap enclosure of diff
<b>meNbITap</b>	.93	private	Minimum Nbl enclosure of Tap
<b>meCapTap</b>	.93	private	Minimum Cap enclosure of Tap

CDF device params

<b>msTapDiff</b>	\$tap_nmos_msTapDiff	private	Minimum tap diffusion to device diffusion spacing
<b>msTapImpl</b>	\$tap_nmos_msTapImpl	private	Minimum tap diffusion to device implant spacing
<b>meTimpTap</b>	\$tap_nmos_meTimpTap	private	Minimum tap implant enclosure of tap diffusion
<b>msTimpDiff</b>	\$tap_nmos_msTimpDiff	private	Minimum tap implant to device diffusion spacing
<b>msTimpImpl</b>	\$tap_nmos_msTimpImpl	private	Minimum tap implant to device implant spacing
<b>maTap</b>	\$tap_nmos_maTap	private	Minimum tap diffusion area
<b>maTimp</b>	\$tap_nmos_maTimp	private	Minimum tap implant area

CDF cellview

<b>symbol</b>	baseline_gpd180	nmoscap4n3	symbol	
<b>spectre</b>	baseline_gpd180	nmoscap4n3	symbol	
<b>auLvs</b>	baseline_gpd180	nmoscap4n3	symbol	
<b>auCdl</b>	baseline_gpd180	nmoscap4n3	symbol	
<b>ads</b>	baseline_gpd180	nmoscap4n3	symbol	
<b>ivpcell</b>	gpd180	nmoscap3	symbol	40

CDF siminfo

<b>simulator</b>	auCdl
<b>componentName</b>	nmoscap
<b>modelName</b>	nmoscap

CDF siminfo

<b>simulator</b>	auLvs
<b>componentName</b>	nmoscap

## pmoscap

CDF gpdk180 pmoscap

include macro moscap

CDF parameters

<b>name</b>	"model"
<b>defValue</b>	"pmos1"

CDF device params

<b>grid</b>	\$grid	private	MFG Grid
<b>scale</b>	1e-6	private	Dimension scale factor
<b>model</b>	"pmos1"	private	Device Model Name
<b>category</b>	"cap"	private	Library Manager Category

CDF device params

<b>meImplDiff</b>	\$pmos_meImplDiff	private	Minimum implant enclosure of diffusion
<b>meImplGate</b>	\$pmos_meImplGate	private	Minimum implant enclosure of gate
<b>meImplGate</b>	\$pmos_meImplGate	private	Minimum implant end enclosure of gate
<b>meImplPoly</b>	\$pmos_meImplPoly	private	Minimum implant enclosure of poly on field
<b>meWellDiff</b>	\$pmos_meWellDiff	private	Minimum well enclosure of diffusion
<b>meCapDiff</b>	\$pmos_meWellDiff	private	Minimum cap enclosure of diffusion

CDF device params

<b>msTapDiff</b>	\$tap_pmos_msTapDiff	private	Minimum tap diffusion to device diffusion spacing
<b>msTapImpl</b>	\$tap_pmos_msTapImpl	private	Minimum tap diffusion to device implant spacing
<b>meTimpTap</b>	\$tap_pmos_meTimpTap	private	Minimum tap implant enclosure of tap diffusion
<b>msTimpDiff</b>	\$tap_pmos_msTimpDiff	private	Minimum tap implant to device diffusion spacing
<b>msTimpImpl</b>	\$tap_pmos_msTimpImpl	private	Minimum tap implant to device implant spacing
<b>maTap</b>	\$tap_pmos_maTap	private	Minimum tap diffusion area
<b>maTimp</b>	\$tap_pmos_maTimp	private	Minimum tap implant area
<b>meWellTap</b>	\$tap_pmos_meWellTap	private	Minimum well enclosure of tap diffusion
<b>meCapTap</b>	\$tap_pmos_meWellTap	private	Minimum cap enclosure of tap diffusion

CDF cellview

<b>symbol</b>	baseline_gpdk	pmoscap4n	symbol	
<b>spectre</b>	baseline_gpdk	pmoscap4n	symbol	
<b>auLvs</b>	baseline_gpdk	pmoscap4n	symbol	
<b>auCdl</b>	baseline_gpdk	pmoscap4n	symbol	
<b>ads</b>	baseline_gpdk	pmoscap4n	symbol	
<b>ivpcell</b>	gpdk180	pmoscap	symbol	40

CDF siminfo

<b>simulator</b>	auCdl
<b>componentName</b>	pmoscap
<b>modelName</b>	"\$cell"

CDF siminfo

<b>simulator</b>	auLvs
<b>componentName</b>	pmoscap



## pmoscap3

CDF gpdk180 pmoscap3

include macro moscap

CDF parameters

<b>name</b>	"model"
<b>defValue</b>	"pmos1"

CDF device params

<b>grid</b>	\$grid	private	MFG Grid
<b>scale</b>	1e-6	private	Dimension scale factor
<b>model</b>	"pmos1"	private	Device Model Name
<b>category</b>	"cap"	private	Library Manager Category

CDF device params

<b>meImplDiff</b>	\$pmos_meImplDiff	private	Minimum implant enclosure of diffusion
<b>meImplGate</b>	\$pmos_meImplGate	private	Minimum implant enclosure of gate
<b>meImplGate</b>	\$pmos_meImplGate	private	Minimum implant end enclosure of gate
<b>meImplPoly</b>	\$pmos_meImplPoly	private	Minimum implant enclosure of poly on field
<b>meWellDiff</b>	\$pmos_meWellDiff	private	Minimum well enclosure of diffusion
<b>meCapDiff</b>	\$pmos_meWellDiff	private	Minimum cap enclosure of diffusion

CDF device params

<b>msTapDiff</b>	\$tap_pmos_msTapDiff	private	Minimum tap diffusion to device diffusion spacing
<b>msTapImpl</b>	\$tap_pmos_msTapImpl	private	Minimum tap diffusion to device implant spacing
<b>meTimpTap</b>	\$tap_pmos_meTimpTap	private	Minimum tap implant enclosure of tap diffusion
<b>msTimpDiff</b>	\$tap_pmos_msTimpDiff	private	Minimum tap implant to device diffusion spacing
<b>msTimpImpl</b>	\$tap_pmos_msTimpImpl	private	Minimum tap implant to device implant spacing
<b>maTap</b>	\$tap_pmos_maTap	private	Minimum tap diffusion area
<b>maTimp</b>	\$tap_pmos_maTimp	private	Minimum tap implant area
<b>meWellTap</b>	\$tap_pmos_meWellTap	private	Minimum well enclosure of tap diffusion
<b>meCapTap</b>	\$tap_pmos_meWellTap	private	Minimum cap enclosure of tap diffusion

CDF cellview

<b>symbol</b>	baseline_gpdk	pmoscap4n3	symbol	
<b>spectre</b>	baseline_gpdk	pmoscap4n3	symbol	
<b>auLvs</b>	baseline_gpdk	pmoscap4n3	symbol	
<b>auCdl</b>	baseline_gpdk	pmoscap4n3	symbol	
<b>ads</b>	baseline_gpdk	pmoscap4n3	symbol	
<b>ivpcell</b>	gpdk180	pmoscap3	symbol	40

CDF siminfo

<b>simulator</b>	auCdl
<b>componentName</b>	pmoscap
<b>modelName</b>	pmoscap

CDF siminfo

<b>simulator</b>	auLvs
<b>componentName</b>	pmoscap

## Component Callback Definitions

### Common Callback Procedures

#### CDF Callback

```

.....
...
...
... gpdK180_PDKblankValueCheck()
...
...
... Purpose:
... Determine whether a test value is blank
...
... Parameters:
... paramName: name of parameter being checked which is used in the error
... statement if the value is blank
... testValue: value which is being compared for a blank entry
... defValue: default value of parameter being checked which is used for the
... new value of the parameter being checked if the test value is
... determined to be blank
...
... Outputs:
... returnValue: testValue that is adjusted depending on whether it is blank
...
.....

procedure(gpdK180_PDKblankValueCheck(paramName testValue defValue)
prog((stringCondition numberCondition returnValue)

.....
...
...
... determine whether the test value is a string or non string and whether it
... it has a blank value
...
...
.....

stringCondition = stringp(testValue) &&
                    (testValue == "" || testValue == " ")
numberCondition = !stringp(testValue) && !testValue

.....
...
...
... print an error message if a blank value is encountered
...
...
.....

if(stringCondition || numberCondition then
    printf("*Error* %s must have a value - setting to default\n"
        paramName)
    returnValue = defValue
else
    returnValue = testValue...

```

## diode

### CDF Callback

```

procedure( gpd180_diodeCB()
let((MinL MinW MaxL MaxW
  cell grid scale sGrid libName deviceName device
  lVal wVal mult areaVal lVar wVar multVar perimVal
)

  setq(cell or(cdfgData->id->master cdfgData->id))
  libName = or(cdfgData->id->libName cdfgData->id->lib->name )
  deviceName = or(cdfgData->id->cellName cdfgData->id->name )
  setq(device PasGetPdkParams(libName deviceName))
  grid = device->grid
  scale = device->scale || 1e-6
  sGrid = grid * scale

  MinL = device->MinL
  MinW = device->MinW
  MaxL = device->MaxL
  MaxW = device->MaxW

  lVar = nil
  wVar = nil
  multVar = nil

  if( (mult = cdfParseFloatString(cdfgData->m->value)) &&
      typep(mult) != 'flonum then
    multVar = t
  )
  if( mult == " " || mult == "" then
    artError("Multiplier value must be a positive integer - set to default" )
    cdfgData->m->value = "1"
    mult = 1
    multVar = nil
  )
  if(!multVar && (mult = fix(mult)) < 1 then
    artError("Multiplier value must be a positive integer - set to default" )
    cdfgData->m->value = "1"
    mult = 1
    multVar = nil
  )

  lVal = cdfParseFloatString(cdfgData->l->value)
  wVal = cdfParseFloatString(cdfgData->w->value)

  if(typep(lVal) != 'flonum then
    lVar = t
  )
  if(lVar && deGetEditCellView()->cellViewType == "maskLayout" then...

```

## inductor

### CDF Callback

```

procedure( gpd180_indCB()
  prog( (
    cell grid scale sGrid libName deviceName device
    wVal sVal rVal indVal
    MinR MaxR MinS MaxS MinW MaxW
    k1 k2 u0 nr s w r)

  setq(cell or(cdfgData->id->master cdfgData->id))
  libName = or(cdfgData->id->libName cdfgData->id->lib->name )
  deviceName = or(cdfgData->id->cellName cdfgData->id->name )
  setq(device PasGetPdkParams(libName deviceName))
  grid = device->grid
  scale = device->scale || 1e-6
  sGrid = grid * scale

  MinR = device->MinR
  MaxR = device->MaxR
  MinS = device->MinS
  MaxS = device->MaxS
  MinW = device->MinW
  MaxW = device->MaxW

; Check Radius Values
  rVal = cdfParseFloatString(cdfgData->rad->value)
  unless( rVal && (typep(rVal) == 'flonum)
    artError("Inductor Radius value must be a number - set to default" )
    cdfgData->rad->value = cdfgData->rad->defValue
  )
  rVal = evalstring(cdfgData->rad->value)
  if(rVal < MinR then
    artError("Radius is smaller than minimum - setting to min")
    rVal = MinR
  )
  if(rVal > MaxR then
    artError("Radius is larger than maximum - setting to max")
    rVal = MaxR
  )
  r = gpd180_PDKsnapToGrid(rVal sGrid)

; Check Space Values
  sVal = cdfParseFloatString(cdfgData->space->value)
  unless( sVal && (typep(sVal) == 'flonum)
    artError("Inductor Space value must be a number - set to default" )
    cdfgData->space->value = cdfgData->space->defValue
  )
  sVal = evalstring(cdfgData->space->value)
  if(sVal < MinS then
    artError("Space is smaller than minimum - setting to min")
    sVal = MinS...

```

## bipolar

### CDF Callback

```

procedure( gpd180_bjtCB()
let((
  cell grid scale sGrid libName deviceName device
  Ewidth area MinE MaxE mult multVar)

;get cdf data
setq(cell or(cdfgData->id->master cdfgData->id))
libName = or(cdfgData->id->libName cdfgData->id->lib->name )
deviceName = or(cdfgData->id->cellName cdfgData->id->name )
setq(device PasGetPdkParams(libName deviceName))
grid = device->grid
scale = device->scale || 1e-6
sGrid = grid * scale

MinE = device->MinE
MaxE = device->MaxE

;get parameter values

if( (mult = cdfParseFloatString(cdfgData->m->value)) &&
    typep(mult) != 'flonum then
  multVar = t
)
if( mult == " " || mult == "" then
  artError("Multiplier value must be a positive integer - set to default" )
  cdfgData->m->value = "1"
  mult = 1
  multVar = nil
)
if(!multVar && (mult = fix(mult)) < 1 then
  artError("Multiplier value must be a positive integer - set to default" )
  cdfgData->m->value = "1"
  mult = 1
  multVar = nil
)

Ewidth = cdfParseFloatString(cdfgData->Ewidth->value)

unless( Ewidth && (typep(Ewidth) == 'flonum)
  artError("Emitter Width value must be a number - set to default" )
  cdfgData->Ewidth->value = cdfgData->Ewidth->defValue
  Ewidth = cdfParseFloatString( cdfgData->Ewidth->value )
)

;calculate and check min E -> this is also the value of E length

if((Ewidth - MinE) < -.001 then
  artError("Emitter Width is below minimum - setting to min")...
```

## nmoscap / pmoscap

### CDF Callback

```

procedure(gpdk180_moscapCB()
let((
  cell grid scale sGrid libName deviceName device
  MinL MaxL MinW MaxW MaxF CapA CapP
  cVar wVar fwVar IVar fingersVar
  m fingers c totalC l w wtemp fw entryModeC entryModeW)

.....
...
... retrieve process information necessary for callback procedure
...
.....

setq(cell or(cdfgData->id~>master cdfgData->id))
libName = or(cdfgData~>id~>libName cdfgData~>id~>lib~>name )
deviceName = or(cdfgData~>id~>cellName cdfgData~>id~>name )
setq(device PasGetPdkParams(libName deviceName))
grid = device->grid
scale = device->scale || 1e-6
sGrid = grid * scale

MinL = device->MinL
MinW = device->MinW
MaxL = device->MaxL
MaxW = device->MaxW
MaxF = device->MaxF
CapA = device->CapA
CapP = device->CapP

scale = device->scale
setq(sGrid times(grid scale))

.....
...
... set initial variable status of capacitance, width, length, and fingers
...
.....

cVar = nil
wVar = nil
fwVar = nil
IVar = nil
fingersVar = nil

.....
...
... determine entry mode for capacitance and width values
...
.....

```

## Component PCell Code

### Common Pcell Procedures

#### Skill Procedures

```
procedure( FpCeiling( value grid "ff" )  
  ;; Returns the smallest multiple of grid not smaller than the given value  
  ceiling((value - grid/1000)/grid) * grid  
)  
  
procedure( Xor( a b "gg" )  
  (not(a) && b) || (a && not(b))  
)
```





# pdio

pcell gpd180 pdio

PCell formal parameters

w	float	cdf
l	float	cdf

ImpEncAnode = {4C}

OxideSp = 0.5

ImpEncCathode = {3C}

CathodeEncCont = 0.2

ContSp = {6B}

ContW = {6A}

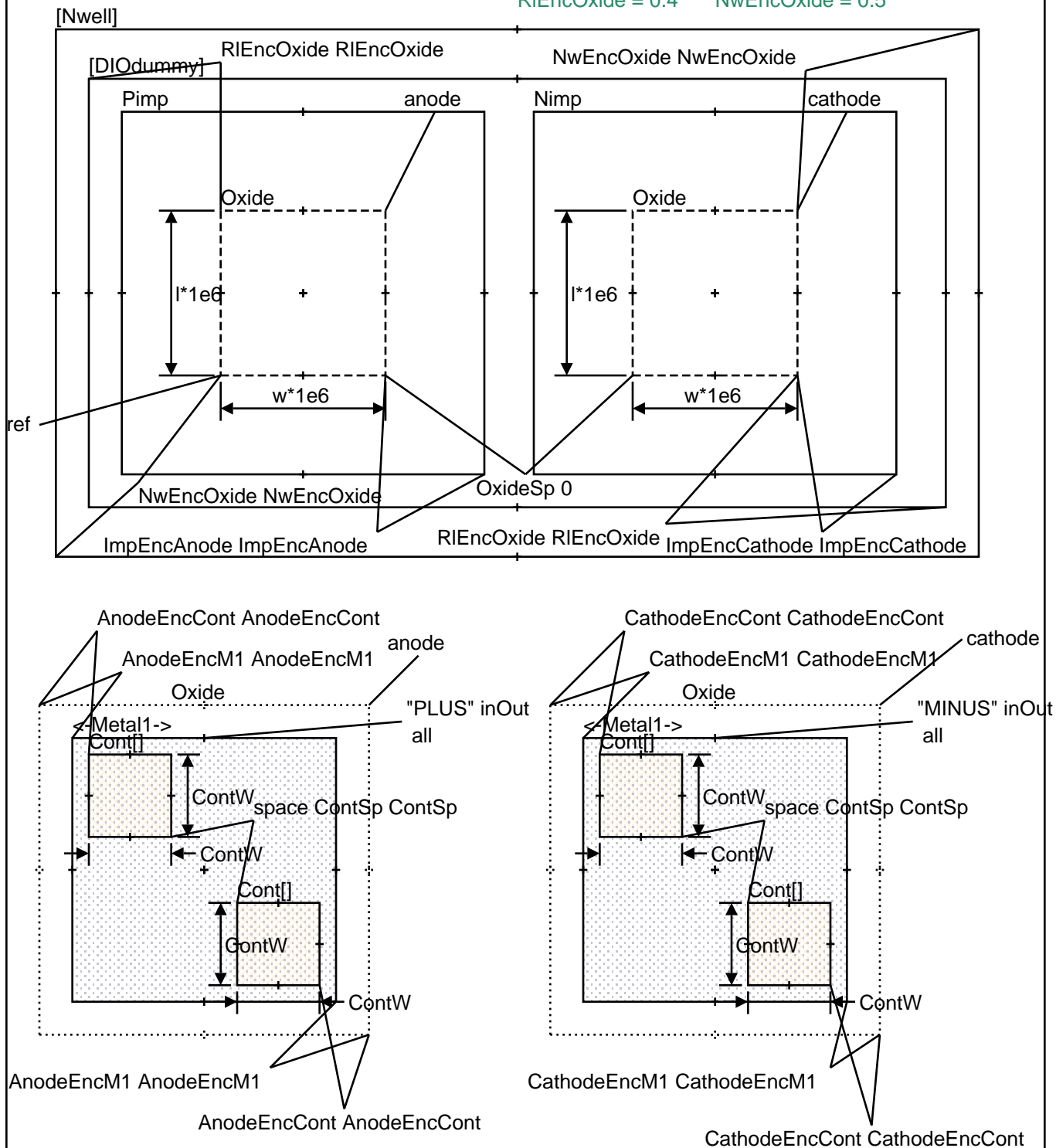
M1EncCont = {7C}

AnodeEncCont = 0.2

AnodeEncM1 = AnodeEncCont - M1EncCont

CathodeEncM1 = CathodeEncCont - M1EncCont

RIEncOxide = 0.4 NwEncOxide = 0.5



# inductor

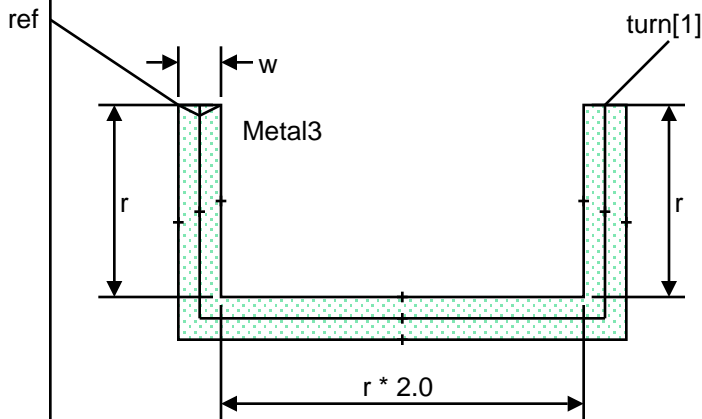
pcell gpd180 ind

PCell formal params

nr	string	cdf
rad	float	cdf
space	float	cdf
width	float	cdf

```

w = width * 1e6
s = space * 1e6
r = rad * 1e6
n = cdfParseFloatString(nr)
numHalfTurns = fix(n * 2.0)
MinusTabL = RIEnclnd
RIEnclnd = 10.0
RI2Enclnd = 0.0
MetEncVia = 0.9
PlusTabL = fix(n) * (s + w) + RIEnclnd
    
```

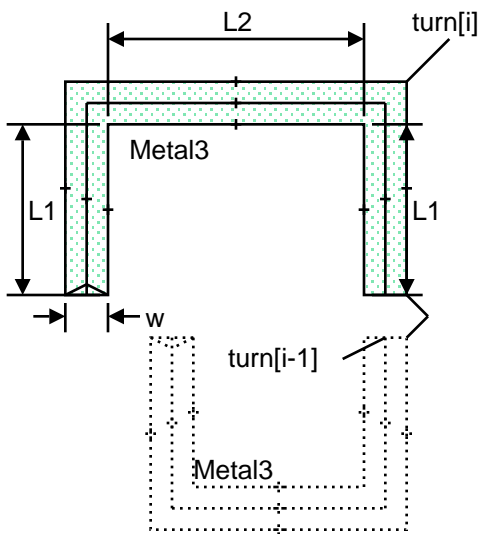


2: for i 2 numHalfTurns

if evenp(i)

```

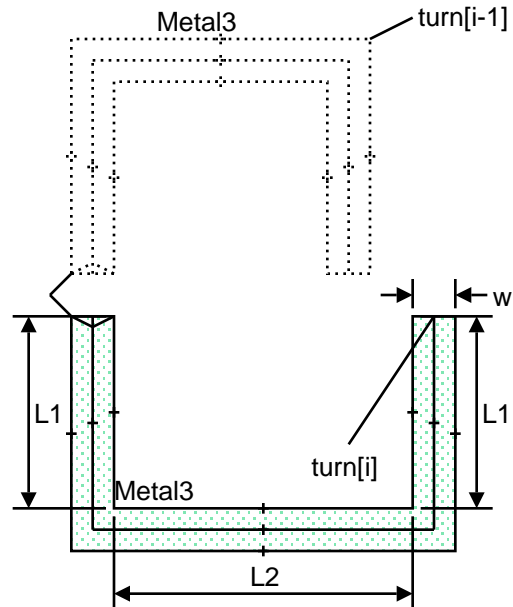
L1 = r + (i / 2 - 1) * (s + w)
L2 = r * 2.0 + (i - 1) * (s + w)
    
```



if oddp(i)

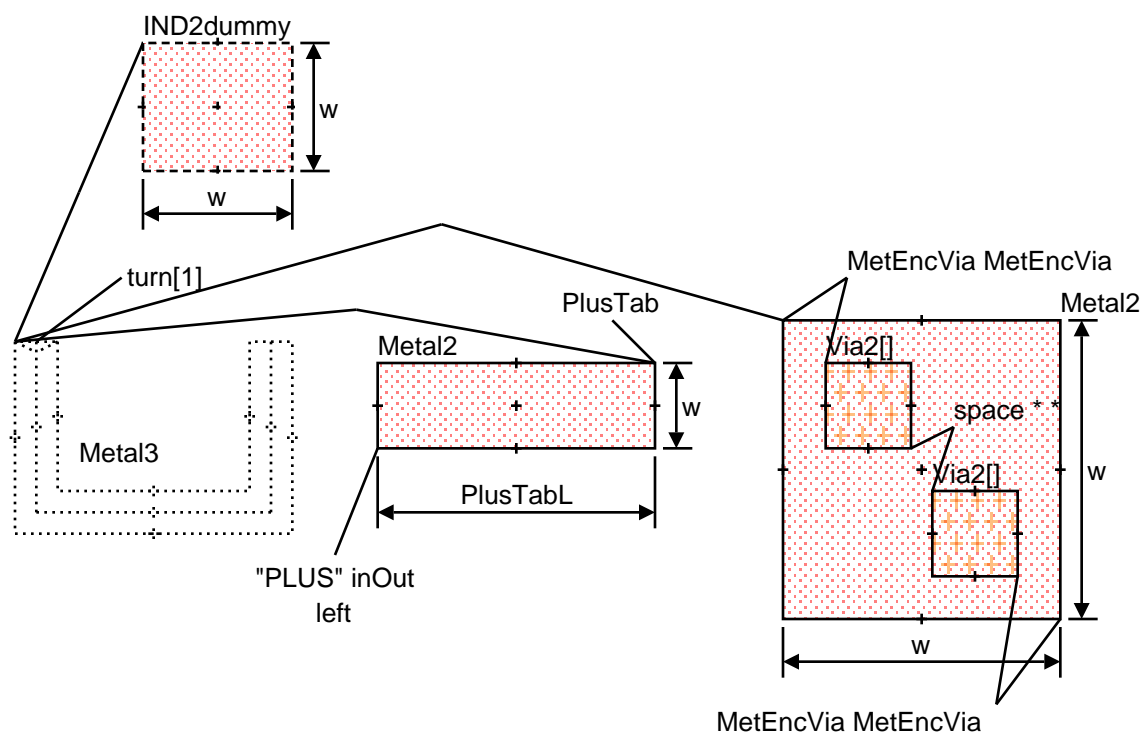
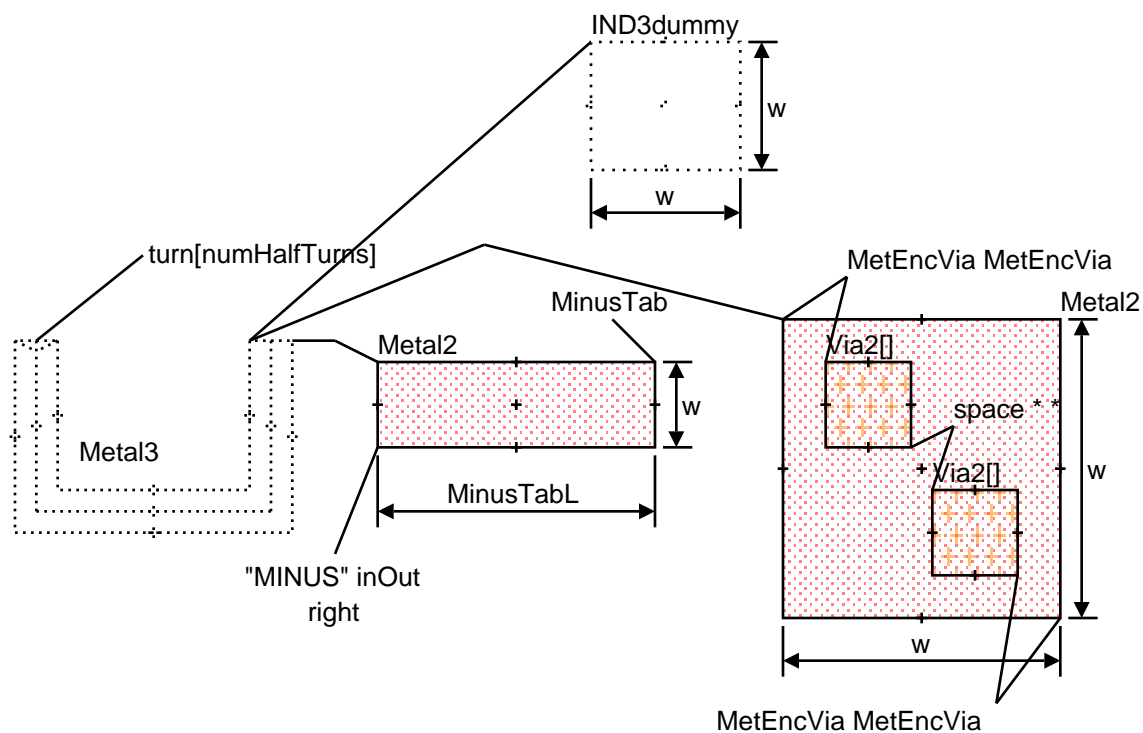
```

L1 = r + (i - 1) / 2.0 * (s + w)
L2 = r * 2.0 + (i - 1) * (s + w)
    
```



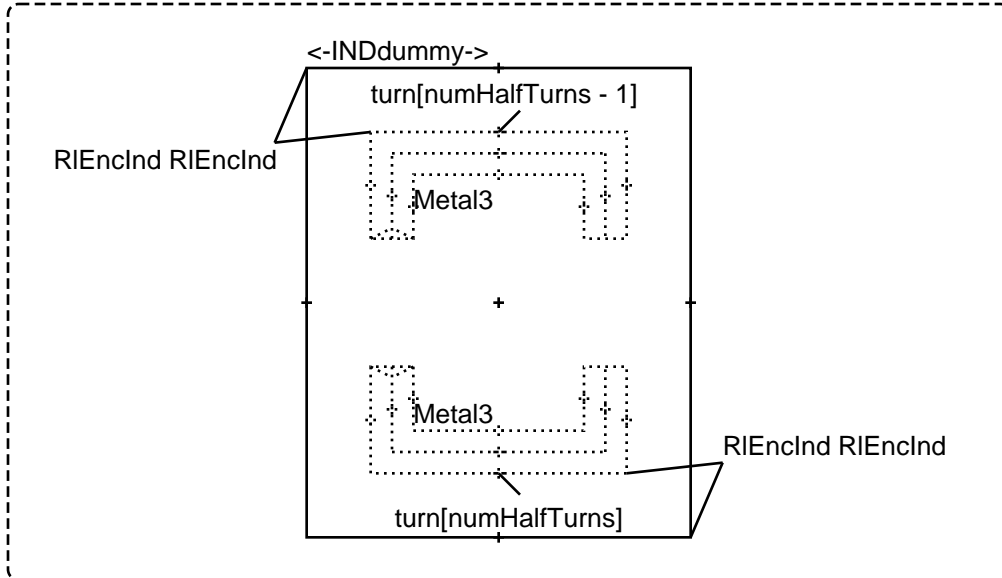
pcell gpd180 ind

3: include



pcell gpd180 ind

4: include



# npn

pcell gpd180 npn

PCell formal params

<b>Ewidth</b>	float	cdf
---------------	-------	-----

alEmit = 0.6

meImpIDiff = {3C}

msEmitBase = 0.5

msCollBase = 1.0

meWellDiff = {2C}

msCont = {6B}

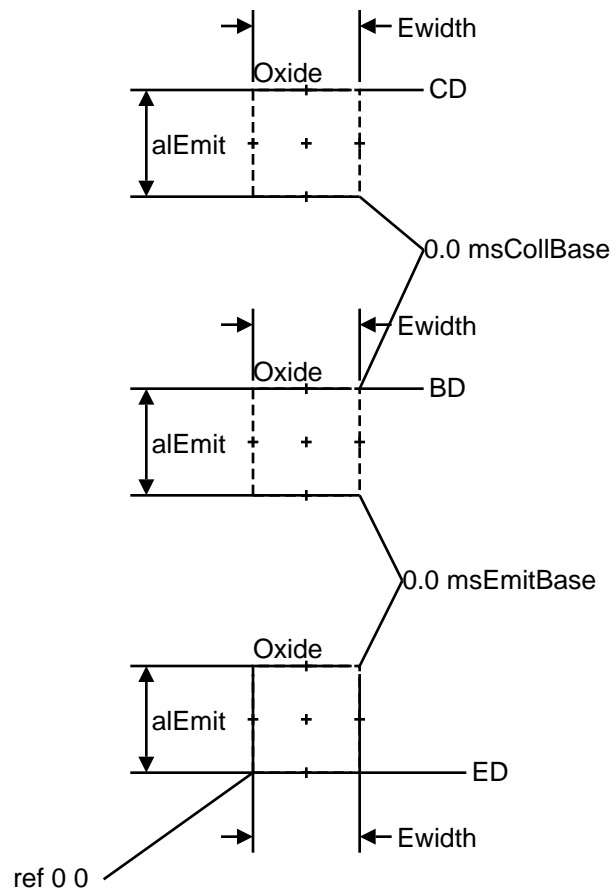
mwCont = {6A}

meDiffCont = {6C}

meM1Cont = {7C}

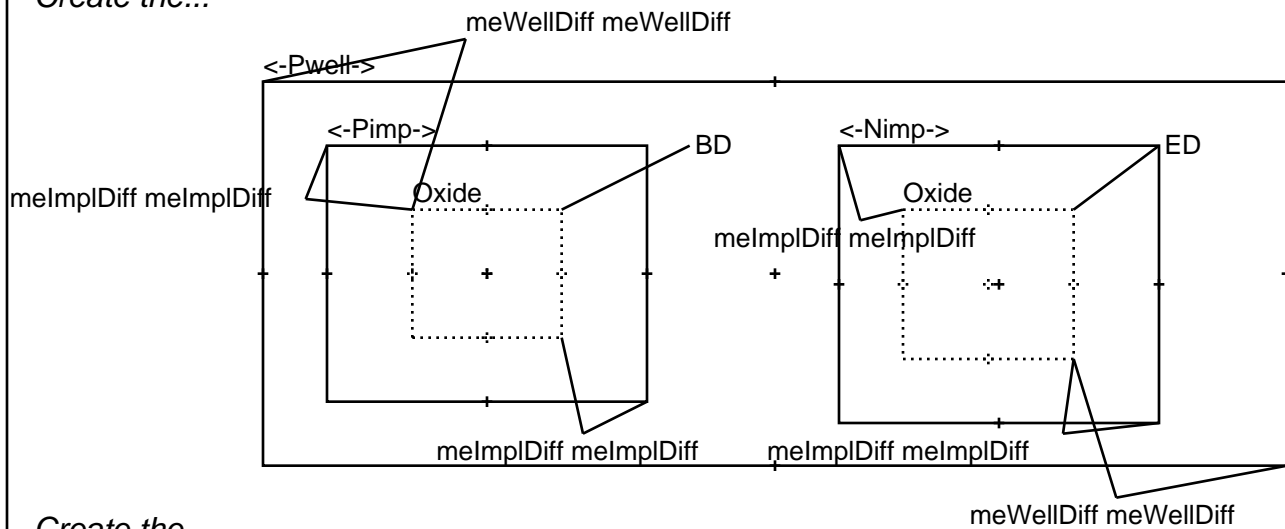
meThreshDiff = 0.8

meM1Diff = meM1Cont - meDiffCont

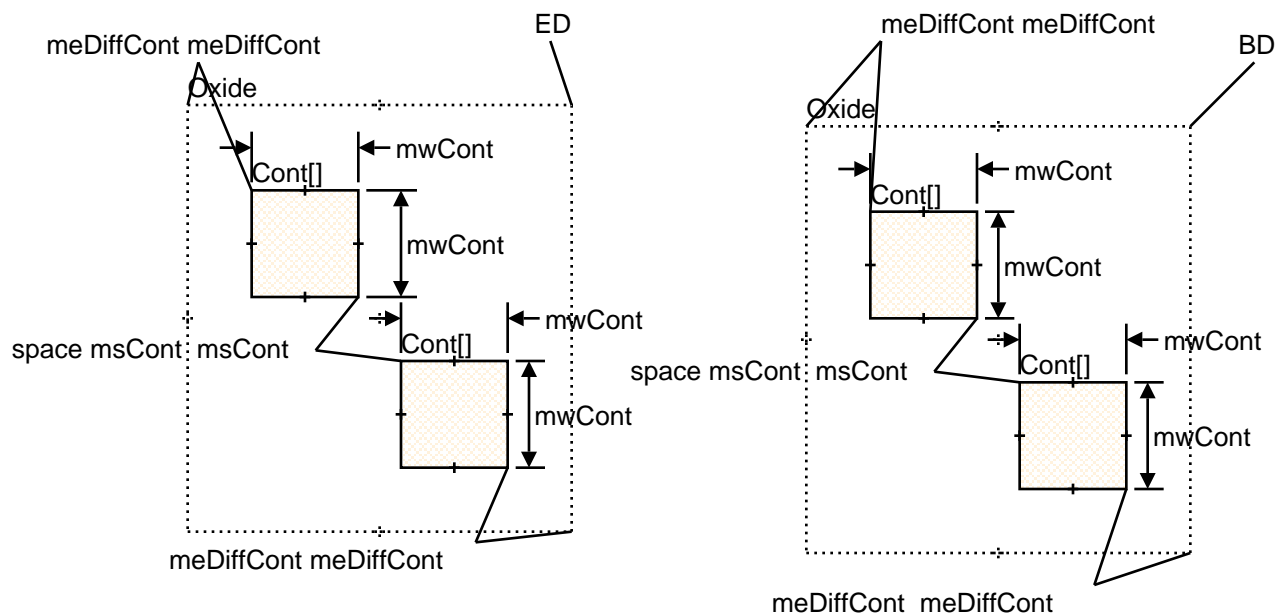


pcell gpd180 npn

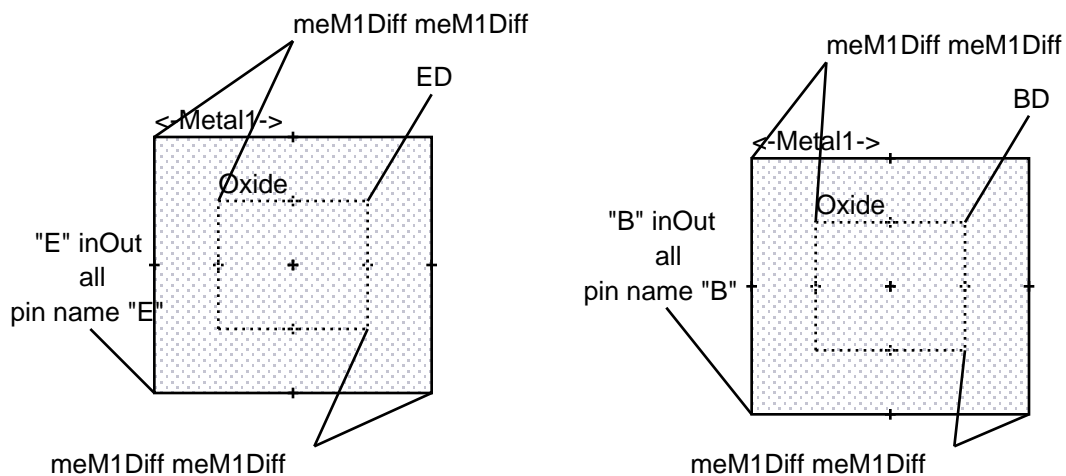
Create the...



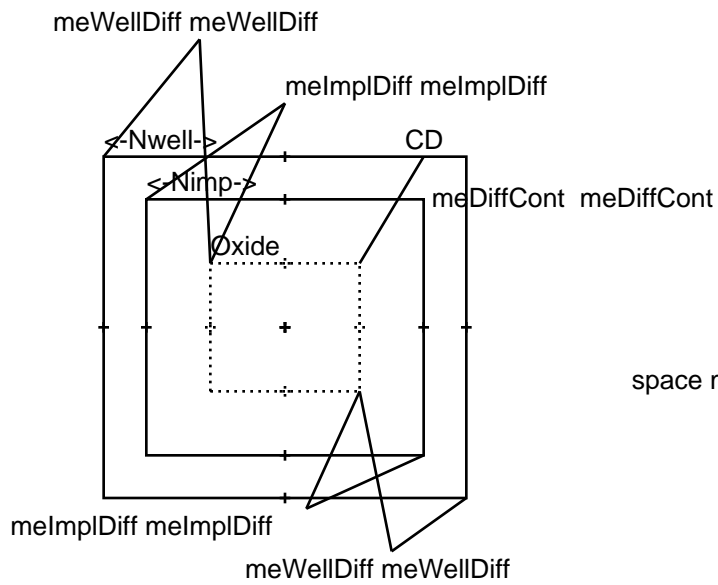
Create the...



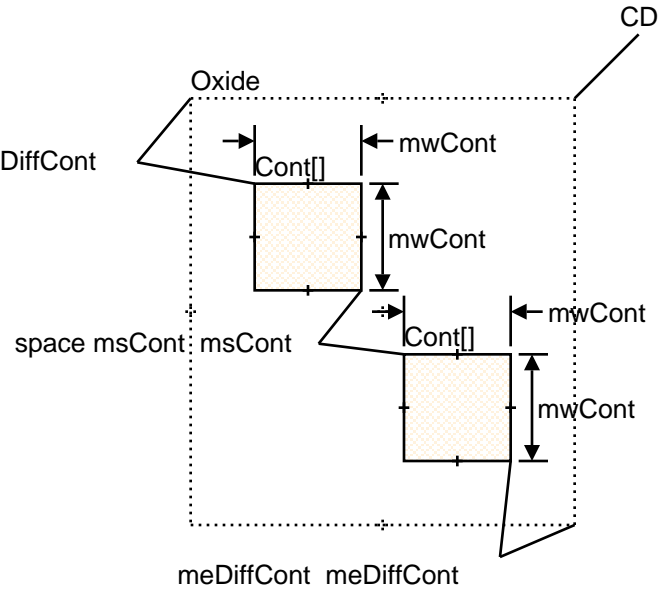
Create the metal...



pcell gpd180 npn

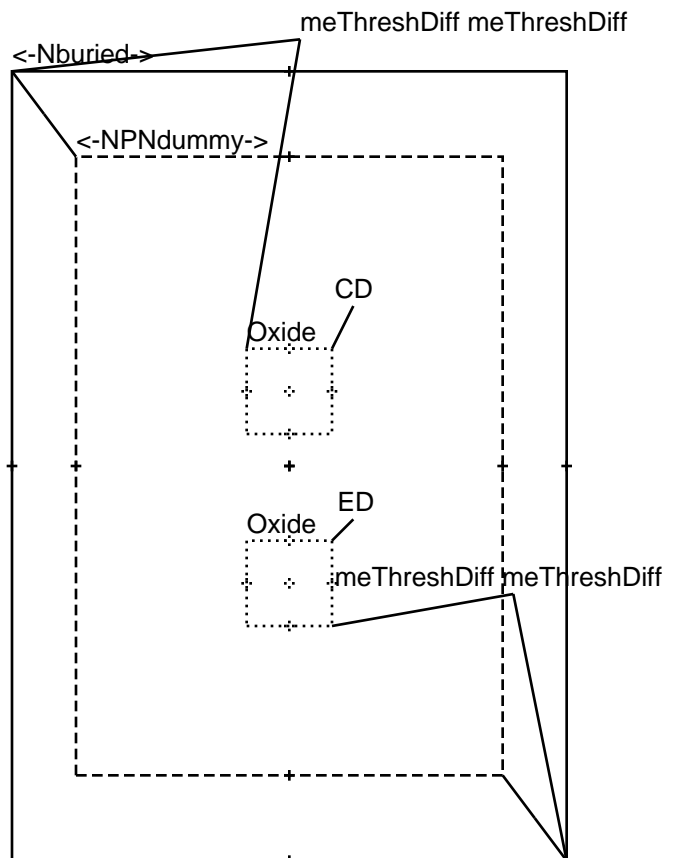
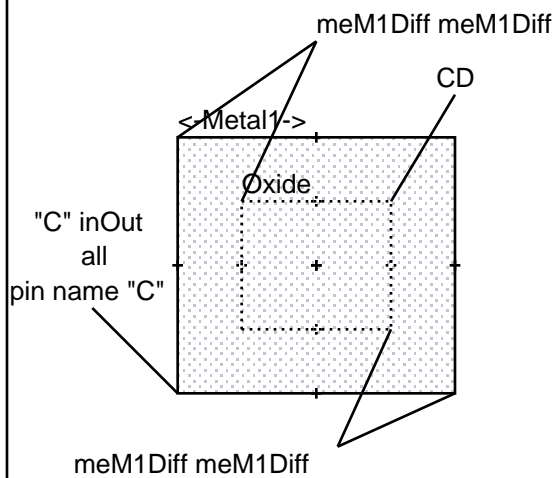


Create the collector contacts:



Create the PWel, NBuried, and the NPNdummy:

Create the metal...



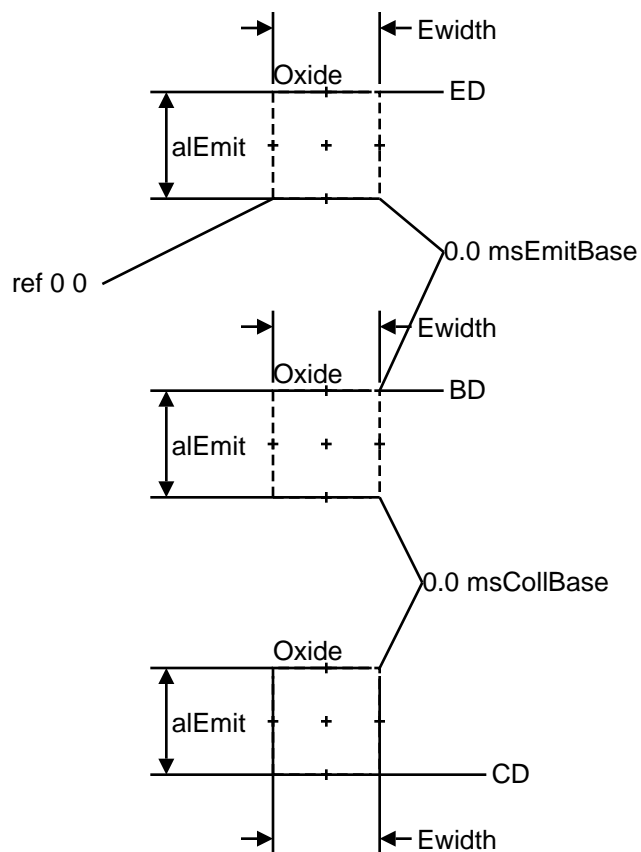
# pnp

pcell gpd180 pnp

PCell formal params

<b>Ewidth</b>	float	cdf
---------------	-------	-----

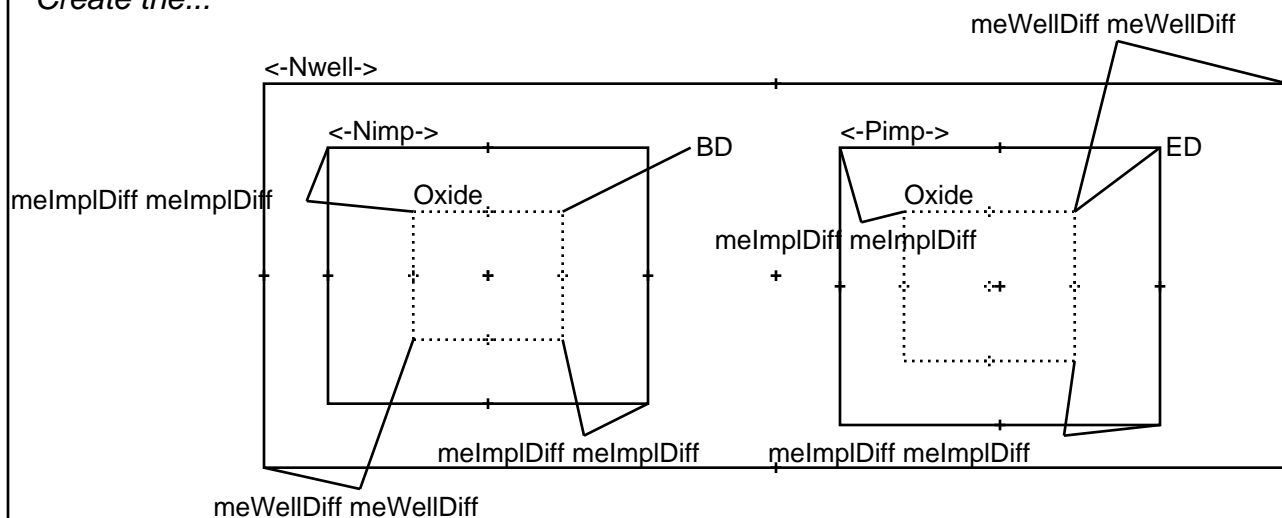
$alEmit = 0.6$   
 $meImplDiff = \{3C\}$   
 $msEmitBase = 0.5$   
 $msCollBase = 1.0$   
 $meWellDiff = \{2C\}$   
 $msCont = \{6B\}$   
 $mwCont = \{6A\}$   
 $meDiffCont = \{6C\}$   
 $meM1Cont = \{7C\}$   
 $meBaseDiff = 0.8$   
 $meThreshDiff = 1.1$   
 $meM1Diff = meM1Cont - meDiffCont$



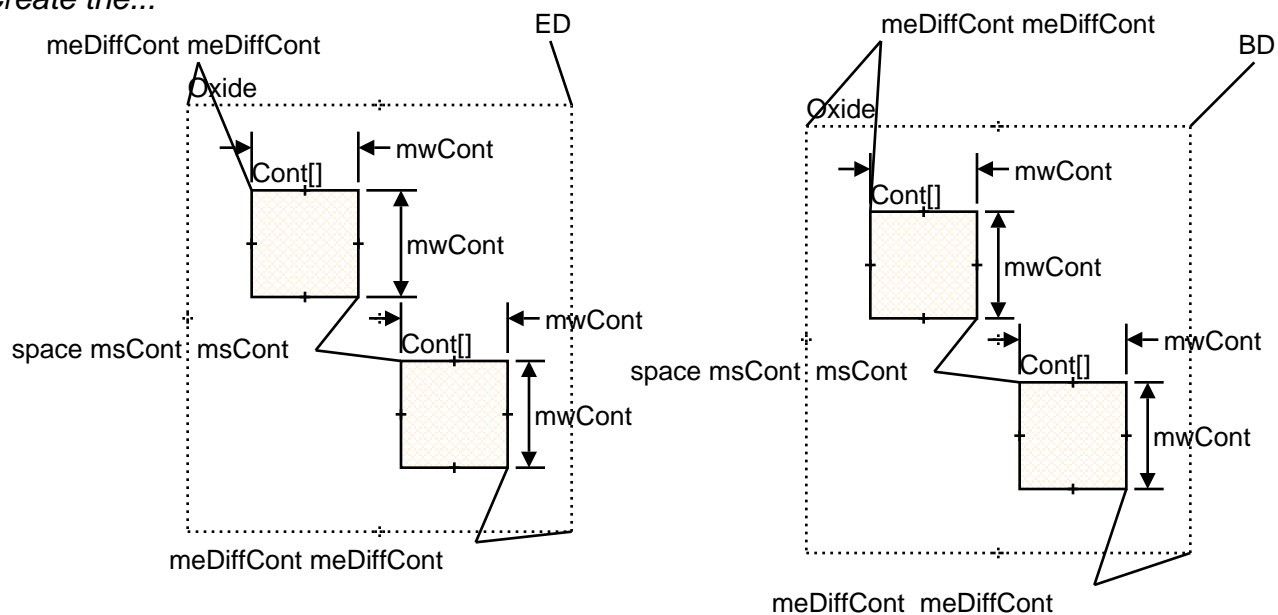


pcell gpdk180 pnp

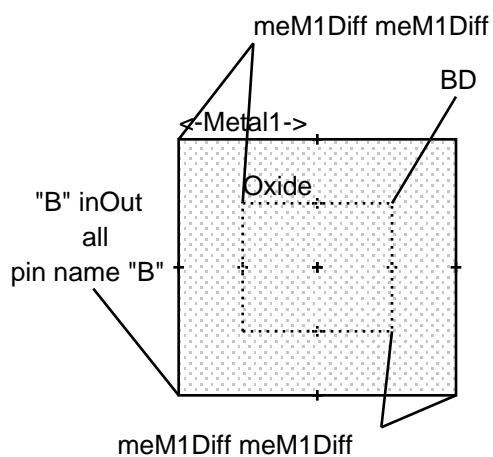
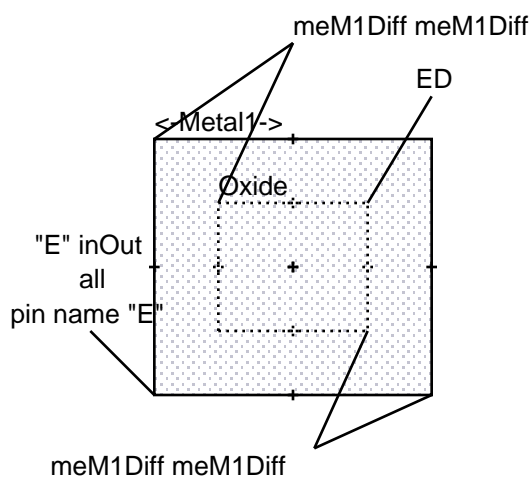
Create the...



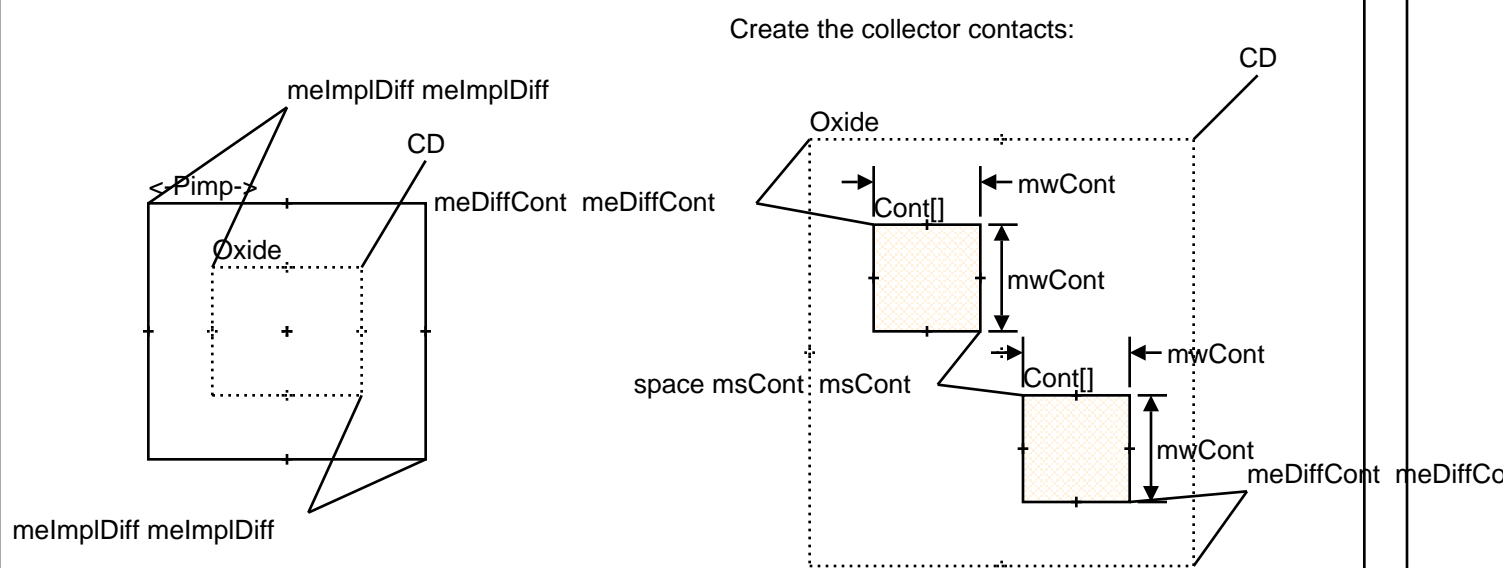
Create the...



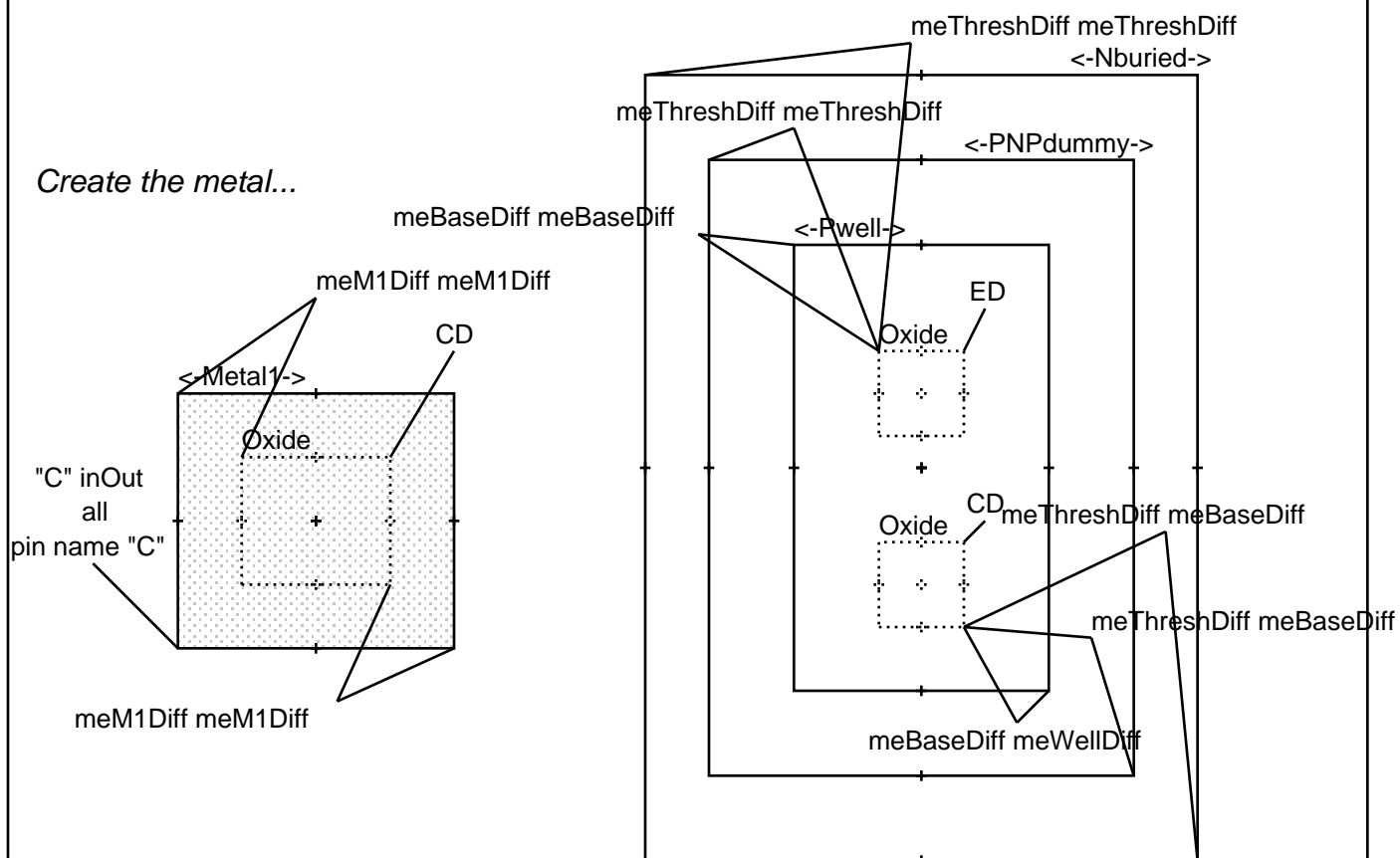
Create the metal...



pcell gpdk180 pnp



Create the PWel, NBuried, and the NPNdummy:



**vpnp**

pcell gpd180 vpnp

PCell formal params

<b>EmitterSize</b>	string	cdf
--------------------	--------	-----

*These are the variables to control the pcell. The first section will map to DRC rules and the second section are derived values.*

```
Ewidth = evalstring(
    substring( EmitterSize 1
    difference(
        strlen( EmitterSize )
        strlen( index( EmitterSize "x" ) ) ) ) ) )
```

```
Elength = evalstring( substring( index( EmitterSize "x" ) 2 ) )
```

```
mePimpDiff = {4C}          msCont = {6B}
meNimpDiff = {3C}          mwCont = {6A}
msPimpNimp = 0.4           meDiffCont = {6C}
msNimpPimp = 0.4           meM1Cont = {7C}
meWellDiff = {2C}         meBJTDum = 0.05
msWellDiff = {2D}
```

```
meWellPimp = max( difference( meWellDiff mePimpDiff ) 0.0 )
meWellNimp = max( difference( meWellDiff meNimpDiff ) 0.0 )
msWellPimp = max( difference( msWellDiff mePimpDiff ) mePimpDiff )
msWellNimp = max( difference( msWellDiff meNimpDiff ) meNimpDiff )
msEmitBase = mePimpDiff + msPimpNimp + meNimpDiff
msBaseColl = meNimpDiff + meWellNimp + msWellPimp + mePimpDiff
meDiffM1 = meDiffCont - meM1Cont

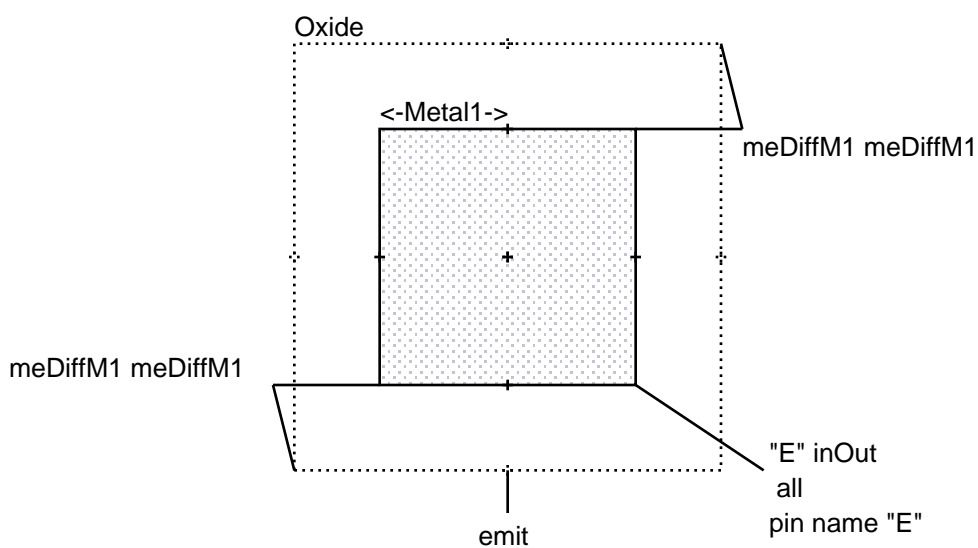
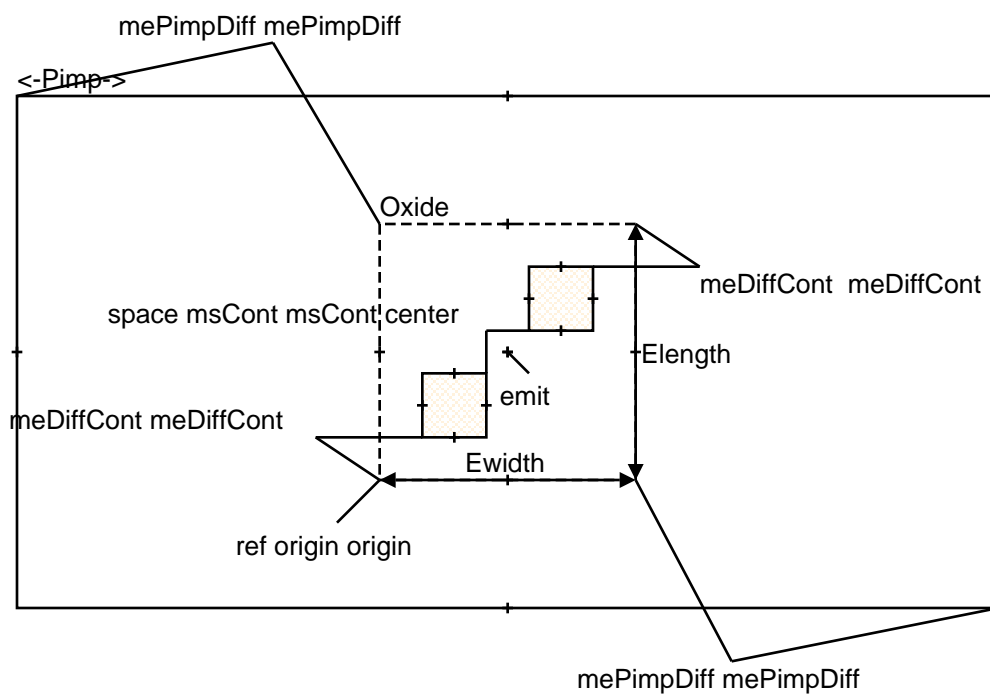
mwBase = mwCont + 2.0 * meDiffCont
mwColl = mwCont + 2.0 * meDiffCont
mwM1 = mwCont + 2.0 * meM1Cont
mwNimp = mwBase + 2.0 * meNimpDiff
mwPimp = mwBase + 2.0 * mePimpDiff

meBJTM1 = meDiffM1 + mePimpDiff + meBJTDum
```

pcell gpd180 vnp

*This page defines the...*

$\text{origin} = \text{msEmitBase} + \text{mwBase} + \text{msBaseColl} + \text{mwColl} + \text{mePimpDiff} + \text{meBJTDum}$

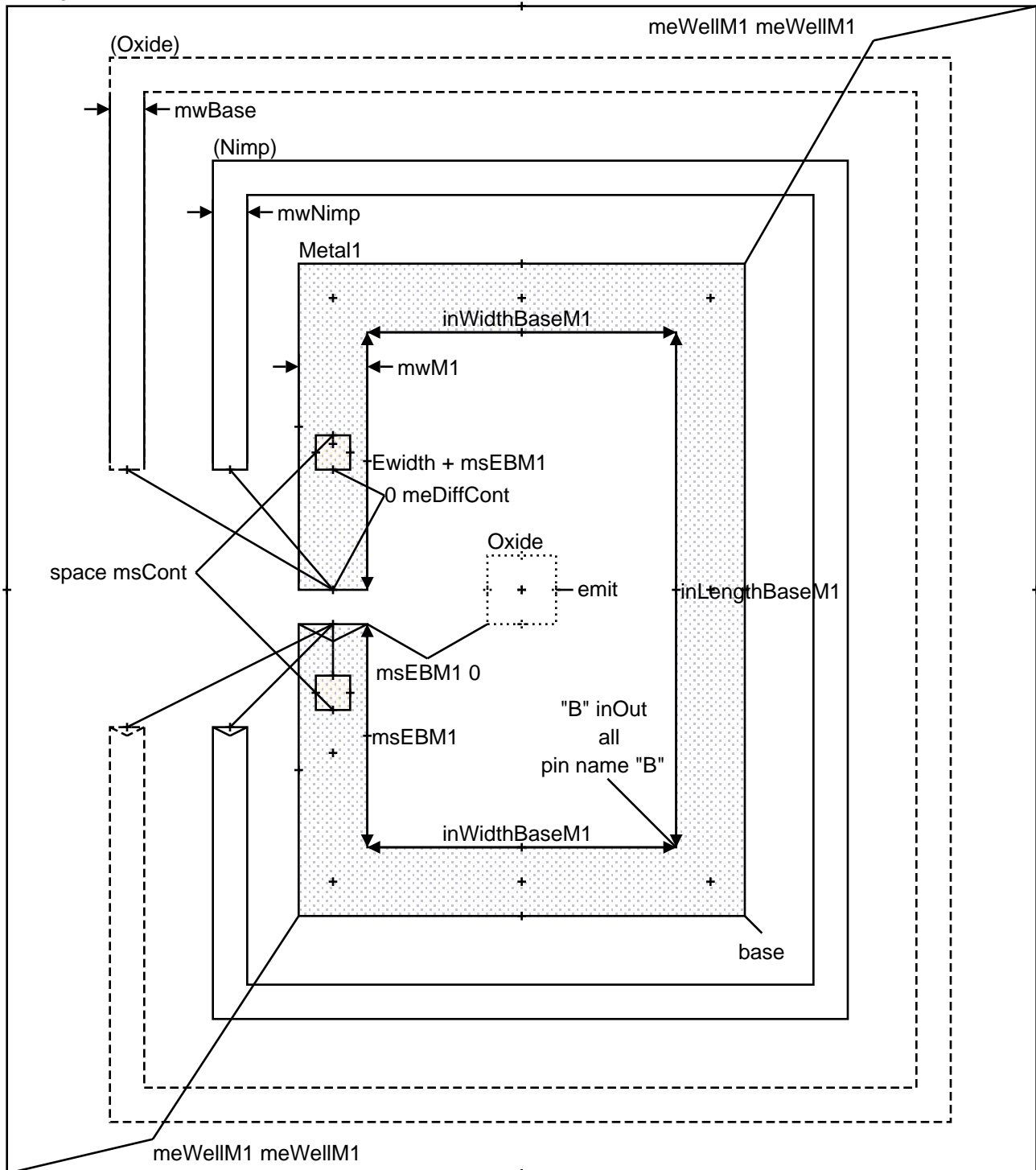


pcell gpd180 vnp

*This page defines the base.*

$msEBM1 = msEmitBase + meDiffM1$   
 $inWidthBaseM1 = Ewidth + 2.0 * msEBM1$   
 $inLengthBaseM1 = Elength + 2.0 * msEBM1$   
 $meWellM1 = meWellDiff + meDiffM1$

<-Nwell->



pcell gpd180 vnp

*This page defines the...*

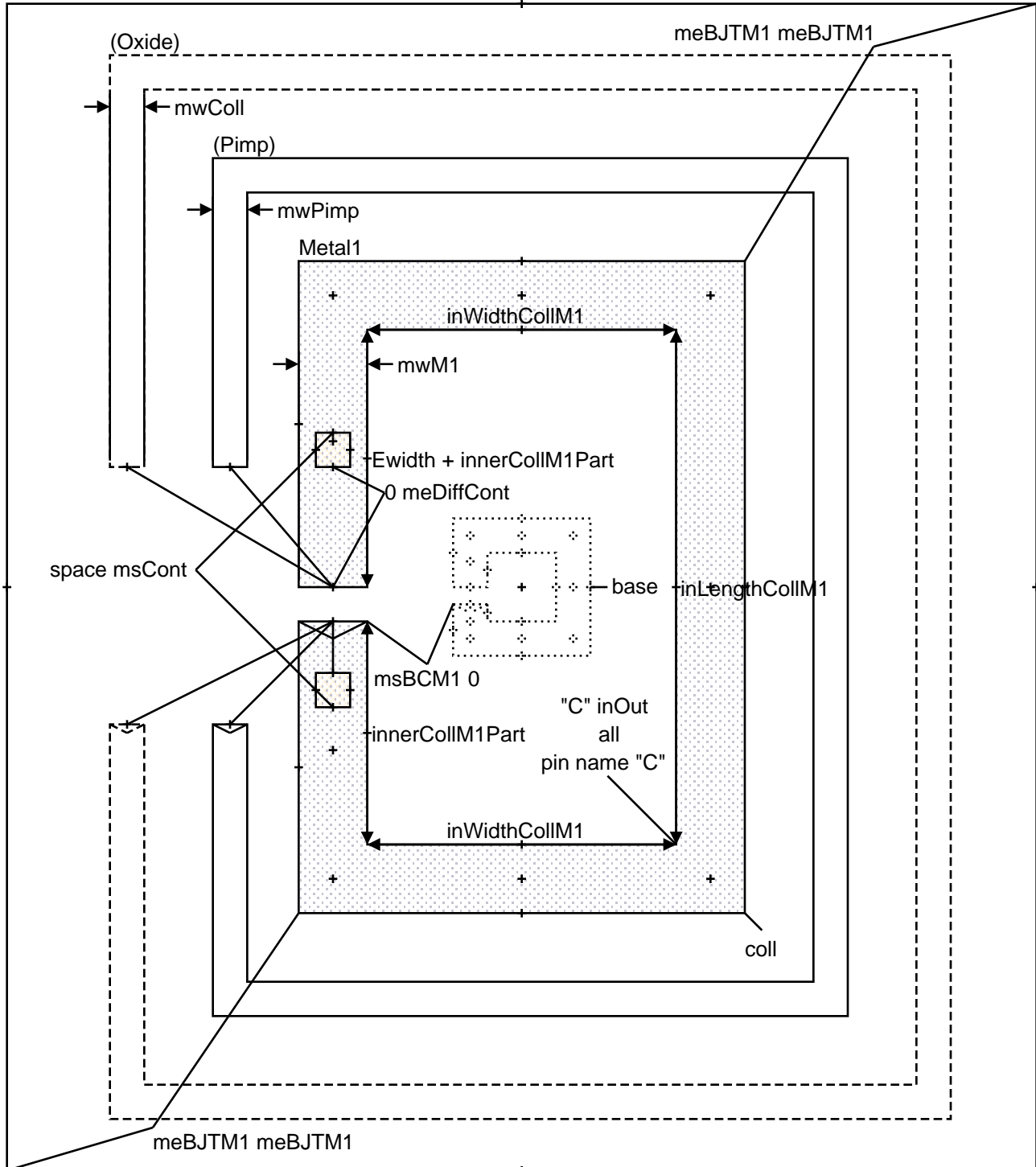
$$msBCM1 = msBaseColl + 2 * meDiffM1$$

$$innerCollM1Part = msEmitBase + mwBase + msBaseColl + meDiffM1$$

$$inWidthCollM1 = Ewidth + 2.0 * innerCollM1Part$$

$$inLengthCollM1 = Elength + 2.0 * innerCollM1Part$$

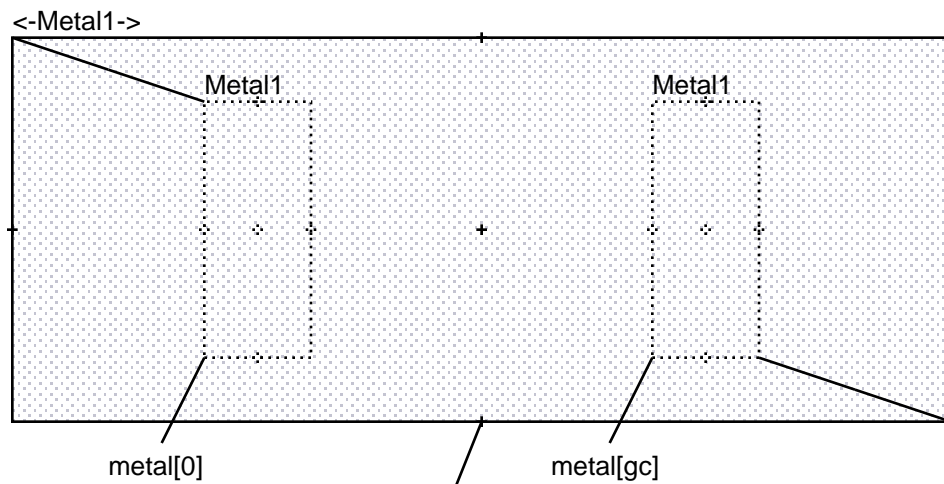
&lt;-BJTdum-&gt;



## moscap (macro)

pcell macro moscap

999: include



"D" inOut  
all  
must group drain

**nmoscap (uses nmos pcell)**

pcell gpd180 nmoscap

PCell formal params

<b>fw</b>	float	CDF
<b>l</b>	float	CDF
<b>fingers</b>	float	CDF
<b>connectGates</b>	string	CDF
<b>connectSD</b>	string	CDF
<b>switchSD</b>	boolean	CDF
<b>mtlCvg</b>	string	CDF
<b>sdMtlWidth</b>	float	CDF
<b>leftAbut</b>	int	CDF
<b>rightAbut</b>	int	CDF
<b>tap</b>	string	CDF
<b>topTap</b>	boolean	CDF
<b>bottomTap</b>	boolean	CDF
<b>leftTap</b>	boolean	CDF
<b>rightTap</b>	boolean	CDF
<b>tapExtension</b>	string	CDF
<b>tapCntRows</b>	int	CDF

include macro mos

include macro mos\_nplus

include macro mos\_nbl\_tap

include macro mos\_cap\_tap

include macro mosTap

include macro mosTap\_pplus

include macro mosltap\_pplus

**nmoscap3 (uses nmos pcell)**

pcell gpd180 nmoscap3

PCell formal params

<b>fw</b>	float	CDF
<b>l</b>	float	CDF
<b>fingers</b>	float	CDF
<b>connectGates</b>	string	CDF
<b>connectSD</b>	string	CDF
<b>switchSD</b>	boolean	CDF
<b>mtlCvg</b>	string	CDF
<b>sdMtlWidth</b>	float	CDF
<b>leftAbut</b>	int	CDF
<b>rightAbut</b>	int	CDF
<b>tap</b>	string	CDF
<b>topTap</b>	boolean	CDF
<b>bottomTap</b>	boolean	CDF
<b>leftTap</b>	boolean	CDF
<b>rightTap</b>	boolean	CDF
<b>tapExtension</b>	string	CDF
<b>tapCntRows</b>	int	CDF

include macro mos

include macro mos\_nplus

include macro mos\_nbl\_tap

include macro mos\_cap\_tap

include macro mosTap

include macro mosTap\_pplus

include macro mosltap\_pplus

include macro moscap



**pmoscap (uses pmos pcell)**

pcell gpd180 pmoscap

PCell formal params

<b>fw</b>	float	CDF
<b>l</b>	float	CDF
<b>fingers</b>	float	CDF
<b>connectGates</b>	string	CDF
<b>connectSD</b>	string	CDF
<b>switchSD</b>	boolean	CDF
<b>mtlCvg</b>	string	CDF
<b>sdMtlWidth</b>	float	CDF
<b>leftAbut</b>	int	CDF
<b>rightAbut</b>	int	CDF
<b>tap</b>	string	CDF
<b>topTap</b>	boolean	CDF
<b>bottomTap</b>	boolean	CDF
<b>leftTap</b>	boolean	CDF
<b>rightTap</b>	boolean	CDF
<b>tapExtension</b>	string	CDF
<b>tapCntRows</b>	int	CDF

```
include macro mos
include macro mos_pplus
include macro mos_nw_tap
include macro mos_cap_tap
include macro mosTap
include macro mosTap_nplus
include macro mosltap_nplus
```

**pmoscap3 (uses pmos pcell)**

pcell gpd180 pmoscap3

PCell formal params

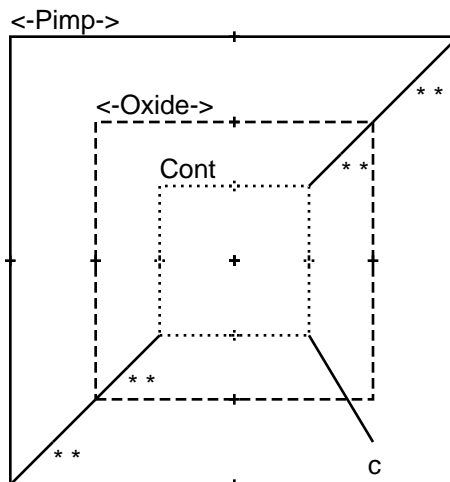
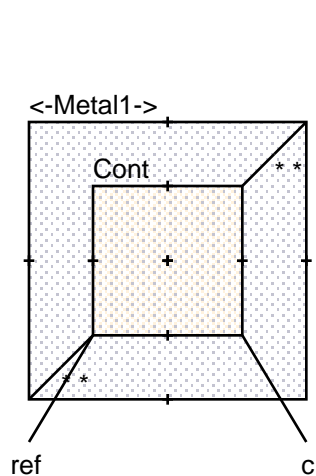
<b>fw</b>	float	CDF
<b>l</b>	float	CDF
<b>fingers</b>	float	CDF
<b>connectGates</b>	string	CDF
<b>connectSD</b>	string	CDF
<b>switchSD</b>	boolean	CDF
<b>mtlCvg</b>	string	CDF
<b>sdMtlWidth</b>	float	CDF
<b>leftAbut</b>	int	CDF
<b>rightAbut</b>	int	CDF
<b>tap</b>	string	CDF
<b>topTap</b>	boolean	CDF
<b>bottomTap</b>	boolean	CDF
<b>leftTap</b>	boolean	CDF
<b>rightTap</b>	boolean	CDF
<b>tapExtension</b>	string	CDF
<b>tapCntRows</b>	int	CDF

```
include macro mos
include macro mos_pplus
include macro mos_nw_tap
include macro mos_cap_tap
include macro mosTap
include macro mosTap_nplus
include macro mosltap_nplus
include macro moscap
```

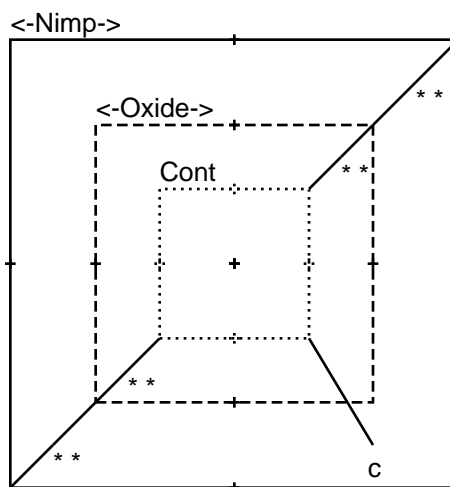
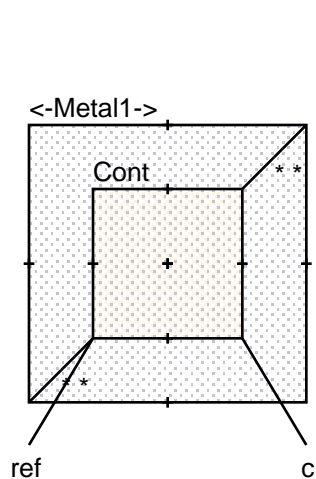
## **Basic Test Structures**

# Taps

passcell gpd180\_layouts ptap



passcell gpd180\_layouts ntap

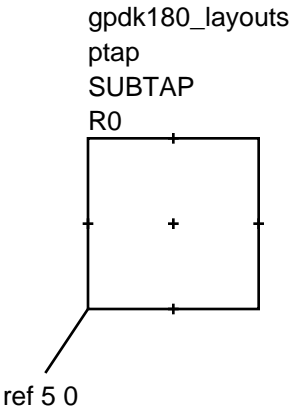
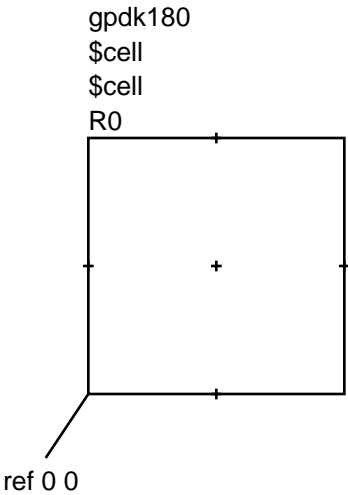


# Create Layouts

passcell gpd180\_layouts placer

Pcell Macro Table

\$lib	\$cell
gpd180_layouts	nmos
gpd180_layouts	nmos3
gpd180_layouts	pmos
gpd180_layouts	pmos3
gpd180_layouts	nplusres
gpd180_layouts	pplusres
gpd180_layouts	nwellres
gpd180_layouts	polyres
gpd180_layouts	polyhres
gpd180_layouts	mimcap
gpd180_layouts	nmoscap
gpd180_layouts	pmoscap
gpd180_layouts	ind
gpd180_layouts	ndio
gpd180_layouts	pdio
gpd180_layouts	npn
gpd180_layouts	pnp
gpd180_layouts	vpnp



CDB layers

ResWdum	0	ResWdum
Oxide	1	OXIDE
Nwell	2	NWELL
Poly	3	POLY
Nimp	4	NIMP
Pimp	5	PIMP
Cont	6	CONT
Metal1	7	METAL1
Via1	8	VIA1
Metal2	9	METAL2
Via2	10	VIA2
Metal3	11	METAL3
Capdum	12	CAPDUM
Resdum	13	RESDUM
CapMetal	14	CAPMETA
BJTdum	15	BJTDUM
INDdummy	16	INDdumm
IND2dummy	17	IND2dum

CDB layers

Pwell	18	PWELL
Nburied	19	NBURIED
NPNdummy	20	NPNDUMM
PNPdummy	21	PNPDUMM
DIOdummy	22	DIODUMM
SiProt	23	SiProt
ThickOxide	24	ThickOxide
Via3	30	VIA3
Metal4	31	METAL4
Via4	32	VIA4
Metal5	33	METAL5
Via5	34	VIA5
Metal6	35	METAL6
Bondpad	36	Bondpad
WellBody	50	WELLBOD

CDB layers

Psubiso	51	PSUBISO
scaPort	66	scaP
scaNwell	67	scaNW
scaNburied	68	scaNB
scaSelect	69	scaSel
IND3dummy	70	IND3dum
M1dummy	71	M1dum
M2dummy	72	M2dum
M3dummy	73	M3dum
M4dummy	74	M4dum
M5dummy	75	M5dum
M6dummy	76	M6dum
allGeoShare	100	allGeo
OVERLAP	101	OVERLAP

CDB layers

Unrouted	200	Unroute
Row	201	Row
Group	202	Group
Cannotoccupy	203	noOccupy
Canplace	204	Canplac
hardFence	205	hardFnc
softFence	206	softFnc
y0	207	y0
y1	208	y1
y2	209	y2
y3	210	y3
y4	211	y4
y5	212	y5
y6	213	y6
y7	214	y7
y8	215	y8
y9	216	y9

CDB layers









































designFlow	217	dsnFlow
stretch	218	stretch
edgeLayer	219	edgeLyr
changedLayer	220	chngLyr
unset	221	unset
unknown	222	unknown
spike	223	spike
hiz	224	hiz
resist	225	resist
drive	226	drive
supply	227	supply
wire	228	wire
pin	229	pin








































CDB layers

text	230	text
device	231	device
border	232	border
snap	233	snap
align	234	align
prBoundary	235	prBndry
instance	236	instnce
annotate	237	anotate
marker	238	marker
select	239	select
grid	251	grid
axis	252	axis
hilite	253	hilite
background	254	bkgground





























## CDB purposes










































GeoShare	1	GeoShare
warning	234	wng
tool1	235	tl1
tool0	236	tl0
label	237	lbl
flight	238	flt
error	239	err
annotate	240	ant
drawing1	241	dr1
drawing2	242	dr2
drawing3	243	dr3
drawing4	244	dr4
drawing5	245	dr5
drawing6	246	dr6
drawing7	247	dr7
drawing8	248	dr8
drawing9	249	dr9
boundary	250	bnd
pin	251	pin
drawing	252	drw
net	253	net
cell	254	cel
all	255	all






































	\$layer1	dummy	
	\$layer2	dummy	
	\$pcLayer1	dummy	
	ALL_GATES	( Poly and NPLUS andnot Nwell ) or ( Poly and PPLUS and Nwell )	device_recognition
	BJTdum	input 15;0 df2order 104 packet zbp	
	Bondpad	input 36;0 df2order 96 packet pass	
	Bondpad_boundary	input (Bondpad boundary) df2order 97 packet pbaseBnd notChg notValid	
	Cannotoccupy	input df2order 208 notChg notValid	
	Cannotoccupy_boundary	input (Cannotoccupy boundary) df2order 209 packet CannotoccupyBnd notChg notValid	
	Canplace	input df2order 210 notChg notValid	
	CapMetal	input 14;0 df2order 66 packet m4	
	CapMetal_boundary	input (CapMetal boundary) df2order 68 packet m4 notChg notValid	
	CapMetal_net	input (CapMetal net) df2order 67 packet m4 notChg notValid	
	Capdum	input 12;0 df2order 98 packet zcap	
	Cont	input 6;0 df2order 69 packet cw	via
	Cont_boundary	input (Cont boundary) df2order 72 packet cwBnd notChg notValid	
	Cont_net	input (Cont net) df2order 71 packet cwNet notChg notValid	
	Cont_pin	input (Cont pin) df2order 70 packet cwPin notChg notValid	
	Contdum	Cont and Metal1	
	DIOdummy	input 22;0 df2order 107 packet zdiode	
	FIELD_POLY1	Poly andnot ALL_GATES	
	FIELD_POLY1_CONNECTION	FIELD_POLY1 and Poly	via
	Group	input df2order 206 notChg notValid	
	Group_label	input (Group label) df2order 207 packet GroupLbl notChg notValid	
	IND2dummy	input 17;0 df2order 100 packet zind2	
	IND3dummy	input 70;0 df2order 101 packet zind3	
	INDUCTOR	(Metal3 or IND2dummy or IND3dummy) and INDdummy	device_recognition
	INDdummy	input 16;0 df2order 99 packet zind	
	INDdummy_net	input (INDdummy net) packet zindnet notChg notValid	
	INDterm1	INDdummy and IND2dummy	
	INDterm1Cont	Metal2 and INDdummy and IND2dummy	via
	INDterm2	INDdummy and IND3dummy	
	INDterm2Cont	Metal2 and INDdummy and IND3dummy	via
	ISONMOS	POLYterm andnot Nwell and NPLUS andnot ThickOxide andnot Capdum and Nburied	device_recognition
	ISONMOSCAP	POLYterm andnot Nwell and NPLUS and Capdum and Nburied	device_recognition
	ISONMOSHV	POLYterm andnot Nwell and NPLUS and ThickOxide andnot Capdum and Nburied andnot RFdummy	device_recognition
	ISONMOSRF	POLYterm andnot Nwell and NPLUS and ThickOxide andnot Capdum and Nburied and RFdummy	device_recognition
	ISONSDRES	NSD and Resdum and Nburied	device_recognition
	ISOPWELL	((Nburied enclose (holes Nwell)) and (holes Nwell))andnot Pwell	
	JVAR1dummy	input 43;0 packet zjvar1	device_recognition


































 JVAR1dummy_net	input (JVAR1dummy net) packet zjvar1 notChg notValid	
 JVAR2dummy	input 44;0 packet zjvar2	device_recognition
 JVAR3dummy	input 48;0 packet zjvar3	device_recognition
 JVARNF	JVAR1dummy and Nwell	device_recognition
 JVARW40	JVAR2dummy and Nwell	device_recognition
 JVARanode	(JVAR1dummy or JVAR2dummy) andnot NSDterm	
 JVARterm	PSDterm and (JVAR1dummy or JVAR2dummy) and JVAR3dummy	soft_via
 M1dummy	input 37;0 df2order 103 packet m1dum	
 M1res	Metal1 and M1dummy	device_recognition
 M1term	Metal1 andnot M1dummy	
 M2dummy	input 38;0 df2order 103 packet m2dum	
 M2res	Metal2 and M2dummy	device_recognition
 M2term	Metal2 andnot M2dummy	
 M3dummy	input 39;0 df2order 103 packet m3dum	
 M3res	Metal3 and M3dummy	device_recognition
 M3term	Metal3 andnot M3dummy	
 M4dummy	input 40;0 df2order 103 packet m4dum	
 M4res	Metal4 and M4dummy	device_recognition
 M4term	Metal4 andnot M4dummy	
 M5dummy	input 41;0 df2order 103 packet m5dum	
 M5res	Metal5 and M5dummy	device_recognition
 M5term	Metal5 andnot M5dummy	
 M6dummy	input 42;0 df2order 103 packet m6dum	
 M6res	Metal6 and M6dummy	device_recognition
 M6term	Metal6 andnot M6dummy	
 MIMCAP	CapMetal and Metal2 and Capdum	device_recognition
 Metal1	input 7;0 df2order 2 packet m1 Attach Text: 7;3 (Metal1 label)	
 Metal1_boundary	input (Metal1 boundary) df2order 5 packet m1Bnd notChg notValid	
 Metal1_drawing4	input 7;4 (Metal1 drawing4) df2order 1 packet m1 notChg notValid	
 Metal1_label	input 7;3 (Metal1 label) df2order 0 packet m1 notChg notValid	
 Metal1_net	input 7;2 (Metal1 net) df2order 4 packet m1Net notChg notValid	
 Metal1_pin	input 7;1 (Metal1 pin) df2order 3 packet m1Pin notChg notValid fillStyle X	
 Metal2	input 9;0 df2order 14 packet m2 Attach Text: 9;3 (Metal2 label)	
 Metal2_boundary	input (Metal2 boundary) df2order 17 packet m2Bnd notChg notValid	
 Metal2_drawing4	input 9;4 (Metal2 drawing4) df2order 13 packet m2 notChg notValid	
 Metal2_label	input 9;3 (Metal2 label) df2order 12 packet m2 notChg notValid	
 Metal2_net	input 9;2 (Metal2 net) df2order 16 packet m2Net notChg notValid	
 Metal2_pin	input 9;1 (Metal2 pin) df2order 15 packet m2Pin notChg notValid fillStyle X	
 Metal3	input 11;0 df2order 26 packet m3 Attach Text: 11;3 (Metal3 label)	



 Metal3_boundary	input (Metal3 boundary) df2order 29 packet m3Bnd notChg notValid	
 Metal3_drawing4	input 11;4 (Metal3 drawing4) df2order 25 packet m3 notChg notValid	
 Metal3_label	input 11;3 (Metal3 label) df2order 24 packet m3 notChg notValid	
 Metal3_net	input 11;2 (Metal3 net) df2order 28 packet m3Net notChg notValid	
 Metal3_pin	input 11;1 (Metal3 pin) df2order 27 packet m3Pin notChg notValid fillStyle X	
 Metal4	input 31;0 df2order 38 packet m4 Attach Text: 31;3 (Metal4 label)	
 Metal4_boundary	input (Metal4 boundary) df2order 41 packet m4Bnd notChg notValid	
 Metal4_drawing4	input 31;4 (Metal4 drawing4) df2order 37 packet m4 notChg notValid	
 Metal4_label	input 31;3 (Metal4 label) df2order 36 packet m4 notChg notValid	
 Metal4_net	input 31;2 (Metal4 net) df2order 40 packet m4Net notChg notValid	
 Metal4_pin	input 31;1 (Metal4 pin) df2order 39 packet m4Pin notChg notValid fillStyle X	
 Metal5	input 33;0 df2order 50 packet m5 Attach Text: 33;3 (Metal5 label)	
 Metal5_boundary	input (Metal5 boundary) df2order 53 packet m5Bnd notChg notValid	
 Metal5_drawing4	input 33;4 (Metal5 drawing4) df2order 49 packet m5 notChg notValid	
 Metal5_label	input 33;3 (Metal5 label) df2order 48 packet m5 notChg notValid	
 Metal5_net	input 33;2 (Metal5 net) df2order 52 packet m5Net notChg notValid	
 Metal5_pin	input 33;1 (Metal5 pin) df2order 51 packet m5Pin notChg notValid fillStyle X	
 Metal6	input 35;0 df2order 62 packet m6 Attach Text: 35;3 (Metal6 label)	
 Metal6_boundary	input (Metal6 boundary) df2order 65 packet m6Bnd notChg notValid	
 Metal6_drawing4	input 35;4 (Metal6 drawing4) df2order 61 packet m6 notChg notValid	
 Metal6_label	input 35;3 (Metal6 label) df2order 60 packet m6 notChg notValid	
 Metal6_net	input 35;2 (Metal6 net) df2order 64 packet m6Net notChg notValid	
 Metal6_pin	input 35;1 (Metal6 pin) df2order 63 packet m6Pin notChg notValid fillStyle X	
 NBVIA	NWELLterm and Nburied	soft_via
 NDIODE	DIOdummy and ( NSDterm andnot Nwell )	device_recognition
 NMOS	POLYterm andnot Nwell and NPLUS andnot ThickOxide andnot Capdum andnot Nburied	device_recognition
 NMOSCAP	POLYterm andnot Nwell and NPLUS and Capdum andnot Nburied	device_recognition
 NMOSHV	POLYterm andnot Nwell and NPLUS and ThickOxide andnot Capdum andnot Nburied andnot RFdummy	device_recognition

 NMOSRF	POLYterm andnot Nwell and NPLUS and ThickOxide andnot Capdum andnot Nburied and RFdummy	device_recognition
 NPLUS	Nimp and Oxide	
 NPN	NPNdummy and Pwell and Nburied and NSDterm	device_recognition
 NPNdummy	input 20;0 df2order 105 packet znpn	
 NSD	NPLUS andnot Poly	
 NSDRES	NSD and Resdum andnot Nburied	device_recognition
 NSDarea	(NPLUS cut Poly) andnot Poly	
 NSDcont	Cont and NPLUS	via
 NSDterm	NSD andnot Resdum	
 NWELLRES	Nwell and ResWdum	device_recognition
 NWELLterm	Nwell andnot ResWdum	
 NWVIA	NSDterm and NWELLterm	soft_via
 Nburied	input 19;0 df2order 88 packet npblk	
 Nburied_boundary	input (Nburied boundary) df2order 90 packet npblkBnd notChg notValid	
 Nburied_net	input (Nburied net) df2order 89 packet npblk notChg notValid	
 Nimp	input 4;0 df2order 91 packet nplus	
 Nimp_boundary	input (Nimp boundary) df2order 92 packet nplusBnd notChg notValid	
 Nimp_net	input (Nimp net) packet nplusBnd notChg notValid	
 Nwell	input 2;0 df2order 79 packet nwell	
 Nwell_boundary	input (Nwell boundary) df2order 81 packet nwellBnd notChg notValid	
 Nwell_net	input (Nwell net) df2order 80 packet nwellNet notChg notValid	
 OVERLAP	input 101;0 df2order 116 packet ovlap notSel notValid	
 OVERLAP_boundary	input 101;5 (OVERLAP boundary) df2order 118 packet ovlap notSel notChg	
 OVERLAP_label	input 101;3 (OVERLAP label) df2order 117 packet ovlap notSel notChg notValid	
 Oxide	input 1;0 df2order 85 packet tox	
 Oxide_boundary	input (Oxide boundary) df2order 87 packet toxBnd notChg notValid	
 Oxide_net	input (Oxide net) df2order 86 packet toxBnd notChg notValid	
 PDIODE	DIODummy and PSDterm and Nwell	device_recognition
 PMOS	POLYterm and Nwell and PPLUS andnot ThickOxide andnot Capdum	device_recognition
 PMOSCAP	POLYterm and Nwell and PPLUS and Capdum	device_recognition
 PMOSHV	POLYterm and Nwell and PPLUS and ThickOxide andnot Capdum andnot RFdummy	device_recognition
 PMOSRF	POLYterm and Nwell and PPLUS and ThickOxide andnot Capdum and RFdummy	device_recognition
 PNP	PNPdummy and Nwell and Pwell and Nburied and PSDterm	device_recognition
 PNPdummy	input 21;0 df2order 106 packet znpn	
 POLYHRES	Poly and Resdum and SiProt	device_recognition
 POLYRES	Poly and Resdum andnot SiProt	device_recognition
 POLYcont	Cont and Poly	via
 POLYterm	Poly andnot Resdum	
 PPLUS	Pimp and Oxide	
 PSD	PPLUS andnot Poly	
 PSDRES	PPLUS and Resdum and Nwell	device_recognition






































	PSDarea	(PPLUS cut Poly) andnot Poly	
	PSDcont	Cont and PPLUS	via
	PSDterm	PSD andnot Resdum	
	PSUB	(bulk andnot (Nburied or Nwell or Pwell)) or BJTdum	
	PWNBVIA	(PSDterm and ISOPWELL)	soft_via
	PWVIA	(PSDterm and Pwell) andnot Nwell	soft_via
	Pimp	input 5;0 df2order 93 packet pplus	
	Pimp_boundary	input (Pimp boundary) df2order 94 packet pplusBnd notChg notValid	
	Pimp_net	input (Pimp net) packet pplusBnd notChg notValid	
	Poly	input 3;0 df2order 75 packet poly1	
	Poly_boundary	input (Poly boundary) df2order 78 packet poly1Bnd notChg notValid	
	Poly_drawing4	input 3;4 (Poly drawing4) df2order 74 packet poly1 notChg notValid	
	Poly_label	input 3;3 (Poly label) df2order 73 packet poly1 notChg notValid	
	Poly_net	input 3;2 (Poly net) df2order 77 packet poly1Net notChg notValid	
	Poly_pin	input 3;1 (Poly pin) df2order 76 packet poly1Pin notChg notValid	
	Psubiso	input 51;0 df2order 109	
	Pwell	input 18;0 df2order 82 packet pwell	
	Pwell_boundary	input (Pwell boundary) df2order 84 packet pwellBnd notChg notValid	
	Pwell_net	input (Pwell net) df2order 83 packet pwellNet notChg notValid	
	RFdummy	input 69;0 df2order 102 packet zrf	
	ResWdum	input 71;0 df2order 102 packet zrwell	
	Resdum	input 13;0 df2order 102 packet zrpoly	
	Row	input df2order 204 packet prBoundaryLbl notChg	
	Row_label	input (Row label) df2order 205 packet RowLbl notChg notValid	
	SUBVIA	(PSUB and PSDterm) andnot Nburied andnot Nwell	soft_via
	SiProt	input 72;0 df2order 95 packet siprot	
	ThickOxide	input 24;0 df2order 85 packet thox	
	Unrouted	input df2order 211 notChg notValid	
	Unrouted_drawing1	input (Unrouted drawing1) df2order 212 packet Unrouted1 notChg notValid	
	Unrouted_drawing2	input (Unrouted drawing2) df2order 213 packet Unrouted2 notChg notValid	
	Unrouted_drawing3	input (Unrouted drawing3) df2order 214 packet Unrouted3 notChg notValid	
	Unrouted_drawing4	input (Unrouted drawing4) df2order 215 packet Unrouted4 notChg notValid	
	Unrouted_drawing5	input (Unrouted drawing5) df2order 216 packet Unrouted5 notChg notValid	
	Unrouted_drawing6	input (Unrouted drawing6) df2order 217 packet Unrouted6 notChg notValid	
	Unrouted_drawing7	input (Unrouted drawing7) df2order 218 packet Unrouted7 notChg notValid	
	Unrouted_drawing8	input (Unrouted drawing8) df2order 219 packet Unrouted8 notChg notValid	
	Unrouted_drawing9	input (Unrouted drawing9) df2order 220 packet Unrouted9 notChg notValid	

 VNP	(PSDterm and Nwell) interact (holes(PSDterm) interact holes(NSDterm) notInteract Poly) and BJTdum	device_recognition
 Via1	input 8;0 df2order 9 packet v1	via
 Via1_boundary	input (Via1 boundary) df2order 11 packet v1Bnd notChg notValid	
 Via1_drawing4	input 8;4 (Via1 drawing4) df2order 8 packet v1 notChg notValid	
 Via1_label	input 8;3 (Via1 label) df2order 6 packet v1 notChg notValid	
 Via1_net	input (Via1 net) df2order 10 packet v1Net notChg notValid	
 Via1_pin	input 8;1 (Via1 pin) df2order 7 packet v1 notChg notValid	
 Via1dum	Via1 and Metal2	
 Via2	input 10;0 df2order 21 packet v2	via
 Via2Cap	Via2 and CapMetal	via
 Via2NoCapInd	Via2 and Metal2 andnot (CapMetal or INDdummy)	via
 Via2_boundary	input (Via2 boundary) df2order 23 packet v2Bnd notChg notValid	
 Via2_drawing4	input 10;4 (Via2 drawing4) df2order 20 packet v2 notChg notValid	
 Via2_label	input 10;3 (Via2 label) df2order 19 packet v2 notChg notValid	
 Via2_net	input (Via2 net) df2order 22 packet v2Net notChg notValid	
 Via2_pin	input 10;1 (Via2 pin) df2order 18 packet v2 notChg notValid	
 Via2dum	Via2NoCapInd and Metal3	
 Via3	input 30;0 df2order 33 packet v3	via
 Via3_boundary	input (Via3 boundary) df2order 35 packet v3Bnd notChg notValid	
 Via3_drawing4	input 30;4 (Via3 drawing4) df2order 32 packet v3 notChg notValid	
 Via3_label	input 30;3 (Via3 label) df2order 31 packet v3 notChg notValid	
 Via3_net	input (Via3 net) df2order 34 packet v3Net notChg notValid	
 Via3_pin	input 30;1 (Via3 pin) df2order 30 packet v3 notChg notValid	
 Via3dum	Via3 and Metal4	
 Via4	input 32;0 df2order 45 packet v4	via
 Via4_boundary	input (Via4 boundary) df2order 47 packet v4Bnd notChg notValid	
 Via4_drawing4	input 32;4 (Via4 drawing4) df2order 44 packet v4 notChg notValid	
 Via4_label	input 32;3 (Via4 label) df2order 43 packet v4 notChg notValid	
 Via4_net	input (Via4 net) df2order 46 packet v4Net notChg notValid	
 Via4_pin	input 32;1 (Via4 pin) df2order 42 packet v4 notChg notValid	
 Via4dum	Via4 and Metal5	
 Via5	input 34;0 df2order 57 packet v5	via
 Via5_boundary	input (Via5 boundary) df2order 59 packet v5Bnd notChg notValid	

<input type="checkbox"/> Via5_drawing4	input 34;4 (Via5 drawing4) df2order 56 packet v5 notChg notValid	
<input type="checkbox"/> Via5_label	input 34;3 (Via5 label) df2order 55 packet v5 notChg notValid	
<input type="checkbox"/> Via5_net	input (Via5 net) df2order 58 packet v5Net notChg notValid	
<input type="checkbox"/> Via5_pin	input 34;1 (Via5 pin) df2order 54 packet v5 notChg notValid	
<input type="checkbox"/> Via5dum	Via5 and Metal6	
<input type="checkbox"/> WellBody	input df2order 108 notVis notSel notChg notValid	
<input type="checkbox"/> align	input df2order 137 notChg notValid	
<input type="checkbox"/> allGeoShare	input 100;1 df2order 114 packet overlap notVis notSel notChg notValid	
<input type="checkbox"/> allGeoShare_GeoShare	input 100;0 (allGeoShare GeoShare) df2order 115 packet overlap	
<input type="checkbox"/> annotate	input df2order 122 packet annotate notChg notValid	
<input type="checkbox"/> annotate_drawing1	input (annotate drawing1) df2order 123 packet annotate1 notChg notValid	
<input type="checkbox"/> annotate_drawing2	input (annotate drawing2) df2order 124 packet annotate2 notChg notValid	
<input type="checkbox"/> annotate_drawing3	input (annotate drawing3) df2order 125 packet annotate3 notChg notValid	
<input type="checkbox"/> annotate_drawing4	input (annotate drawing4) df2order 126 packet annotate4 notChg notValid	
<input type="checkbox"/> annotate_drawing5	input (annotate drawing5) df2order 127 packet annotate5 notChg notValid	
<input type="checkbox"/> annotate_drawing6	input (annotate drawing6) df2order 128 packet annotate6 notChg notValid	
<input type="checkbox"/> annotate_drawing7	input (annotate drawing7) df2order 129 packet annotate7 notChg notValid	
<input type="checkbox"/> annotate_drawing8	input (annotate drawing8) df2order 130 packet annotate8 notChg notValid	
<input type="checkbox"/> annotate_drawing9	input (annotate drawing9) df2order 131 packet annotate9 notChg notValid	
<input type="checkbox"/> axis	input df2order 156 packet axis notSel notChg notValid	
<input checked="" type="checkbox"/> background	input df2order 119 packet background notSel notChg notDrg notValid	
<input type="checkbox"/> border	input df2order 144 packet border notChg notValid	
<input type="checkbox"/> bulk	substrate	
<input type="checkbox"/> changedLayer_tool0	input (changedLayer tool0) df2order 200 packet changedLayerTI0 notVis notSel notChg notDrg notValid	
<input type="checkbox"/> changedLayer_tool1	input (changedLayer tool1) df2order 201 packet changedLayerTI1 notVis notSel notChg notDrg notValid	
<input checked="" type="checkbox"/> designFlow	input df2order 190 packet designFlow notSel notChg notDrg	
<input checked="" type="checkbox"/> designFlow_drawing1	input (designFlow drawing1) df2order 191 packet designFlow1 notSel notChg notDrg	
<input checked="" type="checkbox"/> designFlow_drawing2	input (designFlow drawing2) df2order 192 packet designFlow2 notSel notChg notDrg	
<input checked="" type="checkbox"/> designFlow_drawing3	input (designFlow drawing3) df2order 193 packet designFlow3 notSel notChg notDrg	
<input checked="" type="checkbox"/> designFlow_drawing4	input (designFlow drawing4) df2order 194 packet designFlow4 notSel notChg notDrg	

 designFlow_drawing5	input (designFlow drawing5) df2order 195 packet designFlow5 notSel notChg notDrg	
 designFlow_drawing6	input (designFlow drawing6) df2order 196 packet designFlow6 notSel notChg notDrg	
 designFlow_drawing7	input (designFlow drawing7) df2order 197 packet designFlow7 notSel notChg notDrg	
 designFlow_drawing8	input (designFlow drawing8) df2order 198 packet designFlow8 notSel notChg notDrg	
 designFlow_drawing9	input (designFlow drawing9) df2order 199 packet designFlow9 notSel notChg notDrg	
 device	input df2order 145 packet device notChg notValid	
 device_annotate	input (device annotate) df2order 149 packet deviceAnt notChg notValid	
 device_drawing1	input (device drawing1) df2order 147 packet device1 notChg notValid	
 device_drawing2	input (device drawing2) df2order 148 packet device2 notChg notValid	
 device_label	input (device label) df2order 146 packet deviceLbl notChg notValid	
 drive	input df2order 183 packet drive notChg notValid	
 edgeLayer	input df2order 157 packet edgeLayer notChg notValid	
 edgeLayer_pin	input (edgeLayer pin) df2order 158 packet edgeLayerPin notChg notValid	
 grid	input df2order 120 packet grid notSel notChg notDrg notValid	
 grid_drawing1	input (grid drawing1) df2order 121 packet grid1 notSel notChg notDrg notValid	
 hardFence	input df2order 138 packet hardFence notChg notValid	
 hilite	input df2order 172 packet hilite notChg notValid	
 hilite_drawing1	input (hilite drawing1) df2order 173 packet hilite1 notChg notValid	
 hilite_drawing2	input (hilite drawing2) df2order 174 packet hilite2 notChg notValid	
 hilite_drawing3	input (hilite drawing3) df2order 175 packet hilite3 notChg notValid	
 hilite_drawing4	input (hilite drawing4) df2order 176 packet hilite4 notChg notValid	
 hilite_drawing5	input (hilite drawing5) df2order 177 packet hilite5 notChg notValid	
 hilite_drawing6	input (hilite drawing6) df2order 178 packet hilite6 notChg notValid	
 hilite_drawing7	input (hilite drawing7) df2order 179 packet hilite7 notChg notValid	
 hilite_drawing8	input (hilite drawing8) df2order 180 packet hilite8 notChg notValid	
 hilite_drawing9	input (hilite drawing9) df2order 181 packet hilite9 notChg notValid	
 hiz	input df2order 184 packet hiz notChg notValid	
 instance	input df2order 132 packet instance notChg notValid	
 instance_label	input (instance label) df2order 133 packet instanceLbl notChg notValid	
 marker_error	input (marker error) df2order 203 packet markerErr notChg notValid	
 marker_warning	input (marker warning) df2order 202 packet markerWarn notChg notValid	



 pin	input df2order 154 packet pin notChg notValid	
 pin_annotate	input (pin annotate) df2order 155 packet pinAnt notChg notValid	
 pin_label	input (pin label) df2order 153 packet pinLbl notChg notValid	
 prBoundary	input 235;0 df2order 134 packet prBoundary notChg	
 prBoundary_boundary	input 235;5 (prBoundary boundary) df2order 135 packet prBoundaryBnd notChg notValid	
 prBoundary_label	input (prBoundary label) df2order 136 packet prBoundaryLbl notChg notValid	
 resist	input df2order 185 packet resist notChg notValid	
 scaNburied_net	input (scaNburied net) df2order 111 notSel notChg notDrg notValid noTranslate	
 scaNwell_net	input (scaNwell net) df2order 112 notSel notChg notDrg notValid noTranslate	
 scaPort_net	input (scaPort net) df2order 113 notSel notChg notDrg notValid noTranslate	
 scaSelect	input df2order 110 notChg noTranslate	
 select	input df2order 182 packet select notChg notValid	
 snap	input df2order 159 packet snap notChg notValid	
 snap_boundary	input (snap boundary) df2order 160 packet snap notChg notValid	
 softFence	input df2order 139 packet softFence notChg notValid	
 spike	input df2order 186 packet spike notChg notValid	
 stretch	input df2order 161 packet stretch notChg notValid	
 supply	input df2order 187 packet supply notChg notValid	
 text	input 230;0 df2order 140 packet notChg	
 text_drawing1	input (text drawing1) df2order 142 packet text1 notChg notValid	
 text_drawing2	input (text drawing2) df2order 143 packet text2 notChg notValid	
 text_label	input 230;3 (text label) df2order 141 packet notChg	
 unknown	input df2order 188 packet unknown notChg notValid	
 unset	input df2order 189 packet unset notChg notValid	
 wire	input df2order 150 packet wire notChg notValid	
 wire_flight	input (wire flight) df2order 152 packet wireFlt notChg notValid	
 wire_label	input (wire label) df2order 151 packet wireLbl notChg notValid	
 y0	input df2order 162 packet y0 notChg notValid	
 y1	input df2order 163 packet y1 notChg notValid	
 y2	input df2order 164 packet y2 notChg notValid	
 y3	input df2order 165 packet y3 notChg notValid	
 y4	input df2order 166 packet y4 notChg notValid	
 y5	input df2order 167 packet y5 notChg notValid	
 y6	input df2order 168 packet y6 notChg notValid	
 y7	input df2order 169 packet y7 notChg notValid	
 y8	input df2order 170 packet y8 notChg notValid	
 y9	input df2order 171 packet y9 notChg notValid	

0A	Nburied width must be $\geq 1.0$ $\mu\text{m}$
0B	Nburied to Nburied spacing must be $\geq 1.0$ $\mu\text{m}$
1A	Nwell width must be $\geq 1.0$ $\mu\text{m}$
1B	Nwell to Nwell spacing must be $\geq 1.0$ $\mu\text{m}$
1C	Nburied to Nwell enclosure must be $\geq 0.3$ $\mu\text{m}$
1D	Pwell width must be $\geq 1.0$ $\mu\text{m}$
1E	Pwell to Pwell spacing must be $\geq 1.0$ $\mu\text{m}$
1F	Nburied to Pwell enclosure must be $\geq 0.3$ $\mu\text{m}$
2.5A	ThickOxide width must be $\geq 0.5$ $\mu\text{m}$
2.5B	ThickOxide to ThickOxide spacing must be $\geq 0.4$ $\mu\text{m}$
2.5C	ThickOxide to Oxide enclosure must be $\geq 0.25$ $\mu\text{m}$
2.5D	ThickOxide to Oxide spacing must be $\geq 0.25$ $\mu\text{m}$
2.5E	ThickOxide to Poly spacing must be $\geq 0.4$ $\mu\text{m}$
2.5F	ThickOxide to Poly enclosure must be $\geq 0.4$ $\mu\text{m}$
2A	Oxide width must be $\geq 0.4$ $\mu\text{m}$
2B	Oxide to Oxide spacing must be $\geq 0.3$ $\mu\text{m}$
2C	Nwell to Oxide enclosure must be $\geq 0.5$ $\mu\text{m}$
2C	Pwell to Oxide enclosure must be $\geq 0.5$ $\mu\text{m}$
2D	Nwell to Oxide spacing must be $\geq 0.5$ $\mu\text{m}$
2D	Pwell to Oxide spacing must be $\geq 0.5$ $\mu\text{m}$
3A	Nimp width must be $\geq 0.4$ $\mu\text{m}$
3B	Nimp to Nimp spacing must be $\geq 0.4$ $\mu\text{m}$
3C	Nimp to Oxide enclosure must be $\geq 0.2$ $\mu\text{m}$
3D	Nburied to Nimp enclosure must be $\geq 0.6$ $\mu\text{m}$
4A	Pimp width must be $\geq 0.4$ $\mu\text{m}$
4B	Pimp to Pimp spacing must be $\geq 0.4$ $\mu\text{m}$
4C	Pimp to Oxide enclosure must be $\geq 0.2$ $\mu\text{m}$
4D	Nburied to Pimp enclosure must be $\geq 0.6$ $\mu\text{m}$
5A	Poly width must be $\geq 0.18$ $\mu\text{m}$
5B	Poly to Poly spacing must be $\geq 0.3$ $\mu\text{m}$
5C	Poly to Oxide enclosure must be $\geq 0.2$ $\mu\text{m}$
5D	Oxide to Poly enclosure must be $\geq 0.4$ $\mu\text{m}$
5E	Oxide to Poly spacing must be $\geq 0.2$ $\mu\text{m}$
6A	Cont shapes must be $0.2 \times 0.2$ rectangles
6B	Cont to Cont spacing must be $\geq 0.2$ $\mu\text{m}$
6C	Oxide to Cont enclosure must be $\geq 0.2$ $\mu\text{m}$
6D	Poly to Cont enclosure must be $\geq 0.2$ $\mu\text{m}$
6E	Poly to Cont spacing must be $\geq 0.2$ $\mu\text{m}$
6F	Pimp to Cont enclosure must be $\geq 0.1$ $\mu\text{m}$
6G	Nimp to Cont enclosure must be $\geq 0.1$ $\mu\text{m}$
6H	Oxide to Cont spacing must be $\geq 0.2$ $\mu\text{m}$
7A	Metal1 width must be $\geq 0.3$ $\mu\text{m}$
7B	Metal1 to Metal1 spacing must be $\geq 0.3$ $\mu\text{m}$
7C	Metal1 to Cont enclosure must be $\geq 0.1$ $\mu\text{m}$
8A	Via1 shapes must be $0.2 \times 0.2$ rectangles
8B	Via1 to Via1 spacing must be $\geq 0.3$ $\mu\text{m}$
8C	Metal1 to Via1 enclosure must be $\geq 0.1$ $\mu\text{m}$
9A	Metal2 width must be $\geq 0.3$ $\mu\text{m}$



9B	Metal2 to Metal2 spacing must be $\geq 0.3$ $\mu\text{m}$
9C	Metal2 to Via1 enclosure must be $\geq 0.1$ $\mu\text{m}$
10A	Via2 shapes must be $0.2 \times 0.2$ rectangles
10B	Via2 to Via2 spacing must be $\geq 0.3$ $\mu\text{m}$
10C	Metal2 to Via2 enclosure must be $\geq 0.1$ $\mu\text{m}$
11A	Metal3 width must be $\geq 0.3$ $\mu\text{m}$
11B	Metal3 to Metal3 spacing must be $\geq 0.3$ $\mu\text{m}$
11C	Metal3 to Via2NoCapInd enclosure must be $\geq 0.1$ $\mu\text{m}$
11D	Metal3 to Via2Cap enclosure must be $\geq 0.1$ $\mu\text{m}$
12A	CapMetal width must be $\geq 0.5$ $\mu\text{m}$
12B	Metal2 to Via2Cap enclosure must be $\geq 0.1$ $\mu\text{m}$
12C	CapMetal to Via2Cap enclosure must be $\geq 0.2$ $\mu\text{m}$
12D	CapMetal to Metal3 enclosure must be $\geq 0.3$ $\mu\text{m}$
13A1	P+SD to NW tap spacing must be $\leq 10.0$ $\mu\text{m}$
13A2	N+SD to Iso Psub tap spacing must be $\leq 10.0$ $\mu\text{m}$
13A3	N+SD to Psub tap spacing must be $\leq 10.0$ $\mu\text{m}$
13A3	N+SD to PW tap spacing must be $\leq 10.0$ $\mu\text{m}$
14A	Via3 shapes must be $0.2 \times 0.2$ rectangles
14B	Via3 to Via3 spacing must be $\geq 0.3$ $\mu\text{m}$
14C	Metal3 to Via3 enclosure must be $\geq 0.1$ $\mu\text{m}$
15A	Metal4 width must be $\geq 0.3$ $\mu\text{m}$
15B	Metal4 to Metal4 spacing must be $\geq 0.3$ $\mu\text{m}$
15C	Metal4 to Via3 enclosure must be $\geq 0.1$ $\mu\text{m}$
16A	Via4 shapes must be $0.2 \times 0.2$ rectangles
16B	Via4 to Via4 spacing must be $\geq 0.3$ $\mu\text{m}$
16C	Metal4 to Via4 enclosure must be $\geq 0.1$ $\mu\text{m}$
17A	Metal5 width must be $\geq 0.3$ $\mu\text{m}$
17B	Metal5 to Metal5 spacing must be $\geq 0.3$ $\mu\text{m}$
17C	Metal5 to Via4 enclosure must be $\geq 0.1$ $\mu\text{m}$
18A	Via5 shapes must be $0.2 \times 0.2$ rectangles
18B	Via5 to Via5 spacing must be $\geq 0.3$ $\mu\text{m}$
18C	Metal5 to Via5 enclosure must be $\geq 0.1$ $\mu\text{m}$
19A	Metal6 width must be $\geq 0.3$ $\mu\text{m}$
19B	Metal6 to Metal6 spacing must be $\geq 0.3$ $\mu\text{m}$
19C	Metal6 to Via5 enclosure must be $\geq 0.1$ $\mu\text{m}$
20A	Minimum Pad width = 45.0
20B	Minimum Pad spacing to Pad = 10.0
20C	Metal1 to Bondpad enclosure must be $\geq 3.0$ $\mu\text{m}$
20D	Metal2 to Bondpad enclosure must be $\geq 3.0$ $\mu\text{m}$
20E	Metal3 to Bondpad enclosure must be $\geq 3.0$ $\mu\text{m}$
20F	Metal4 to Bondpad enclosure must be $\geq 3.0$ $\mu\text{m}$
20G	Metal5 to Bondpad enclosure must be $\geq 3.0$ $\mu\text{m}$
20H	Metal6 to Bondpad enclosure must be $\geq 3.0$ $\mu\text{m}$