

Listing 1: SKILL code for ind\_single

```

1  /* Intellectual Properties of RFVLSI LAB, NCTU, TAIWAN.
   Intended for Internal Use Only, All Rights Reserved, 2012
3  *DO NOT DISCLOSE*
   Author : Tao-Yi Lee */
5
6  strCellName="ind_single";
7
8  pcDefinePCell(
9  list(ddGetObj(strLibName),strCellName, "layout"),
10 (
11     (OD "float" 50.0)
12     (W "float" 2.0)
13     (OPENING "float" 5.0)
14     (LEAD "float" 10.0)
15     (S "float" 2.0)
16     (strName "string" "ind_single")
17     (NT_N "boolean" nil)
18 )
19 let((P DUMMYL m_lead_pair m_vias m_base.ind_hud_cross m_oct m_base.oct_fill i),
20     P=W+S /* Line pitch: width + space */
21     DUMMYL = "RFVLSI"
22     dbCreateLabel(pcCellView, list(DUMMYL, "dummy5"), 0:2, sprintf(nil "%L" W), "centerCenter", "R0",
23         "roman", 1)
24     dbCreateLabel(pcCellView, list(DUMMYL, "dummy6"), 0:4, sprintf(nil "%L" S), "centerCenter", "R0",
25         "roman", 1)
26     dbCreateLabel(pcCellView, list(DUMMYL, "dummy7"), 0:6, sprintf(nil "%L" OD), "centerCenter", "R0",
27         "roman", 1)
28
29     m_lead_pair = dbOpenCellViewByType(pcCellView~>lib "base_lead_pair" "layout" )
30     m_vias = dbOpenCellViewByType(pcCellView~>lib "vias" "layout" )
31     m_oct = dbOpenCellViewByType(pcCellView~>lib "base_oct" "layout" );
32     m_base.oct_fill = dbOpenCellViewByType(pcCellView~>lib "base_oct_fill" "layout" )
33
34     dbCreateParamInst(pcCellView m_base.oct_fill nil 0:0 "R0" 1 list(
35         list("OD" "float" OD)
36         list("layer" "string" "RFVLSI")
37         list("purp" "string" "drawing")))
38     dbCreateParamInst(pcCellView m_base.oct_fill nil 0:0 "R0" 1 list(
39         list("OD" "float" OD)
40         list("layer" "string" "RFVLSILVS")
41         list("purp" "string" "dummy8")))
42     dbCreateParamInst(pcCellView m_base.oct_fill nil 0:0 "R0" 1 list(
43         list("OD" "float" OD)
44         list("layer" "string" "RFVLSILVS")
45         list("purp" "string" "dummy9")))
46
47     dbCreateLabel(pcCellView, list(DUMMYL, "drawing"), 0:0, strName, "centerCenter", "R0", "roman",
48         2)
49
50     if (!NT_N
51     then
52         dbCreateLabel(pcCellView, list("text", "drawing"), 0:0, "Add_NT_N_Manually!", "
53             centerCenter", "R0", "roman", 3)
54     ) /* end of if (!NT_N */
55
56     /* #NOT# if (evenp(NT) */
57     dbCreateParamInst(pcCellView m_oct
58         nil 0:0 "R0" 1
59         list(
60             list("OD" "float" OD)
61             list("W" "float" W)
62             list("LOP" "float" 0.0)
63             list("ROP" "float" OPENING)
64             list("MET" "int" 9)
65         ) /* end of parameter list */
66     ) /* end of dbCreateParamInst */
67
68     /**** LEAD UNDER PASS ****/
69     dbCreateParamInst(pcCellView m_lead_pair nil OD/2:0 "R0" 1
70         list(
71             list("LEAD" "float" LEAD)
72             list("W" "float" W)
73             list("OPENING" "float" OPENING)
74             list("P1TXT" "string" "P1")
75             list("P2TXT" "string" "N1")
76             list("TOP_ME" "string" "9")
77             list("BTML_ME" "string" "8")
78         ) /* end of parameter list */
79     ) /* end of dbCreateParamInst */

```

```

77 /* EM Guard-ring */
78 rfvlslsiEMDummyCreateParamInst(pcCellView m_vias nil -OD/2-LEAD:-OD/2-10.0 "R0" 1
79 list(
80 list("Width" "float" 5.0)
81 list("Length" "float" OD+20.0)
82 list("TOP_ME" "int" 1)
83 list("BTMME" "int" 1)
84 ) 1) /* end of EM Guard-ring */
85
86 /* EM Guard-ring */
87 rfvlslsiEMDummyCreateParamInst(pcCellView m_vias nil OD/2+LEAD:-OD/2-10.0 "MY" 1
88 list(
89 list("Width" "float" 5.0)
90 list("Length" "float" OD+20.0)
91 list("TOP_ME" "int" 1)
92 list("BTMME" "int" 1)
93 ) 1) /* end of EM Guard-ring */
94
95 /* EM Guard-ring */
96 rfvlslsiEMDummyCreateParamInst(pcCellView m_vias nil -OD/2-LEAD:OD/2+5.0 "R0" 1
97 list(
98 list("Width" "float" OD+2*LEAD)
99 list("Length" "float" 5.0)
100 list("TOP_ME" "int" 1)
101 list("BTMME" "int" 1)
102 ) 1) /* end of EM Guard-ring */
103
104 /* EM Guard-ring */
105 rfvlslsiEMDummyCreateParamInst(pcCellView m_vias nil -OD/2-LEAD:-OD/2-5.0 "MX" 1
106 list(
107 list("Width" "float" OD+2*LEAD)
108 list("Length" "float" 5.0)
109 list("TOP_ME" "int" 1)
110 list("BTMME" "int" 1)
111 ) ,1) /* end of EM Guard-ring */
112
113 rfvlslsiEMVport("P1",OD/2+LEAD, OPENING,OD/2+LEAD,OPENING+W,0,8,1);
114 rfvlslsiEMVport("N1",OD/2+LEAD, -OPENING,OD/2+LEAD,-OPENING-W,0,8,1);
115 boundary_ext = 15
116 rfvlslsiEMBoundary(-OD - boundary_ext,-OD-boundary_ext,-300.0, 2*(OD+boundary_ext),2*(OD+
117 boundary_ext),600.0,1)
118 rfvlslsiEMDie(-OD - boundary_ext*.9,-OD-boundary_ext*0.9, 2*(OD+boundary_ext*0.9), 2*(OD+
119 boundary_ext*0.9),1)
120 )) /* end of */

```

Listing 2: SKILL code for inductor parameterized cell

```

/* Intellectual Properties of RFVLSI LAB, NCTU, TAIWAN.
2 Intended for Internal Use Only, All Rights Reserved, 2012
3 *DO NOT DISCLOSE*
4 Author : Tao-Yi Lee */
5
6 strCellName="ind-sym";
7
8 pcDefinePCell(
9 list(ddGetObj(strLibName),strCellName, "layout"),
10 (
11 (OD "float" 50.0)
12 (W "float" 2.0)
13 (OPENING "float" 5.0)
14 (LEAD "float" 10.0)
15 (S "float" 2.0)
16 (NT "int" 2)
17 (strName "string" "ind-sym")
18 (NT_N "boolean" nil)
19 (dummy "boolean" t)
20 )
21 let((P DUMMYL m_lead_pair m_vias m_base_ind_hud_cross m_oct m_base_oct_fill i m_base_em_gr),
22 P=W+S /* Line pitch: width + space */
23 DUMMYL = "RFVLSI"
24
25 m_lead_pair = dbOpenCellViewByType(pcCellView~>lib "base_lead_pair" "layout" )
26 m_base_em_gr = dbOpenCellViewByType(pcCellView~>lib "base_em_gr" "layout" )
27 m_vias = dbOpenCellViewByType(pcCellView~>lib "vias" "layout" )
28 m_base_ind_hud_cross = dbOpenCellViewByType(pcCellView~>lib "base_ind_hud_cross" "layout" )
29 m_oct = dbOpenCellViewByType(pcCellView~>lib "base_oct" "layout" );
30 m_base_oct_fill = dbOpenCellViewByType(pcCellView~>lib "base_oct_fill" "layout" )
31
32 dbCreateParamInst(pcCellView m_base_oct_fill nil 0:0 "R0" 1 list(
33 list("OD" "float" OD)
34 list("layer" "string" "RFVLSI")

```

```

36         list("purp" "string" "drawing"))))
dbCreateParamInst(pcCellView m_base_oct_fill nil 0:0 "R0" 1 list(
38     list("OD" "float" OD)
    list("layer" "string" "RFVLSILVS")
40     list("purp" "string" "dummy8"))))
dbCreateParamInst(pcCellView m_base_oct_fill nil 0:0 "R0" 1 list(
42     list("OD" "float" OD)
    list("layer" "string" "RFVLSILVS")
44     list("purp" "string" "dummy9"))))
if (dummy
46 then
    dbCreateLabel(pcCellView, list(DUMMYL "drawing"), OD/2:0, strName, "centerCenter", "R0",
        "roman", 3)
48     dbCreateLabel(pcCellView, list(DUMMYL "dummy5"), 0:-1, sprintf(nil "%L" W), "
        centerCenter", "R0", "roman", 1)
    dbCreateLabel(pcCellView, list(DUMMYL "dummy6"), 0:-3, sprintf(nil "%L" S), "
        centerCenter", "R0", "roman", 1)
50     dbCreateLabel(pcCellView, list(DUMMYL "dummy7"), 0:1, sprintf(nil "%L" OD), "
        centerCenter", "R0", "roman", 1)
    dbCreateLabel(pcCellView, list(DUMMYL "dummy8"), 0:5, sprintf(nil "%L" NT), "
        centerCenter", "R0", "roman", 1)
52 )
if (!NT_N
54 then
    dbCreateLabel(pcCellView, list("text", "drawing"), 0:0, "Add_NT_N_Manually!", "
        centerCenter", "R0", "roman", 3)
56 ) /* end of if(!NT_N */
58
if (NT>2
60 then
/* Inner L W */
62     for (i 1 NT-2
        if (evenp(i)
64         then
            dbCreateParamInst(pcCellView m_base_ind_hud_cross nil 0:0 "R0" 1 list(
66                list("OD" "float" OD-2*i*P)
                list("W" "float" W)
68                list("S" "float" S)
                list("OPENING" "float" 2*W)
70                list("TOP_ME" "string" "9")
                list("BTM_ME" "string" "8")
72                list("DUMMYL" "string" "RFVLSI")
            )) /* end of dbCreateParamInst */
74         else
76             dbCreateParamInst(pcCellView m_base_ind_hud_cross nil 0:0 "MY" 1
                list(
78                    list("OD" "float" OD-2*i*P)
                    list("W" "float" W)
80                    list("S" "float" S)
                    list("OPENING" "float" 2*W)
82                    list("TOP_ME" "string" "9")
                    list("BTM_ME" "string" "8")
84                    list("DUMMYL" "string" "RFVLSI")
                ) /* end of parameter list */
            ) /* end of dbCreateParamInst */
86
88         ) /* end of if (evenp(i) */
        ) /* end of for (i 1 NT-2 */
90 ) /* end of if (NT>2 */
92
    dbCreateParamInst(pcCellView m_base_ind_hud_cross nil 0:0 "R0" 1
        list(
94            list("OD" "float" OD)
            list("W" "float" W) ; Line width
96            list("S" "float" S)
            list("OPENING" "float" OPENING)
98            list("TOP_ME" "string" "9")
            list("BTM_ME" "string" "8")
100            list("DUMMYL" "string" "RFVLSI")
        ) /* end of parameter list */
102 ) /* end of dbCreateParamInst */
104
if (evenp(NT)
106 then
    dbCreateParamInst(pcCellView m_base_ind_hud_cross nil 0:0 "R0" 1
        list(
108            list("OD" "float" OD-2*(NT-1)*P)
            list("W" "float" W)
            list("S" "float" S)
110            list("OPENING" "float" 0)

```

```

112         list("TOP_ME" "string" "9")
113         list("BTMLME" "string" "8")
114         list("DUMMYL" "string" "RFVLSI")
115         list("under" "boolean" nil)
116     ) /* end of parameter list */
117 ) /* end of dbCreateParamInst */
118 else
119     /* #NOT# if(evenp(NT) */
120     dbCreateParamInst(pcCellView m_oct
121         nil 0:0 "R0" 1
122         list(
123             list("OD" "float" OD-2*(NT-1)*P)
124             list("W" "float" W)
125             list("LOP" "float" 0.0)
126             list("ROP" "float" 2*W)
127             list("MET" "int" 9)
128         ) /* end of parameter list */
129     ) /* end of dbCreateParamInst */
130 ) /* end of if(evenp(NT) */
131
132 /**** LEAD UNDER PASS ****/
133 dbCreateParamInst(pcCellView m_lead_pair nil OD/2:0 "R0" 1
134     list(
135         list("LEAD" "float" LEAD)
136         list("W" "float" W)
137         list("OPENING" "float" OPENING)
138         list("P1TXT" "string" "P1")
139         list("P2TXT" "string" "N1")
140         list("TOP_ME" "string" "9")
141         list("BTMLME" "string" "8")
142         list("dummy" "boolean" dummy)
143     ) /* end of parameter list */
144 ) /* end of dbCreateParamInst */
145
146 dbCreateParamInst(pcCellView m_base_em_gr nil 0:0 "R0" 1
147 list(
148     list("W" "float" (OD+LEAD))
149     list("L" "float" (OD+2*LEAD-4))
150     list("LW" "float" 2.0)
151     list("boundary_ext" "float" 10.0)
152     list("priority" "int" 1)
153 ))
154
155 rfvlslEMVport("P1",OD/2+LEAD, OPENING,OD/2+LEAD,OPENING+W,0,8,1);
156 rfvlslEMVport("N1",OD/2+LEAD, -OPENING,OD/2+LEAD,-OPENING-W,0,8,1);
157 )) /* end of */

```

Listing 3: SKILL code for inductor parameterized cell (CDF)

```

1 ;;; CDF GENERATION OF tg_inv
2
3 strCellName="ind_sym"; The Cell Name is called inductor
4
5 cellId = ddGetObj(strLibName strCellName)
6 if(g_cdfDataId = cdfGetBaseCellCDF(cellId)
7 then
8     ret = cdfDeleteCDF(g_cdfDataId)
9     fprintf(stderr "***_CDF_ of_%L_ deleted_%L_\n" strCellName ret)
10 )
11
12 g_cdfDataId = cdfCreateBaseCellCDF(cellId ?fieldWidth 580 ?fieldHeight 35 ?buttonFieldWidth 340 ?
    promptWidth 300)
13 cdfCreateParam(g_cdfDataId ?name "OD" ?type "float" ?defValue 50.0 ?prompt "Outer_Diameter" ?display
    "t" ?editable "t" ?storeDefault nil ?callback "")
14 cdfCreateParam(g_cdfDataId ?name "LEAD" ?type "float" ?defValue 10.0 ?prompt "Lead_Length" ?display "t
    " ?editable "t" ?storeDefault nil ?callback "")
15 cdfCreateParam(g_cdfDataId ?name "OPENING" ?type "float" ?defValue 5.0 ?prompt "Lead_Opening" ?display "
    t" ?editable "t" ?storeDefault nil ?callback "")
16 cdfCreateParam(g_cdfDataId ?name "W" ?type "float" ?defValue 2.0 ?prompt "Line_Width" ?display "t"
    ?editable "t" ?storeDefault nil ?callback "")
17 cdfCreateParam(g_cdfDataId ?name "S" ?type "float" ?defValue 2.0 ?prompt "Spacing" ?display "t" ?
    editable "t" ?storeDefault nil ?callback "")
18
19 cdfCreateParam(g_cdfDataId ?name "NT_STR" ?type "cyclic" ?defValue "2" ?choices list("2" "4" "6" "8" "10"
    ) ?prompt "Turns" ?display "t" ?editable "t"
    ?storeDefault nil ?callback "cdfgData->NT->value==atoi(cdfgData->NT_STR->value)" )
20
21 cdfCreateParam(g_cdfDataId ?name "NT" ?type "int" ?defValue 2 ?prompt "Turns(int)" ?display "nil" ?
    editable "nil" ?storeDefault nil ?callback "")
22
23 cdfCreateParam(g_cdfDataId ?name "NT_N" ?type "boolean" ?defValue nil ?prompt "Native_N-TYPE_doping" ?
    display "t" ?editable "t" ?storeDefault nil ?parseAsCEL "no" ?callback "")
24 cdfCreateParam(g_cdfDataId ?name "strName" ?type "string" ?defValue "ind_sym" ?prompt "instance_name" ?

```

```

25     display "t" ?editable "nil"
        ?storeDefault nil ?parseAsCEL "no" ?callback "")
cdfCreateParam(g_cdfDataId ?name "L1" ?type "string" ?defValue "488p" ?prompt "Main_Inductance" ?display
"t" ?editable "t"
27     ?storeDefault nil ?parseAsCEL "yes" ?units "inductance" ?parseAsNumber "yes" ?callback "")
cdfCreateParam(g_cdfDataId ?name "R1" ?type "string" ?defValue "2.1305" ?prompt "R1" ?display "t" ?
editable "t"
29     ?storeDefault nil ?parseAsCEL "yes" ?units "resistance" ?parseAsNumber "yes" ?callback "")
31 cdfCreateParam(g_cdfDataId ?name "L2" ?type "string" ?defValue "1.4836n" ?prompt "Secondary_Inductance" ?
display "t" ?editable "t"
        ?storeDefault nil ?parseAsCEL "yes" ?units "inductance" ?parseAsNumber "yes" ?callback "")
33 cdfCreateParam(g_cdfDataId ?name "R2" ?type "string" ?defValue "11.904" ?prompt "R2" ?display "t" ?
editable "t"
        ?storeDefault nil ?parseAsCEL "yes" ?units "resistance" ?parseAsNumber "yes" ?callback "")
35 cdfCreateParam(g_cdfDataId ?name "C12" ?type "string" ?defValue "14.67f" ?prompt "C12" ?display "t" ?
editable "t"
        ?storeDefault nil ?parseAsCEL "yes" ?units "capacitance" ?parseAsNumber "yes" ?callback "")
37 cdfCreateParam(g_cdfDataId ?name "CS1" ?type "string" ?defValue "17.67f" ?prompt "CS1" ?display "t" ?
editable "t"
        ?storeDefault nil ?parseAsCEL "yes" ?units "capacitance" ?parseAsNumber "yes" ?callback "")
39 cdfCreateParam(g_cdfDataId ?name "CS3" ?type "string" ?defValue "17.824f" ?prompt "CS3" ?display "t" ?
editable "t"
        ?storeDefault nil ?parseAsCEL "yes" ?units "capacitance" ?parseAsNumber "yes" ?callback "")
41 cdfCreateParam(g_cdfDataId ?name "RS5" ?type "string" ?defValue "3.427k" ?prompt "RS5" ?display "t" ?
editable "t"
        ?storeDefault nil ?parseAsCEL "yes" ?units "resistance" ?parseAsNumber "yes" ?callback "")
43 cdfCreateParam(g_cdfDataId ?name "CS5" ?type "string" ?defValue "2.32f" ?prompt "CS5" ?display "t" ?
editable "t"
        ?storeDefault nil ?parseAsCEL "yes" ?units "capacitance" ?parseAsNumber "yes" ?callback "")
45 cdfCreateParam(g_cdfDataId ?name "RS7" ?type "string" ?defValue "3.36k" ?prompt "RS7" ?display "t" ?
editable "t"
        ?storeDefault nil ?parseAsCEL "yes" ?units "resistance" ?parseAsNumber "yes" ?callback "")
47 cdfCreateParam(g_cdfDataId ?name "CS7" ?type "string" ?defValue "2.5f" ?prompt "CS7" ?display "t" ?
editable "t"
        ?storeDefault nil ?parseAsCEL "yes" ?units "capacitance" ?parseAsNumber "yes" ?callback "")
51
53 g_cdfDataId->simInfo = list( nil )
55 g_cdfDataId->simInfo->auCdl = '( nil
        dollarEqualParams nil
57         dollarParams      nil
        modelName           "ind_sym"
59         namePrefix        "L"
        propMapping         nil
61         termOrder         (plus minus vss)
        componentName       ind_sym
63         instParameters    (OD NT W S L1)
        otherParameters    nil
65         netlistProcedure  ansCdlSubcktCall
    )
67
69 g_cdfDataId->paramLabelSet = "OD_W_S_NT_L1_R1_L2_R2_C12_CS1_CS5_RS5_CS3_CS7_RS7"
cdfSaveCDF( g_cdfDataId )
71
;libpath = dbFullLibPath(strLibName)
;cdfDump(strLibName strcat(libpath "/" strCellName ".cdf") ?cellName strCellName ?level 'base ?edit nil)

```

Listing 4: SKILL code for base\_ind\_hud\_cross

```

/*      Intellectual Properties of RFVLSI LAB, NCTU, TAIWAN.
2      Intended for Internal Use Only, All Rights Reserved, 2012
        *DO NOT DISCLOSE*
4      Author : Tao-Yi Lee */

6 strCellName="base_ind_hud_cross";
pcDefinePCell(
8 list(ddGetObj(strLibName),strCellName, "layout"),
(
10     (OD      "float" 60.0)
    (W      "float" 2.0)
12     (S      "float" 2.0)
    (OPENING "float" 10.0)
14     (TOP_ME "string" "9")
    (BTM_ME  "string" "9")
16     (under  "boolean" t)
    (dummy   "boolean" t)
18     (DUMMYL "string" "RFVLSI")
)
)

```

```

20 let ((P via_to_next C C2 pi OO OOC OUCH m_base_oct),
22     P=W+S
23     if (atoi(TOP_ME) - atoi(BTM_ME) < 2
24     then
25         via_to_next = nil
26     )
27     pi = 3.141592
28     C=roundtograd(W*tan(pi/8))
29     C2=roundtograd(C/sqrt(2))
30     OO = 2*W+S
31     OOC = 2*W+S-2*C2
32     OUCH = OOC/2

34     m_base_oct = dbOpenCellViewByType(pcCellView~>lib "base_oct" "layout" );
35     dbCreateParamInst(pcCellView m_base_oct
36         nil 0:0 "R0" 1
37         list(
38             list("OD" "float" OD)
39             list("W" "float" W)
40             list("LOP" "float" OUCH + roundtograd(2*sqrt(2.0)- S)+0.01)
41             list("ROP" "float" OPENING)
42             list("MET" "int" 9)
43         )) ; close dbCreateParamInst

44 if (under
45 then
46     dbCreateParamInst(pcCellView
47         dbOpenCellViewByType(pcCellView~>lib "base_xfm_cross" "layout") nil (-OD/2):0 "R0"
48         "1"
49         list(
50             list("WI" "float" S)
51             list("WO" "float" W)
52             list("S" "float" 0.0)
53             list("TOP_ME" "int" atoi(TOP_ME))
54             list("BTM_ME" "int" atoi(TOP_ME)-1)
55             list("dummy" "boolean" dummy)
56         )) ; close dbCreateParamInst

57     dbCreateParamInst(pcCellView
58         dbOpenCellViewByType(pcCellView~>lib "base_xfm_cross" "layout") nil (-OD/2+P+W):0
59         "MY" 1
60         list(
61             list("WI" "float" S)
62             list("WO" "float" W)
63             list("S" "float" 0.0)
64             list("TOP_ME" "int" atoi(TOP_ME))
65             list("BTM_ME" "int" atoi(TOP_ME))
66             list("viat" "boolean" nil)
67             list("viad" "boolean" nil)
68             list("dummy" "boolean" dummy)
69         )) ; close dbCreateParamInst
70 ))

```