Listing 1: SKILL code for ind_single

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

strCellName=”ind_single”;

pcDefinePCell(
list(ddGetObj(strLibName),strCellName, ”layout”),
(
        (OD ”float” 50.0)
        (W ”float” 2.0)
        (OPENING ”float” 5.0)
        (LEAD ”float” 10.0)
        (S ”float” 2.0)
        (strName ”string” ”ind_single”)
        (NT_N ”boolean” nil)
)
let((P DUMMYL m_lead_pair m_vias m_base_ind_hud_cross m_oct m_base_oct_fill i),
        P=W+S /* Line pitch: width + space */
        DUMMYL = ”RFVLSI”
        dbCreateLabel(pcCellView, list(DUMMYL, ”dummy5”), 0:2, sprintf(nil ”%L” W), ”centerCenter”, ”R0”,
             ”roman”, 1)
        dbCreateLabel(pcCellView, list(DUMMYL, ”dummy6”), 0:4, sprintf(nil ”%L” S), ”centerCenter”, ”R0”,
             ”roman”, 1)
        dbCreateLabel(pcCellView, list(DUMMYL, ”dummy7”), 0:6, sprintf(nil ”%L” OD), ”centerCenter”, ”R0”
             , ”roman”, 1)

        m_lead_pair = dbOpenCellViewByType(pcCellView~>lib ”base_lead_pair” ”layout” )
        m_vias = dbOpenCellViewByType(pcCellView~>lib ”vias” ”layout” )
        m_oct = dbOpenCellViewByType(pcCellView~>lib ”base_oct” ”layout” );
        m_base_oct_fill = dbOpenCellViewByType(pcCellView~>lib ”base_oct_fill” ”layout” )

        dbCreateParamInst(pcCellView m_base_oct_fill nil 0:0 ”R0” 1 list(
            list(”OD”      ”float” OD)
            list(”layer” ”string”   ”RFVLSI”)
            list(”purp” ”string”  ”drawing”)))
        dbCreateParamInst(pcCellView m_base_oct_fill nil 0:0 ”R0” 1 list(
            list(”OD”      ”float” OD)
            list(”layer” ”string”   ”RFVLSI_LVS”)
            list(”purp” ”string”  ”dummy8”)))
        dbCreateParamInst(pcCellView m_base_oct_fill nil 0:0 ”R0” 1 list(
            list(”OD”      ”float” OD)
            list(”layer” ”string”   ”RFVLSI_LVS”)
            list(”purp” ”string”  ”dummy9”)))


        dbCreateLabel(pcCellView, list(DUMMYL, ”drawing”), 0:0, strName, ”centerCenter”, ”R0”, ”roman”,
            2)

        if(!NT_N
        then
                dbCreateLabel(pcCellView, list(”text”, ”drawing”), 0:0, ”Add_NT_N_Manually!”, ”
                    centerCenter”, ”R0”, ”roman”, 3)
        )       /* end of if(!NT_N */

                /* #NOT# if(evenp(NT) */
                dbCreateParamInst(pcCellView m_oct
                nil 0:0 ”R0” 1
                        list(
                        list(”OD” ”float” OD)
                        list(”W” ”float” W)
                        list(”LOP” ”float” 0.0)
                        list(”ROP” ”float” OPENING)
                        list(”MET” ”int” 9)
                        ) /* end of parameter list */
                ) /* end of dbCreateParamInst */

        /**** LEAD UNDER PASS ****/
        dbCreateParamInst(pcCellView m_lead_pair nil OD/2:0 ”R0” 1
                list(
                list(”LEAD” ”float” LEAD)
                list(”W” ”float” W)
                list(”OPENING” ”float” OPENING)
                list(”P1TXT” ”string” ”P1”)
                list(”P2TXT” ”string” ”N1”)
                list(”TOP_ME” ”string” ”9”)
                list(”BTM_ME” ”string” ”8”)
                ) /* end of parameter list */
        ) /* end of dbCreateParamInst */
        /* EM Guard-ring */
        rfvlsiEMDummyCreateParamInst(pcCellView m_vias nil -OD/2-LEAD:-OD/2-10.0 "R0" 1
        list(
        list("Width" "float" 5.0)
        list("Length" "float" OD+20.0)
        list("TOP_ME" "int" 1)
        list("BTM_ME" "int" 1)
        ) 1) /* end of EM Guard-ring */

        /* EM Guard-ring */
        rfvlsiEMDummyCreateParamInst(pcCellView m_vias nil OD/2+LEAD:-OD/2-10.0 "MY" 1
        list(
        list("Width" "float" 5.0)
        list("Length" "float" OD+20.0)
        list("TOP_ME" "int" 1)
        list("BTM_ME" "int" 1)
        ) 1) /* end of EM Guard-ring */

        /* EM Guard-ring */
        rfvlsiEMDummyCreateParamInst(pcCellView m_vias nil -OD/2-LEAD:OD/2+5.0 "R0" 1
        list(
        list("Width" "float" OD+2*LEAD)
        list("Length" "float" 5.0)
        list("TOP_ME" "int" 1)
        list("BTM_ME" "int" 1)
        ) 1) /* end of EM Guard-ring */

        /* EM Guard-ring */
        rfvlsiEMDummyCreateParamInst(pcCellView m_vias nil -OD/2-LEAD:-OD/2-5.0 "MX" 1
        list(
        list("Width" "float" OD+2*LEAD)
        list("Length" "float" 5.0)
        list("TOP_ME" "int" 1)
        list("BTM_ME" "int" 1)
        ) ,1) /* end of EM Guard-ring */

        rfvlsiEMVport("P1",OD/2+LEAD, OPENING,OD/2+LEAD,OPENING+W,0,8,1);
        rfvlsiEMVport("N1",OD/2+LEAD, -OPENING,OD/2+LEAD,-OPENING-W,0,8,1);
        boundary_ext = 15
        rfvlsiEMBoundary(-OD - boundary_ext,-OD-boundary_ext,-300.0, 2*(OD+boundary_ext),2*(OD+
            boundary_ext),600.0,1)
        rfvlsiEMDie(-OD - boundary_ext*.9,-OD-boundary_ext*0.9, 2*(OD+boundary_ext*0.9), 2*(OD+
            boundary_ext*0.9),1)
)) /* end of */
```

Listing 2: SKILL code for inductor parameterized cell

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

strCellName="ind_sym";

pcDefinePCell(
list(ddGetObj(strLibName),strCellName, "layout"),
(
        (OD "float" 50.0)
        (W "float" 2.0)
        (OPENING "float" 5.0)
        (LEAD "float" 10.0)
        (S "float" 2.0)
        (NT "int" 2)
        (strName "string" "ind_sym")
        (NT_N "boolean" nil)
        (dummy "boolean" t)
)
let((P DUMMYL m_lead_pair m_vias m_base_ind_hud_cross m_oct m_base_oct_fill i m_base_em_gr),
        P=W+S /* Line pitch: width + space */
        DUMMYL = "RFVLSI"

        m_lead_pair = dbOpenCellViewByType(pcCellView~>lib "base_lead_pair" "layout" )
        m_base_em_gr = dbOpenCellViewByType(pcCellView~>lib "base_em_gr" "layout" )
        m_vias = dbOpenCellViewByType(pcCellView~>lib "vias" "layout" )
        m_base_ind_hud_cross = dbOpenCellViewByType(pcCellView~>lib "base_ind_hud_cross" "layout" )
        m_oct = dbOpenCellViewByType(pcCellView~>lib "base_oct" "layout" );
        m_base_oct_fill = dbOpenCellViewByType(pcCellView~>lib "base_oct_fill" "layout" )


        dbCreateParamInst(pcCellView m_base_oct_fill nil 0:0 "R0" 1 list(
            list("OD"     "float" OD)
            list("layer" "string"   "RFVLSI")
            list("purp" "string"  "drawing")))
        dbCreateParamInst(pcCellView m_base_oct_fill nil 0:0 "R0" 1 list(
            list("OD"      "float" OD)
            list("layer" "string"   "RFVLSI_LVS")
            list("purp" "string"  "dummy8")))
        dbCreateParamInst(pcCellView m_base_oct_fill nil 0:0 "R0" 1 list(
            list("OD"      "float" OD)
            list("layer" "string"   "RFVLSI_LVS")
            list("purp" "string"  "dummy9")))
        if(dummy
        then
                dbCreateLabel(pcCellView, list(DUMMYL, "drawing"), OD/2:0, strName, "centerCenter", "R0",
                     "roman", 3)
                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)
                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)
        )

        if(!NT_N
        then
                dbCreateLabel(pcCellView, list("text", "drawing"), 0:0, "Add_NT_N_Manually!", "
                    centerCenter", "R0", "roman", 3)
        )       /* end of if(!NT_N */

        if(NT>2
        then
        /* Inner L W */
                for(i 1 NT-2
                        if(evenp(i)
                        then
                                dbCreateParamInst(pcCellView m_base_ind_hud_cross nil 0:0 "R0" 1 list(
                                list("OD" "float" OD-2*i*P)
                                list("W" "float" W)
                                list("S" "float" S)
                                list("OPENING" "float" 2*W)
                                list("TOP_ME" "string" "9")
                                list("BTM_ME" "string" "8")
                                list("DUMMYL" "string" "RFVLSI")
                        )) /* end of dbCreateParamInst */

                        else
                                dbCreateParamInst(pcCellView m_base_ind_hud_cross nil 0:0 "MY" 1
                                        list(
                                        list("OD" "float" OD-2*i*P)
                                        list("W" "float" W)
                                        list("S" "float" S)
                                        list("OPENING" "float" 2*W)
                                        list("TOP_ME" "string" "9")
                                        list("BTM_ME" "string" "8")
                                        list("DUMMYL" "string" "RFVLSI")
                                        ) /* end of parameter list */
                                ) /* end of dbCreateParamInst */

                        ) /* end of if(evenp(i) */
                ) /* end of for(i 1 NT-2 */
        ) /* end of if(NT>2 */

                dbCreateParamInst(pcCellView m_base_ind_hud_cross nil 0:0 "R0" 1
                        list(
                        list("OD" "float" OD)
                        list("W" "float" W) ; Line widh
                        list("S" "float" S)
                        list("OPENING" "float" OPENING)
                        list("TOP_ME" "string" "9")
                        list("BTM_ME" "string" "8")
                        list("DUMMYL" "string" "RFVLSI")
                        ) /* end of parameter list */
                ) /* end of dbCreateParamInst */

        if(evenp(NT)
        then
                dbCreateParamInst(pcCellView m_base_ind_hud_cross nil 0:0 "R0" 1
                        list(
                        list("OD" "float" OD-2*(NT-1)*P)
                        list("W" "float" W)
                        list("S" "float" S)
                        list("OPENING" "float" 0)
                        list("TOP_ME" "string" "9")
                        list("BTM_ME" "string" "8")
                        list("DUMMYL" "string" "RFVLSI")
                        list("under" "boolean" nil)
                        ) /* end of parameter list */
                ) /* end of dbCreateParamInst */
        else
                /* #NOT# if(evenp(NT) */
                dbCreateParamInst(pcCellView m_oct
                nil 0:0 "R0" 1
                        list(
                        list("OD" "float" OD-2*(NT-1)*P)
                        list("W" "float" W)
                        list("LOP" "float" 0.0)
                        list("ROP" "float" 2*W)
                        list("MET" "int" 9)
                        ) /* end of parameter list */
                ) /* end of dbCreateParamInst */
        ) /* end of if(evenp(NT) */

        /**** LEAD UNDER PASS ****/
        dbCreateParamInst(pcCellView m_lead_pair nil OD/2:0 "R0" 1
                list(
                list("LEAD" "float" LEAD)
                list("W" "float" W)
                list("OPENING" "float" OPENING)
                list("P1TXT" "string" "P1")
                list("P2TXT" "string" "N1")
                list("TOP_ME" "string" "9")
                list("BTM_ME" "string" "8")
                list("dummy" "boolean" dummy)
                ) /* end of parameter list */
        ) /* end of dbCreateParamInst */

        dbCreateParamInst(pcCellView m_base_em_gr nil 0:0 "R0" 1
    list(
        list("W" "float" (OD+LEAD))
        list("L" "float" (OD+2*LEAD-4))
        list("LW" "float" 2.0)
        list("boundary_ext" "float" 10.0)
        list("priority" "int" 1)
    ))

        rfvlsiEMVport("P1",OD/2+LEAD, OPENING,OD/2+LEAD,OPENING+W,0,8,1);
        rfvlsiEMVport("N1",OD/2+LEAD, -OPENING,OD/2+LEAD,-OPENING-W,0,8,1);
)) /* end of */
```

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

```
;;; CDF GENERATION OF tg_inv

strCellName="ind_sym";    The Cell Name is called inductor

cellId = ddGetObj(strLibName strCellName)
if(g_cdfDataId = cdfGetBaseCellCDF(cellId)
then
        ret = cdfDeleteCDF(g_cdfDataId)
        fprintf(stderr "***␣CDF␣of␣%L␣deleted␣(%L)\n" strCellName ret)
)

g_cdfDataId = cdfCreateBaseCellCDF(cellId ?fieldWidth 580 ?fieldHeight 35 ?buttonFieldWidth 340 ?
    promptWidth 300)
cdfCreateParam(g_cdfDataId ?name "OD" ?type "float" ?defValue 50.0      ?prompt "Outer␣Diameter" ?display
     "t" ?editable "t" ?storeDefault nil  ?callback "")
cdfCreateParam(g_cdfDataId ?name "LEAD" ?type "float" ?defValue 10.0    ?prompt "Lead␣Length" ?display "t
    " ?editable "t" ?storeDefault nil  ?callback "")
cdfCreateParam(g_cdfDataId ?name "OPENING" ?type "float" ?defValue 5.0  ?prompt "Lead␣Opening" ?display "
    t" ?editable "t" ?storeDefault nil  ?callback "")
cdfCreateParam(g_cdfDataId ?name "W" ?type "float" ?defValue 2.0        ?prompt "Line␣Width" ?display "t"
     ?editable "t" ?storeDefault nil  ?callback "")
cdfCreateParam(g_cdfDataId ?name "S" ?type "float" ?defValue 2.0        ?prompt "Spacing" ?display "t" ?
    editable "t" ?storeDefault nil  ?callback "")

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)" )

cdfCreateParam(g_cdfDataId ?name "NT" ?type "int" ?defValue 2 ?prompt "Turns(int)" ?display "nil" ?
    editable "nil" ?storeDefault nil  ?callback "")
cdfCreateParam(g_cdfDataId ?name "NT_N" ?type "boolean" ?defValue nil ?prompt "Native␣N-TYPE␣doping" ?
    display "t" ?editable "t" ?storeDefault nil ?parseAsCEL "no" ?callback "")
cdfCreateParam(g_cdfDataId ?name "strName" ?type "string" ?defValue "ind_sym" ?prompt "instance␣name" ?
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"
        ?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"
        ?storeDefault nil ?parseAsCEL "yes" ?units "resistance" ?parseAsNumber "yes" ?callback "")

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 "")
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 "")

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 "")
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 "")
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 "")

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 "")
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 "")

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 "")
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 "")


g_cdfDataId->simInfo = list( nil )
g_cdfDataId->simInfo->auCdl = '( nil
        dollarEqualParams nil
        dollarParams      nil
        modelName         "ind_sym"
        namePrefix        "L"
        propMapping       nil
        termOrder         (plus minus vss)
        componentName     ind_sym
        instParameters    (OD NT W S L1)
        otherParameters   nil
        netlistProcedure  ansCdlSubcktCall
    )

 g_cdfDataId->paramLabelSet = "OD␣W␣S␣NT␣L1␣R1␣L2␣R2␣C12␣CS1␣CS5␣RS5␣CS3␣CS7␣RS7"
cdfSaveCDF( g_cdfDataId )

;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.
        Intended for Internal Use Only, All Rights Reserved, 2012
        *DO NOT DISCLOSE*
        Author : Tao-Yi Lee */

strCellName="base_ind_hud_cross";
pcDefinePCell(
list(ddGetObj(strLibName),strCellName, "layout"),
(
        (OD      "float" 60.0)
        (W       "float" 2.0)
        (S       "float" 2.0)
        (OPENING "float" 10.0)
        (TOP_ME "string" "9")
        (BTM_ME "string" "9")
        (under "boolean" t)
        (dummy "boolean" t)
        (DUMMYL "string" "RFVLSI")
)
let((P via_to_next C C2 pi OO OOC OOCH m_base_oct),
        P=W+S
        if(atoi(TOP_ME) - atoi(BTM_ME) <2
        then
                via_to_next = nil
        )
        pi = 3.141592
        C=roundtogrid(W*tan(pi/8))
        C2=roundtogrid(C/sqrt(2))
        OO = 2*W+S
        OOC = 2*W+S-2*C2
        OOCH = OOC/2

        m_base_oct = dbOpenCellViewByType(pcCellView~>lib "base_oct" "layout" );
        dbCreateParamInst(pcCellView m_base_oct
                        nil 0:0 "R0" 1
                        list(
                        list("OD" "float" OD)
                        list("W" "float" W)
                        list("LOP" "float" OOCH + roundtogrid(2*sqrt(2.0)- S)+0.01)
                        list("ROP" "float" OPENING)
                        list("MET" "int" 9)
                )) ; close dbCreateParamInst

        if(under
        then
                dbCreateParamInst(pcCellView
                        dbOpenCellViewByType(pcCellView~>lib "base_xfm_cross" "layout") nil (-OD/2):0 "R0
                            " 1
                        list(
                                list("WI" "float" S)
                                list("WO" "float" W)
                                list("S" "float" 0.0)
                                list("TOP_ME" "int" atoi(TOP_ME))
                                list("BTM_ME" "int" atoi(TOP_ME)-1)
                                list("dummy" "boolean" dummy)
                                )) ; close dbCreateParamInst
                dbCreateParamInst(pcCellView
                        dbOpenCellViewByType(pcCellView~>lib "base_xfm_cross" "layout") nil (-OD/2+P+W):0
                             "MY" 1
                        list(
                                list("WI" "float" S)
                                list("WO" "float" W)
                                list("S" "float" 0.0)
                                list("TOP_ME" "int" atoi(TOP_ME))
                                list("BTM_ME" "int" atoi(TOP_ME))
                                list("viat" "boolean" nil)
                                list("viad" "boolean" nil)
                                list("dummy" "boolean" dummy)
                                )) ; close dbCreateParamInst
                )
))
```