International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 2 Issue 7 ǁ July. 2014 ǁ PP.01-07

Application of existing MEMS Technology in the MEMS Device Design by means of MEMS Xplorer and SKILL

Nikolay Delibozov[1] Vladimir Stavrov[2]
[1]*(Microelectronics, Technical University of Sofia, Bulgaria)*
[2](AMG Technology Ltd., Industrial Zone Microelectronica, Botevgrad, Bulgaria)

ABSTRACT : *SKILL is not an acronym, it is a name. SKILL was originally based of Lisp called "Franz Lisp" created at UC Berkeley by students of Professor Richard J. Fateman. SKILL is used as a scripting language as well as a description language for PCells which are used in many EDA software suites by Cadence Design Systems (for example Cadence Allegro and Cadence Virtuoso). A parameterized cell or PCell is a graphic, programmable cell that lets to be created a customized instance each time when it is placed. The created PCell is called a master. The master PCell is compiled and stored in the database in the form of SKILL procedure. MEMS Xplorer is fully integrated in the Cadence Virtuoso design platform. Combined with the Cadence Custom IC Tool suite MEMS Xplorer creates a very powerful and robust solution for both MEMS and IC design.*

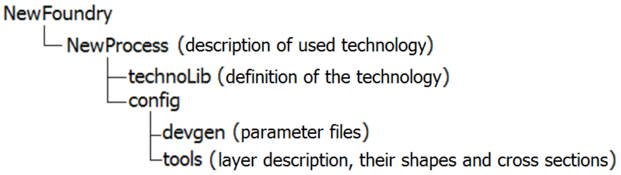
Keywords – *design; library; MEMS; SKILL; technology*

I. INTRODUCTION

In this paper is presented the use of SKILL language in order to design a MEMS device. MEMS devices can be presented with multi-physics signals in mechanical, thermal, magnetic, fluidic, optical, and electrostatic domains in Verilog-A. MEMS devices are designed by MEMS Xplorer DS which provides a set of libraries containing 34 components from which complete MEMS devices can be built [1]. Any of the supplied devices can be adapted to user-defined technology as well as can be built brand new devices in the user-defined technology. In order to create a new user-defined technology is used SKILL description language. SKILL is a Lisp dialect used as a scripting language as well as a description language for PCells which are used in many EDA software suites by Cadence Design Systems (e.g. Cadence Allegro and Cadence Virtuoso). SKILL is based on Lisp called "Franz Lisp" created at UC Berkeley by the students of Professor Richard J. Fateman. Historically, SKILL was known as IL. SKILL was a library of IL functions. A parameterized cell or PCell is a graphic, programmable cell that lets to be created a customized instance each time when it is placed. This cell is created just as creation of any other layout cell, and then necessary parameters are assigned to it. The created PCell is called a master. A master is the combination of the drawn graphic layout to make a cell and the assigned parameters to it. After compiling of the master PCell, it has to be stored in the database in the form of a SKILL procedure. When an instance of a PCell is placed, the defaults can be accepted or any of the values for the assigned parameters can be changed. Using the default values creates an instance identical to the master cellview.

II. MEMS DEVICE CREATION AND CUSTOMIZATION

MEMS devices can be designed by MEMS Xplorer within Virtuoso. MEMS physical design can also be created within the Cadence tool. MEMS Xplorer combined with the Cadence Custom IC Tool suite creates a very powerful and robust solution for both MEMS and IC design. The MEMS Xplorer modules are completely integrated in the Cadence environment making it very easy to learn and use. Functionalities include mixed MEMS/IC schematic capture and simulation, full custom mask layout capability and verification, 3D model generation and visualization, behavioral model creation and links to 3D analysis packages. MEMS Xplorer DS provides a set of libraries containing 34 components each of them with own PCell. In order to create a new component has to be created new technology and new process located in new technology directory Fig. 1.



```
NewFoundry
└NewProcess (description of used technology)
        ├technoLib (definition of the technology)
        └config
              ├devgen (parameter files)
              └tools (layer description, their shapes and cross sections)
```

Figure.1:New technology – directory structure.

A technology file then must be created describing the used layers, DRC rules, etc. The technology file is described using SKILL description language. In order to create new technology has to be created a foundry module called with the name of the new foundry. New foundry has to contain new process. New process contains a "technoLib" directory which defines new technology. "technoLib" directory should contain a binary file named "techfile.cds" which is so called technology file. All parameters files of the new devices are saved under "NewProcess/config/devgen" directory. Any of the supplied device generators can be adapted to user-defined technology. The adaptation of a device generator to a user-defined technology follows two steps:

- Customization of the parameters file for each desired device generator.
- Creation of the PCells inside the MEMS Xplorer DS environment.

The parameter file of a device generator is a text file containing all its necessary variable declaration. The adaptation of a device generator only consists in modifying the declared variables values. Generally the layers will be affected.

III. USES OF PCELL

A parameterized cell or PCell is a graphic, programmable cell that lets to be created a customized instance each time when it is placed [2]. This cell is created just as creation of any other layout cell, and then necessary parameters are assigned to it. When an instance of a PCell is placed, the defaults can be accepted or any of the values for the assigned parameters can be changed. The Virtuoso layout editor treats instances of PCells just like it treats instances of all other predefined cells. An instance of a PCell is almost identical to an instance of any other cellview to all applications that use the Cadence Design Framework II database.

Finally, complete content and organizational editing before formatting. When proofreading spelling and grammar must be addressed the following items:

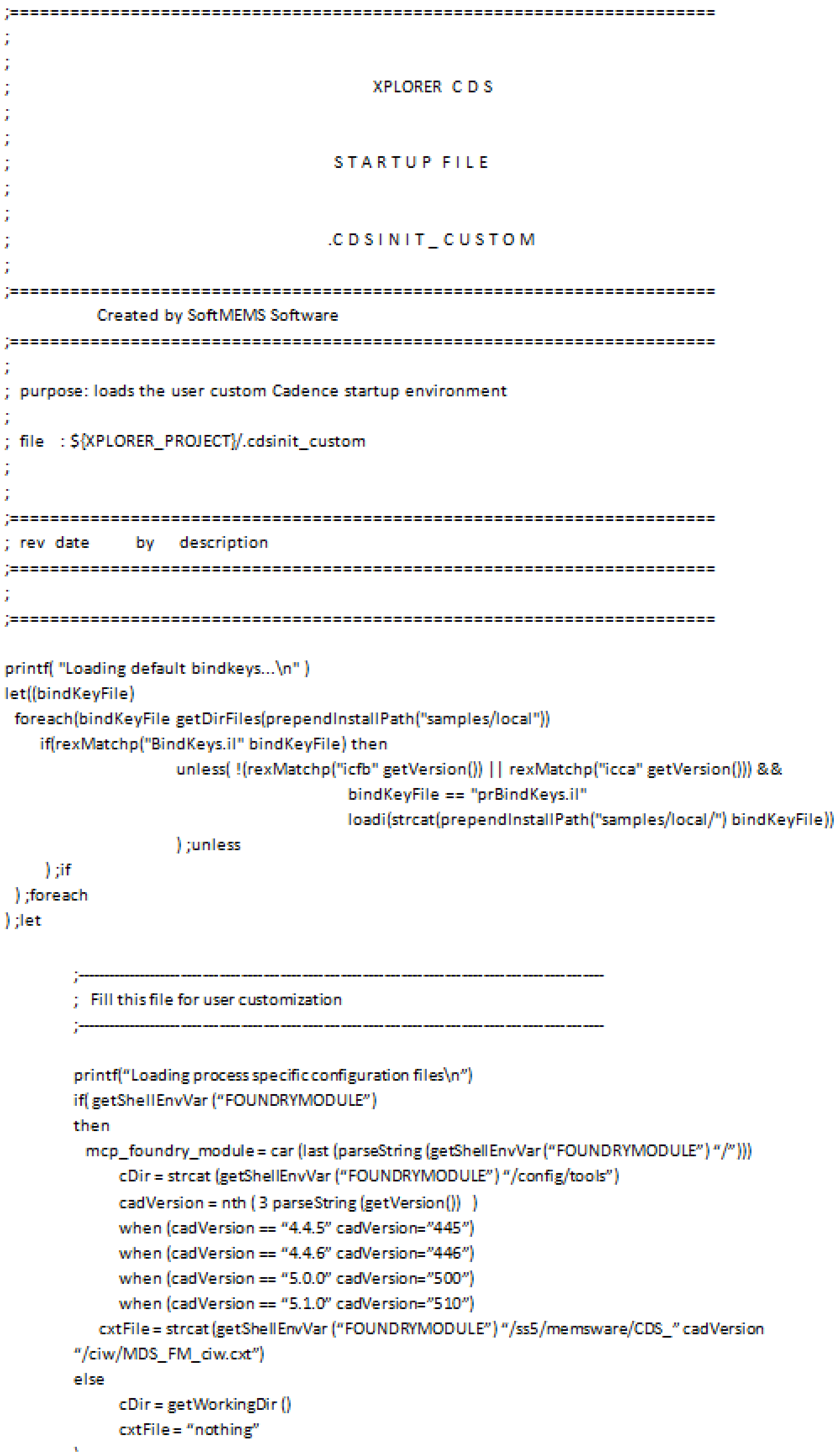
Advantages in using a PCell

PCells are used to:

- Speed up entering layout data by eliminating the need to create duplicate version of the same functional part.
- Save disk space by creating a library of cells for similar parts that are all linked to the same source.
- Eliminate errors that can result in maintaining multiple versions of the same cell.
- Eliminate the need to explode levels of hierarchy when a small detail of a design is changed.

IV. INITIALIZATION FILES

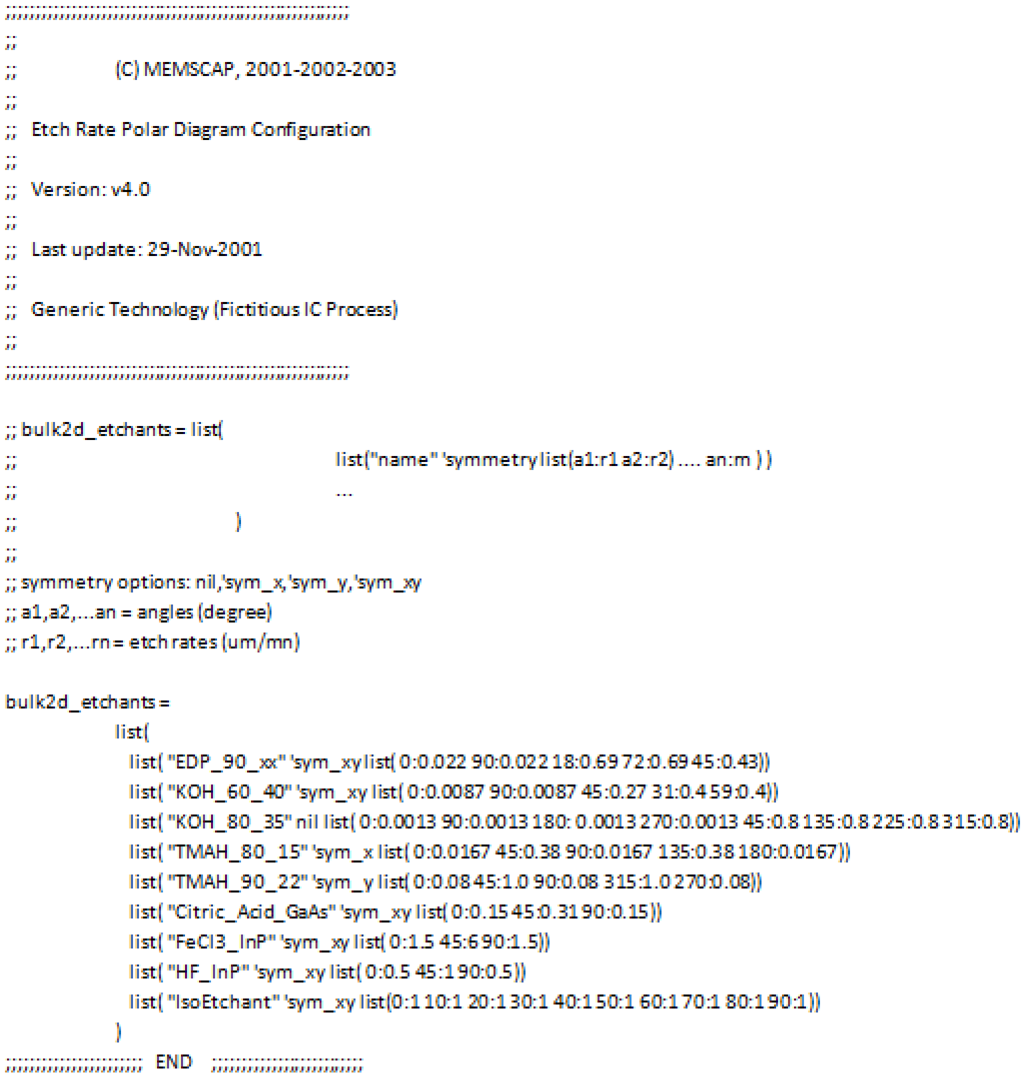
In the working directory, the ".cdsinit_custom" file is created and modified Fig. 2. When PCells section appears with sets of created device generators library and list the loading instruction so that Cadence correctly evaluates the customized PCells, a new session for this new technology could be launched.



```
;========================================================================
;
;
;                                  XPLORER  C D S
;
;
;                              S T A R T U P  F I L E
;
;
;                             .C D S I N I T _ C U S T O M
;
;========================================================================
         Created by SoftMEMS Software
;========================================================================
;
; purpose: loads the user custom Cadence startup environment
;
; file   : ${XPLORER_PROJECT}/.cdsinit_custom
;
;
;========================================================================
; rev  date       by     description
;========================================================================
;
;========================================================================

printf( "Loading default bindkeys...\n" )
let((bindKeyFile)
  foreach(bindKeyFile getDirFiles(prependInstallPath("samples/local"))
      if(rexMatchp("BindKeys.il" bindKeyFile) then
                        unless( !(rexMatchp("icfb" getVersion()) || rexMatchp("icca" getVersion())) &&
                                          bindKeyFile == "prBindKeys.il"
                                          loadi(strcat(prependInstallPath("samples/local/") bindKeyFile))
                        ) ;unless
       ) ;if
  ) ;foreach
) ;let

        ;-----------------------------------------------------------------------------
        ;  Fill this file for user customization
        ;-----------------------------------------------------------------------------

        printf("Loading process specific configuration files\n")
        if( getShellEnvVar ("FOUNDRYMODULE")
        then
          mcp_foundry_module = car (last (parseString (getShellEnvVar ("FOUNDRYMODULE") "/")))
              cDir = strcat (getShellEnvVar ("FOUNDRYMODULE") "/config/tools")
              cadVersion = nth ( 3 parseString (getVersion())   )
              when (cadVersion == "4.4.5" cadVersion="445")
              when (cadVersion == "4.4.6" cadVersion="446")
              when (cadVersion == "5.0.0" cadVersion="500")
              when (cadVersion == "5.1.0" cadVersion="510")
            cxtFile = strcat (getShellEnvVar ("FOUNDRYMODULE") "/ss5/memsware/CDS_" cadVersion
        "/ciw/MDS_FM_ciw.cxt")
        else
              cDir = getWorkingDir ()
              cxtFile = "nothing"
        )
```

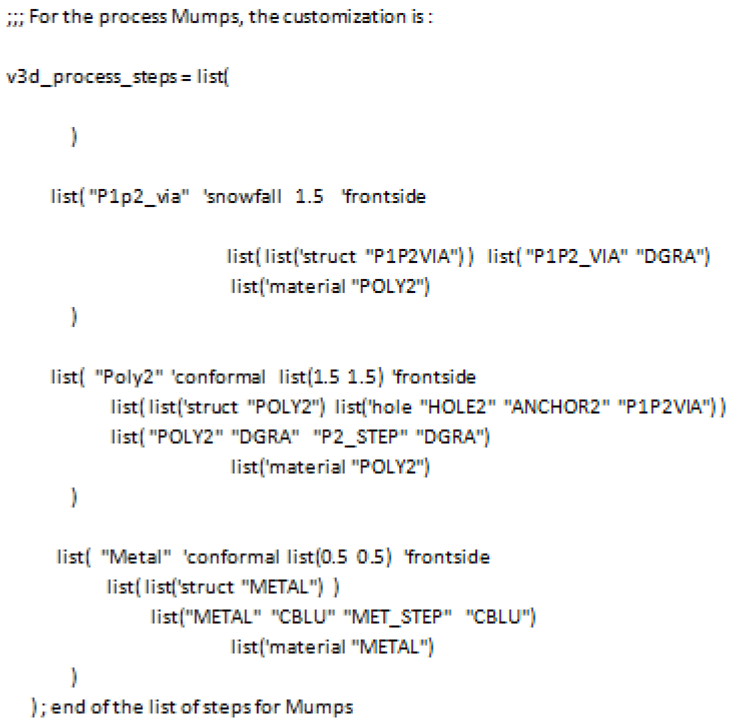
Figure.2:".cdsinit_custom" file.

The ".cdsinit_custom" file contains user customization and details of PCells as well as "process.var", "3DModel.var", "XS_layers.var", and "diagr.var" parameters initialization files (Figs. 3, 4, 5, and 6). These files are used when a new MEMS device is created and when simulations with the new device is made.



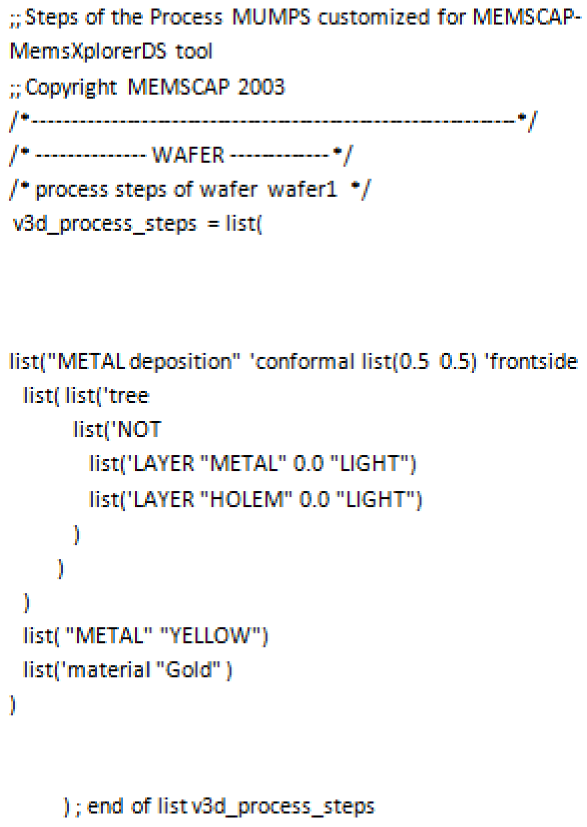
```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;
;;              (C) MEMSCAP, 2001-2002-2003
;;
;;  Etch Rate Polar Diagram Configuration
;;
;;  Version: v4.0
;;
;;  Last update: 29-Nov-2001
;;
;;  Generic Technology (Fictitious IC Process)
;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;; bulk2d_etchants = list(
;;                                              list("name" 'symmetry list(a1:r1 a2:r2) .... an:m ) )
;;                                              ...
;;                              )
;;
;; symmetry options: nil,'sym_x,'sym_y,'sym_xy
;; a1,a2,...an = angles (degree)
;; r1,r2,...rn = etch rates (um/mn)

bulk2d_etchants =
              list(
                list( "EDP_90_xx" 'sym_xy list( 0:0.022 90:0.022 18:0.69 72:0.69 45:0.43))
                list( "KOH_60_40" 'sym_xy list( 0:0.0087 90:0.0087 45:0.27 31:0.4 59:0.4))
                list( "KOH_80_35" nil list( 0:0.0013 90:0.0013 180: 0.0013 270:0.0013 45:0.8 135:0.8 225:0.8 315:0.8))
                list( "TMAH_80_15" 'sym_x list( 0:0.0167 45:0.38 90:0.0167 135:0.38 180:0.0167))
                list( "TMAH_90_22" 'sym_y list( 0:0.08 45:1.0 90:0.08 315:1.0 270:0.08))
                list( "Citric_Acid_GaAs" 'sym_xy list( 0:0.15 45:0.31 90:0.15))
                list( "FeCl3_InP" 'sym_xy list( 0:1.5 45:6 90:1.5))
                list( "HF_InP" 'sym_xy list( 0:0.5 45:1 90:0.5))
                list( "IsoEtchant" 'sym_xy list(0:1 10:1 20:1 30:1 40:1 50:1 60:1 70:1 80:1 90:1))
              )
;;;;;;;;;;;;;;;;;;;;;;;;;;  END   ;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

Figure.3:"diagr.var" file.



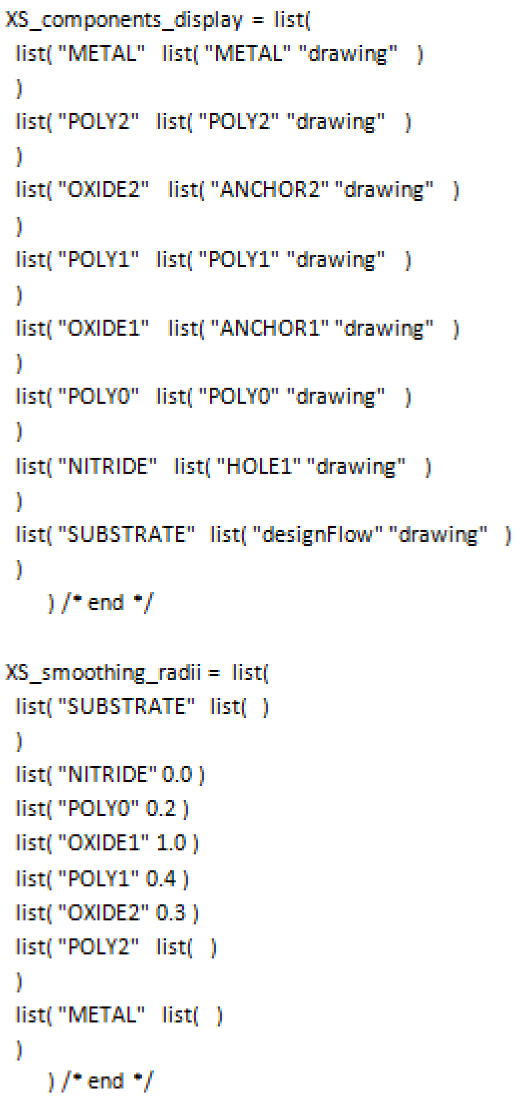
```
;;; For the process Mumps, the customization is :

v3d_process_steps = list(

        )

      list( "P1p2_via"  'snowfall  1.5   'frontside

                               list( list('struct  "P1P2VIA") )  list( "P1P2_VIA"  "DGRA")
                               list('material "POLY2")
        )

      list(  "Poly2"  'conformal  list(1.5 1.5)  'frontside
              list( list('struct  "POLY2")  list('hole  "HOLE2"  "ANCHOR2"  "P1P2VIA") )
              list( "POLY2"  "DGRA"   "P2_STEP"  "DGRA")
                               list('material "POLY2")
        )

       list(  "Metal"  'conformal list(0.5  0.5)  'frontside
              list( list('struct "METAL")  )
                    list("METAL"  "CBLU"  "MET_STEP"   "CBLU")
                               list('material "METAL")
        )
   ) ; end of the list of steps for Mumps
```

Figure.4:“3DModel.var” file.



```
;; Steps of the Process MUMPS customized for MEMSCAP-
MemsXplorerDS tool
;; Copyright MEMSCAP 2003
/*--------------------------------------------------------*/
/* -------------- WAFER ------------*/
/* process steps of wafer wafer1 */
 v3d_process_steps = list(



list("METAL deposition"  'conformal list(0.5  0.5) 'frontside
  list( list('tree
         list('NOT
           list('LAYER "METAL" 0.0 "LIGHT")
           list('LAYER "HOLEM" 0.0 "LIGHT")
         )
       )
  )
  list( "METAL" "YELLOW")
  list('material "Gold" )
)


       ) ; end of list v3d_process_steps
```

Figure.5:“process.var” file.



```
XS_components_display = list(
 list( "METAL"   list( "METAL" "drawing"   )
 )
 list( "POLY2"   list( "POLY2" "drawing"   )
 )
 list( "OXIDE2"   list( "ANCHOR2" "drawing"   )
 )
 list( "POLY1"   list( "POLY1" "drawing"   )
 )
 list( "OXIDE1"   list( "ANCHOR1" "drawing"   )
 )
 list( "POLY0"   list( "POLY0" "drawing"   )
 )
 list( "NITRIDE"   list( "HOLE1" "drawing"   )
 )
 list( "SUBSTRATE"   list( "designFlow" "drawing"   )
 )
     ) /* end */

XS_smoothing_radii = list(
 list( "SUBSTRATE"   list(   )
 )
 list( "NITRIDE" 0.0 )
 list( "POLY0" 0.2 )
 list( "OXIDE1" 1.0 )
 list( "POLY1" 0.4 )
 list( "OXIDE2" 0.3 )
 list( "POLY2"   list(   )
 )
 list( "METAL"   list(   )
 )
     ) /* end */
```

Figure.6:“XS_layers.var”

V. MEMS DESIGNING

When the directory structure of a new technology is created and all necessary files are placed on the corresponding directories, new device generators are created. Every new device generator which is created is placed into a corresponding device category. These device generators are used to help designers when they create a new MEMS device. An example of such MEMS device is shown on the Fig. 7. Several technological layers are used during MEMS device designing.

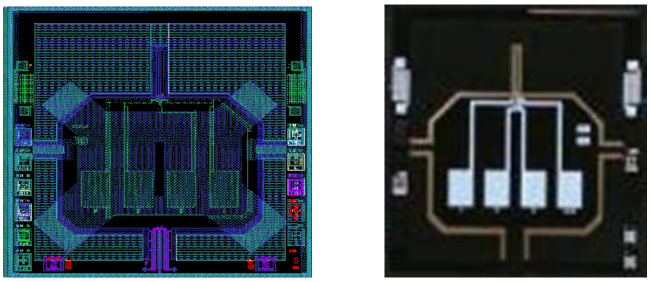


Figure.7: a) Designed MEMS Device b) Fabricated MEMS Device.

For fabrication silicon wafers are used. Technology-dependent modeling parameters required for electromechanical design include nominal, minimum and maximum values for layer thicknesses, structural sidewall overetch and angles, structural release undercut, layer sheet resistance, effective material density, effective Young's modulus, residual stress, and residual stress gradient [3]. Some of these parameters are shown in the Table 1.

Property	Units	Value
Young's Modulus (E)	GPa	0.2
Density	Kg/μm^3	2330
Relative Permittivity	ε	11.7
Poisson's ratio	ν	0.33

Table 1. Technology – dependent modeling parameters

The beam structure has dimensions described in Table 2.

Table 2. Cantilever Beam Dimensions

Dimension	Values
Length	100E-6
Breadth	10E-6
Thickness	2E-6

VI. CONCLUSION

In a conclusion it can be said that SKILL can be used for describing new technology processes and parameters initialization files when new MEMS devices are created. A parameterized cell or PCell is important part of any new MEMS device. The created PCell is called a master. A master PCell is the combination of the drawn graphic layout to make a cell and the assigned parameters to it. After compiling the master PCell, it is stored in the database in the form of a SKILL procedure. In order to create a new component has to be created new technology and new process located in new technology directory. Technology file then must be created describing used layers, DRC rules, etc. The technology file is described using SKILL description language. In the working directory, the ".cdsinit_custom" file is created and modified. When PCells section appears with sets of created device generators library and list the loading instruction so that Cadence correctly evaluates the customized PCells, a new session for this new technology could be launched. Every user-created device generator is placed in the corresponding device category of the new technology for this new type MEMS device. Every one of the device generators are used to help designers when they create a new MEMS device. Several technological layers are needed for MEMS device designing.

MEMS device can be designed with any CAD software. However, design of MEMS devices, using device generators (each one for a separate device part) which are part of specific MEMS design software significantly reduce MEMS device design time and facilitate designers.

Every new technology requires creation of some new device generators and their placement into the corresponding device category.

VII. Acknowledgements

The research is related to the project No BG051PO001-3.3.06-0046 "Development support of PhD students, postdoctoral researchers and young scientists in the field of virtual engineering and industrial technologies". The project is implemented with the financial support of the Operational Programme Human Resources Development, co-financed by the European Union through the European Social Fund."

REFERENCES

[1] D. Nikolay, R. Rossen, and H. Marin, "*MEMS Design using MEMS Xplorer and SKILL*", Journal of the Technical University – Sofia, Plovdiv branch, vol. 19, pp. 193-196, May 2013, ISSN 1310-8271.

[2] SoftMEMS LLC, 2006, Manual, Los Gatos, CA, SoftMEMS LLC.

[3] José Mireles Jr., Humberto Ochoa, and Víctor Hinostroza, "*Design and Analysis of a MEMS Variable Capacitor using Thermal Actuators*", Computación y Sistemas vol. 10 No. 1, 2006, pp 1-15, ISSN 1405-5546.