©Journal of the Technical University – Sofia
Plovdiv branch, Bulgaria
"Fundamental Sciences and Applications" Vol. 19, 2013
International Conference Engineering, Technologies and System
TECHSYS 2013
BULGARIA

ПРОЕКТИРАНЕ НА MEMC, ИЗПОЛЗВАЙКИ MEMS XPLORER И SKILL

НИКОЛАЙ ДЕЛИБОЗОВ, РОСЕН РАДОНОВ, МАРИН ХРИСТОВ

Резюме: *Харви С. Натансон е американски електроинженер, който е изобрертил първото MEMC устройство.От 80-те години в изследователските лаборатории са се разработвали MEMC устройства. MEMC (Микро Електро-Механични Системи) са миниатюрни устройства, конструирани чрез комбиниране на механични части и електронни вериги, обикновено върху полупроводников чип с размери от десети до няколко хиляди микрометра. MEMC са използвани за правенето на сензори за налягане; температура; вибрации и химични сензори. В края на 90-те повечето от MEMC устройствата с различна чувствителност или задвижващи механизми са били произведени, използвайки микрообработена подложка от силиций, чрез повърхностно микрообработване и литогравски; галваноформиращи и (ЛИГА) процеси за формиране.*

Ключови думи: *SKILL, технологичен файл, MEMC*

MEMS DESIGN USING MEMS XPLORER AND SKILL

NIKOLAY DELIBOZOV, ROSSEN RADONOV, MARIN HRISTOV

Abstract: *Harvey C. Nathanson is an American electrical engineer who invented the first MEMS device. Since 1980s in the research labs have been developed MEMS devices. MEMS (Micro Electro Mechanical Systems) are miniature devices formed by combining mechanical parts and electronic circuits, typically on a semiconductor chip, with dimensions from tens to a few hundred micrometres. MEMS are used to make pressure, temperature, chemical and vibration sensors. At the end of 1990s, most of MEMS devices with various sensing or actuating mechanisms were fabricated using silicon bulk micromachining, surface micromachining, and lithography, galvanoforming, moulding (LIGA) processes.*

Key words: *SKILL, technology file, MEMS*

1. Introduction

MEMS are not for any one application or device, nor are defined by a single fabrication process or limited to a few materials [1]. Their fabrication encompasses the advantages of miniaturization, multiple components, and microelectronics to design and construct integrated electromechanical systems. The three characteristic features of MEMS fabrication technologies are miniaturization, multiplicity, and microelectronics. Miniaturization enables the fabrication of compact, quick-response devices. Multiplicity refers to the batch fabrication inherent in semiconductor processing, which allows thousands or millions of components to be easily and concurrently fabricated. Microelectronics provides the intelligence to MEMS and allows the monolithic

merger of sensors, actuators, and logic to build closed-loop feedback components and systems. There are several methods for MEMS to be designed. In this paper is presented method using Cadence design environment, SKILL program language as well as MEMS Xplorer of the SoftMEMS software.

2. MEMS design technology

MEMS components can be classified into six individual categories [2]. These categories of MEMS components are based on their application. These categories include:

- Sensors

Sensors can be chemical, motion, inertial, thermal, and optical.

- Actuators

MEMS actuators can provide power using either an electrostatic or thermal stimulus.

- RF MEMS

RF MEMS are devices used to switch, transmit, filter, and manipulate radio frequency (RF) signals (fig.1).

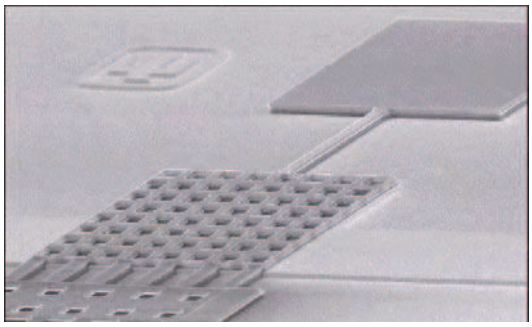


Fig. 1. *RF MEMS switch*

- Optical MEMS

Optical MEMS include optical switches and reflectors (fig.2).

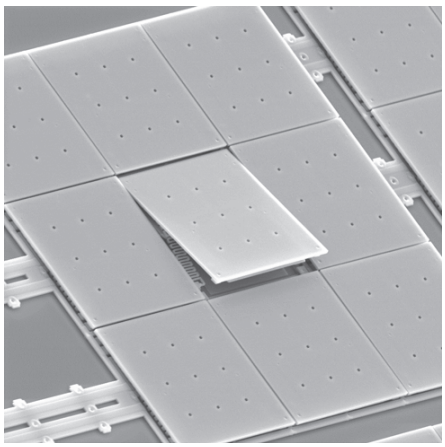


Fig. 2. *3x3 micromirror array with the center mirror actuated to 15° tilt position*

- Microfluidic MEMS

Microfluidic MEMS are designed to interact with fluid-based systems.

- Bio MEMS

Bio MEMS are designed to interact with proteins, biological cells, medical reagents, etc.

MEMS Xplorer provides a set of libraries containing 34 components from which complete MEMS devices can be built (fig.3).

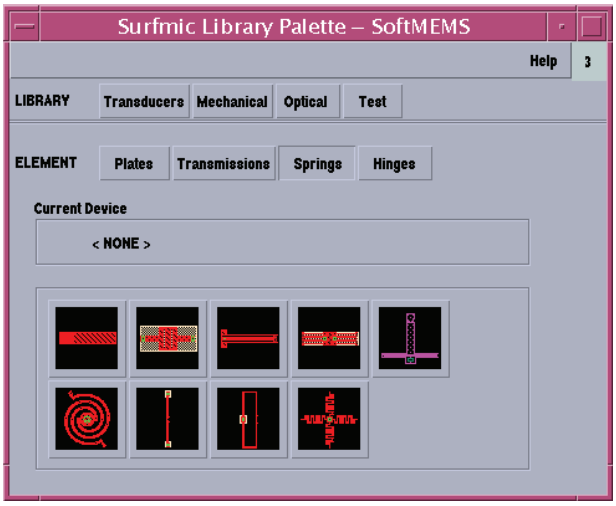


Fig. 3. *Library Components Window*

You can create your own technology as well. Libraries include basic and advanced device generators for BULK and the SURFACE micromachining technologies. When new technology is made all its features are described by SKILL language. SKILL provides a safe, high-level programming environment that automatically handles many traditional system programming operations. When create own user-defined technology is needed binary (ASCII) file **techfile.cds** so-called technology file and parameter files. Technology file consists of Layer Definitions, Layer Rules and Physical Rules. Each layer in Layer Definition is defined by a layer-purpose pair, which consists of a unique layer name and purpose combination (figs.4 a and b).

```
;*********************************
; LAYER DEFINITION
;*********************************
layerDefinitions(

 techPurposes(
 ;( PurposeName              Purpose#   Abbreviation )
 ;( ----------              --------   ------------ )
 ;User-Defined Purposes:
 ;System-Reserved Purposes:
  ( warning                  234        wng          )
  ( label                    237        lbl          )
  ( flight                   238        flt          )
  ( error                    239        err          )
  ( annotate                 240        ant          )
  ( drawing1                 241        dr1          )
  ( drawing2                 242        dr2          )

 ) ;techPurposes
```

Fig. 4a. *Layer Definition*

```
techLayers(
;( LayerName                Layer#     Abbreviation )
;( ---------                ------     ------------ )
;User-Defined Layers:
 ( poly                     1          pol          )
 ( anchor                   2          anc          )
 ( dimple                   3          dmp          )
 ( metal                    4          met          )
 ( holpoly                  5          hpo          )
 ( holmetal                 6          hme          )
 ( contact                  7          cnt          )
;System-Reserved Layers:
 ( Unrouted                 200        Unroute      )
 ( Row                      201        Row          )
 ( Group                    202        Group        )
 ( Cannotoccupy             203        Cannoto      )
 ( Canplace                 204        Canplac      )
 ( hardFence                205        hardFen      )
 ( softFence                206        softFen      )

) ;techLayers

techLayerPurposePriorities(
;layers are ordered from lowest to highest priority
;( LayerName                Purpose    )
;( ---------                -------    )
 ( background               drawing    )
 ( grid                     drawing    )
 ( grid                     drawing1   )
 ( axis                     drawing    )
 ( instance                 drawing    )
 ( instance                 label      )
 ( prBoundary               drawing    )


) ;techLayerPurposePriorities

techDisplays(
;( LayerName    Purpose      Packet          Vis Sel Con2ChgLy DrgEnbl Valid )
;( ---------    -------      ------          --- --- --------- ------- ----- )
 ( background   drawing      blacksolid_s     t nil t nil nil )
 ( grid         drawing      slate            t nil t nil nil )
 ( grid         drawing1     white            t nil t nil nil )
 ( axis         drawing      white            t nil t t nil )
 ( instance     drawing      red              t t t t nil )
 ( instance     label        red              t t t t nil )
 ( prBoundary   drawing      purple           t t t t nil )
 ( prBoundary   boundary     cyan             t t t t nil )


) ;techDisplays

techLayerProperties(
;( PropName              Layer1 [ Layer2 ]          PropValue )
)

) ;layerDefinitions
```

Fig. 4b. *Layer Definition*

The layer name usually indicates a type of manufacturing material. The purpose indicates the use of layer or material. Multiple layers with the same name but different purposes can be created. Layer Rules must be specified to establish the relationships and interactions between layers (fig.5).

```
;*********************************
; LAYER RULES
;*********************************
layerRules(

 streamLayers(
 ;( layer       streamNumber    dataType        translate )
 ;( -----       ------------    --------        --------- )
  ( ("poly" "drawing")          1               0               t       )
  ( ("anchor" "drawing")        2               0               t       )
  ( ("dimple" "drawing")        3               0               t       )
  ( ("metal" "drawing")         4               0               t       )
  ( ("holpoly" "drawing")       5               0               t       )
  ( ("holmetal" "drawing")      6               0               t       )
  ( ("contact" "drawing")       7               0               t       )
 ) ;streamLayers

) ;layerRules
```

Fig. 5. *Layer Rules*

Layer rules define the following:

- Via layers that connect two conducting layers.
- Layers that are physically and electrically equivalent.
- Stream translation data for a layer.

Physical rules must be specified to establish spacing within and between objects in the design and to specify the grid snapping (fig.6).

```
;*********************************
; PHYSICAL RULES
;*********************************
physicalRules(

 mfgGridResolution(
       ( 0.100000 )
 ) ;mfgGridResolution

) ;physicalRules
```

Fig. 6. *Physical Rules*

Physical rules define the following:

- Spacing information for individual objects, for example, width and notch spacing rules.
- Spacing information for two objects, for example, the minimum distance allowed between objects on the same layer or different layers.
- The amount of space required when one object encloses another.
- The manufacturing grid resolution.

Spacing rules specify the distance required between layers and the width of objects and paths.

The parameter file of a device generator is a text file containing all its needed variables declaration (fig.7).

```
;;;; Design Parameters
;;;; Note : all design parameters must be positive.

;;;;      : tsl_lock_length_min must be >  tsl_lock_head_length_min +
;;;;                                       tsl_neck_lock_head_length_min +
;;;;                                       tsl_torsional_beam_width_min
;;;;      : tsl_lock_head_length_min must be > tsl_step_lock_head_width_min
;;;;      : tsl_lock_head_width_min must be > tsl_neck_centering_min + tsl_neck_lock_head_width_min

tsl_lock_head_width_min             = 5
tsl_lock_head_length_min            = 5
tsl_step_lock_head_width_min        = 1
tsl_neck_lock_head_length_min       = 2
tsl_neck_lock_head_width_min        = 2
tsl_neck_centering_min              = 0
tsl_lock_length_min                 = 9
tsl_torsional_beam_length_min       = 5
tsl_torsional_beam_width_min        = 2
tsl_torsional_beams_separation_min  = 2
tsl_connecting_bar_width_min        = 2
tsl_anchor_length_min               = 13
tsl_anchor_width_min                = 13

;;;; Layers
;;;; Note : all layers must exist in your techno file.
tsl_layer1              = "metal"
tsl_layer2              = "anchor"

;;;; Design Rules
tsl_enc_layer12         = 5
```

Fig. 7. *Parameter File*

3. Compiling Technology File

When ASCII technology file is ready it is necessary to be compiled in order to create technology library (fig.8).

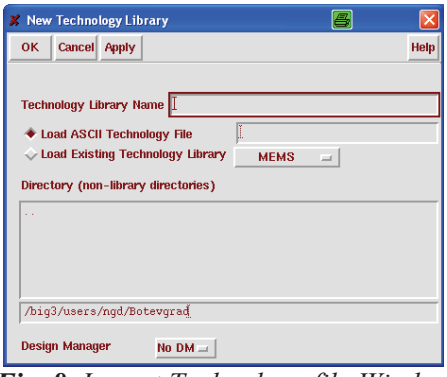


Fig. 8. *Import Technology file Window*

After compiling of ASCII technology file it is necessary to check the file (fig.9).

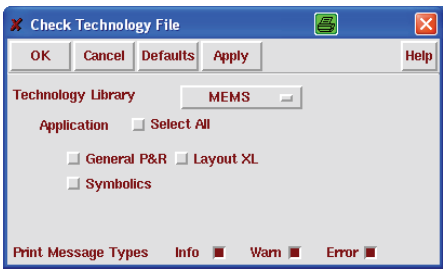


Fig. 9. *Check Window*

Technology library must be attached to a design library to use it in design process (fig.10).

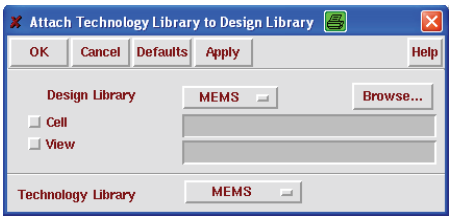
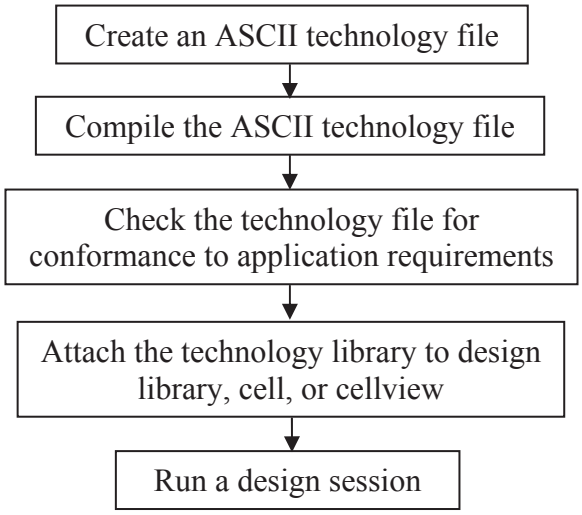


Fig. 10. *Attach Window*

When a technology library is attached the system updates properties of the design library and updates the technology devices in each cellview to reference the new technology file.

4. Conclusion

As a conclusion it can be said that MEMS can be classified into several categories. Each category consists of several components which help us to build desired MEMS device. When we want to make new MEMS device using different type of technology an own user-defined technology is necessary to be created. When new technology is made all its features are described by SKILL language. When we use own user-defined technology we need a technology file and parameter files. The major steps for technology file are described in the following chart flow:



Every design uses a technology library.

ACKNOWLEDGEMENT

This paper was financially supported by contract No. 122ПД0038-03, R&D Department, Technical University of Sofia.

REFERENCE

1. Mohamed Gad-el-Hak MEMS Applications, CRC Pres Taylor & Francis Group, 2006, ISBN: 0-8493-9139-3.

2. Walraven J. Future Challenges for MEMS Failure Analysis, Sandia National Laboratories. Albuquerque, NM USA, 2003.

Department of Microelectronics
Faculty of Electronic Engineering and Technologies
ECAD Laboratory
Technical University – Sofia
8 Kliment Ohridski blvd.
1000 Sofia
BULGARIA
E-mail: n.delibozov@ecad.tu-sofia.bg
E-mail: mhristov@ecad.tu-sofia.bg
E-mail: Rossen.Radonov@ecad.tu-sofia.bg

Постъпила на 17.02.2013 г.