

CREATING A MEMS DEVICE: HOW TO GET STARTED?

NICOLAS WILLIAMS, PRODUCT MARKETING MANAGER, MENTOR GRAPHICS



A M S D E S I G N & V E R I F I C A T I O N

W H I T E P A P E R

www.mentor.com

INTRODUCTION

Even if you are reinventing the wheel (of a drive motor), the goal is to get the job done as easy and quickly as possible. Using the MEMS Library within Tanner L-Edit, you can rapidly assemble your MEMS (microelectro-mechanical systems) layout structure by selecting from a wide variety of primitives. Using the Tanner design tools, you can implement a complete, correct-by-construction MEMS process using the design flow in Figure 1.

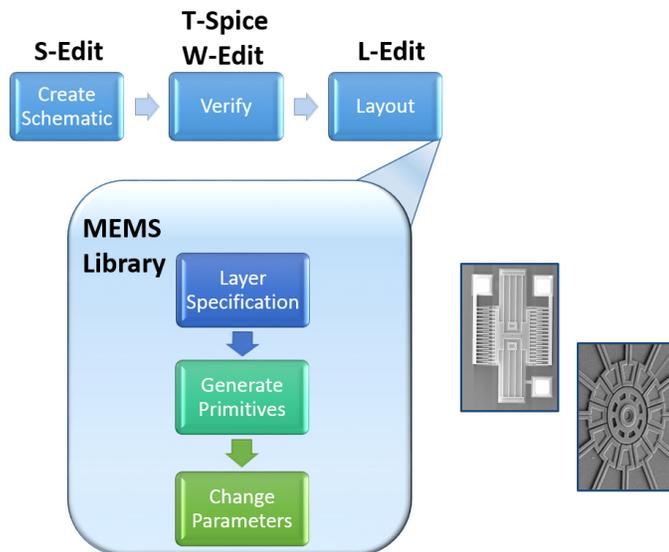


Figure 1: The MEMS creation design flow.

There are several traditional design flows that exist in the world of MEMS creation. Figure 1 shows a top-down flow that bridges the different design domains required for MEMS design. This flow provides a seamless solution for design teams working to implement an IC that contains digital, analog, and MEMS elements.

PRELIMINARY STEPS: CREATING AND VERIFYING THE SCHEMATIC

First, start by creating a new cell for a MEMS device in the S-Edit schematic editor and then connecting instances of the primitives that comprise the device. Then, using T-Spice, you can analyze the design and then interact with waveforms using the W-Edit waveform viewer in order to determine if the device works as expected.

SPECIFYING LAYER INFORMATION

After you are satisfied with the schematic, create a physical layout of the MEMS device using L-Edit. It is within L-Edit that you have access to the MEMS primitives within the MEMS Library.

You have control over the layer information for each of the primitives and can specify layer information for the two fundamental types of MEMS components:

- **Surface Micro-Machining.** You can specify the number of structural layers and for each layer define parameters. You can also specify design rules for each layer, such as spacing, enclose, and minimum line requirements.
- **Fluidic Channels.** You can specify the number of structural layers and define the set of parameter that appear for each fluidic primitive in the MEMS Library.

This layer information applies automatically to each appropriate primitive in the MEMS Library.

CREATING PRIMITIVES USING THE MEMS LIBRARY

The MEMS Library contains over 40 parameterized primitives to layout a MEMS device. Library primitives are continually added to the library with each tool release. The primitives available in the MEMS Library are grouped by category as Table 1 presents.

TABLE 1: AVAILABLE PRIMITIVES

Category	Sample Primitives
Active Elements	Motors: linear comb & side-drive unidirectional, bi-directional, & rotary
Passive Elements	Linear crab leg & folded beam, dual Archimedean spiral, journal bearing, & serpentine spring
Test Elements	Dielectric isolation, crossover, columns, Guckel ring tests, & multi-layer pad
Thermal Elements	Horizontal & vertical heat actuators, thermal forward & backward motors
Optical Elements	Direct & indirect side deflective mirrors
Resonator Elements	Plate, comb drive, folded spring, ground plate, & bonding pad
Fluidic Elements	Channel, elbows, T & Y junctions, cross, U-turn, transition, & reservoir

To create MEMS devices, simply select each primitive from the MEMS Library by using the Library Palette as Figure 2 shows.

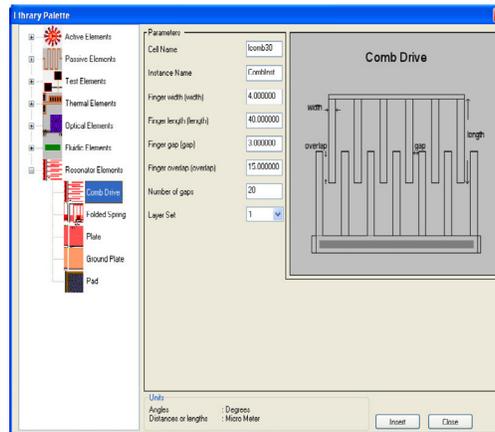


Figure 2: The Library Palette is used to assemble the MEMS device for layout.

For example, to create a lateral, comb-drive resonator, you instantiate the required primitives and then connect them within the layout. To illustrate how straightforward the process is, let's create a MEMS transducer that becomes a sensor by exploiting the high sensitivity of its resonant frequency to physical parameters. Figure 3 illustrates the creation steps:

1. Instantiate the plate.
2. Instantiate a comb drive and make a copy of it.
3. Connect the two comb drives to the plate.
4. Instantiate the folded spring, flip it, and make a copy of it.
5. Connect the two folded springs to the plate.
6. Instantiate the ground plate and connect it to the structure on the proper layer.
7. Instantiate three bonding pads and connect them to the proper layers and elements.
8. For each instantiated primitive, adjust the parameters to meet your needs (see the next section).

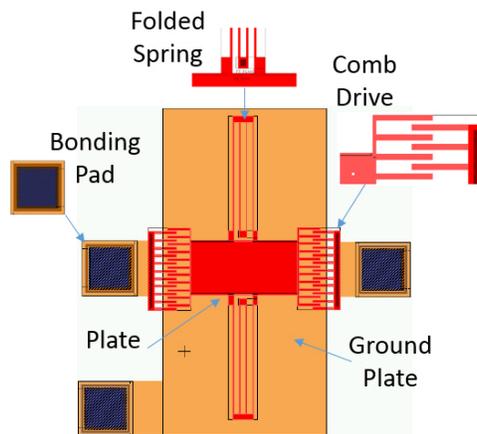


Figure 3: A completed lateral-drive resonator.

CHANGING PARAMETERS

You can change any of the defined parameters to affect each element of the MEMS structure. For example, Figure 4 shows that changing one key parameter of a harmonic, side drive motor primitive, quickly modifies the layout.

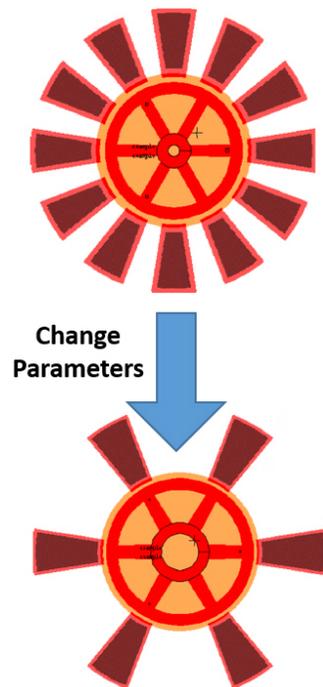


Figure 4: Quickly alter a motor primitive by changing just one parameter.

Parameterized primitives allow you to quickly build up MEMS structures by instantiating them, copying and pasting, changing the orientation, and then changing parameters to make each primitive unique. Each primitive contains a default set of parameter values which allows you to place them quickly and then select each primitive and change the parameters afterwards.

MAKING YOUR OWN PRIMITIVES

The MEMS Library contains over 40 primitives that you can build up into MEMS structures. But, because the world of MEMS design is vast, there could be an occasion when you need to define your own primitive and make it available on the Library Palette for your team to use. Figure 5 shows the process for making your own primitives.



Figure 5: An overview of the process for making your own primitives.

The steps to create your own primitive are:

1. Make a schematic symbol. Define the symbol body, its properties, and ports using S-Edit.
2. Create SPICE model. To simulate the primitive in T-Spice, you need to define a SPICE model. MEMS devices can be multiple-domain systems: electrical and non-electrical. This requires definition of the mapping between SPICE electrical variables and the variables in the non-electrical domain. For example, pressure on a plate can be mapped to current in the electrical domain.
3. Write layout generator. Using the C programming language and L-Edit UPI calls, define a layout generator for the custom primitive. The C program takes in parameters for the primitive and generates the required layout information.

COMPLETING THE PROCESS

After you have constructed your MEMS device layout using the Library Palette in L-Edit, there are a few final steps, as Figure 6 shows.

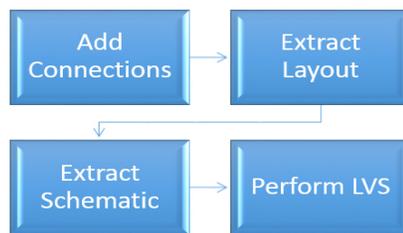


Figure 5: The steps to complete the MEMS design.

Completing the process includes:

- Add connections. A commonly-overlooked step in the completion process is adding connections between the primitives in order to correctly extract a SPICE netlist. Each primitive instanced using the Library Palette automatically has the correct ports created. L-Edit can make connections between two or more primitive ports. In our resonator example, you would connect a port from the plate to a port on one of the comb drives.
- Extract Layout. Extracting the layout using L-Edit produces a SPICE netlist that consists of the MEMS device and connectivity information. You use this extracted information for comparing the layout to the schematic (known as layout versus schematic, or LVS).
- Extract Schematic. For LVS, you extract a schematic netlist using S-Edit. This netlist contains the device description, connectivity, and geometrical parameters.
- Perform LVS. With the LVS tool, you can compare the layout to the schematic to ensure that both describe the same device.

Each MEMS device that you create can be tailored to your fabrication process in order to guarantee that the structures adhere to design rules. L-Edit lets you choose from a set of pre-defined fabrication processes. Process information includes layer definitions, design and extraction rules, model parameter values, macros, and process definitions.

3D ANALYSIS AND SYSTEM-LEVEL SIMULATION

You can automatically generate a 3D view of your device using the completed layout data. Interact with the 3D view and add technology information before sending the design to a 3rd-party, 3D analysis tool to examine physical interactions between multiple domains, such as mechanical, thermal, and electrostatic. You can then extract a behavioral model of the MEMS device for system-level simulations based on the 3D analysis data.

CONCLUSION

With the Tanner MEMS Library collection of parameterized primitives, you can quickly create complex surface micro-machined or fluidic MEMS devices. Instancing each primitive, setting the parameters, and then assembling the layout in L-Edit allows you to quickly define and analyze the device. You might be reinventing the MEMS motor wheel, but you are completing that task as fast as possible using the Tanner toolset.

For the latest product information, call us or

www.mentor.com

©2015 Mentor Graphics Corporation, all rights reserved. This document contains information that is proprietary to Mentor Graphics Corporation and may be duplicated in whole or in part by the original recipient for internal business purposes only, provided that this entire notice appears in all copies. In accepting this document, the recipient agrees to make every reasonable effort to prevent unauthorized use of this information. All trademarks mentioned in this document are the trademarks of their respective owners.

Corporate Headquarters
Mentor Graphics Corporation
8005 SW Boeckman Road
Wilsonville, OR 97070-7777
Phone: 503.685.7000
Fax: 503.685.1204

Silicon Valley
Mentor Graphics Corporation
46871 Bayside Parkway
Fremont, CA 94538 USA
Phone: 510.354.7400
Fax: 510.354.7467

Europe
Mentor Graphics
Deutschland GmbH
Arnulfstrasse 201
80634 Munich
Germany
Phone: +49.89.57096.0
Fax: +49.89.57096.400

Pacific Rim
Mentor Graphics (Taiwan)
11F, No. 120, Section 2,
Gongdao 5th Road
HsinChu City 300,
Taiwan, ROC
Phone: 886.3.513.1000
Fax: 886.3.573.4734

Japan
Mentor Graphics Japan Co., Ltd.
Gotenyama Garden
7-35, Kita-Shinagawa 4-chome
Shinagawa-Ku, Tokyo 140-0001
Japan
Phone: +81.3.5488.3033
Fax: +81.3.5488.3004



Sales and Product Information
Phone: 800.547.3000
sales_info@mentor.com

North American Support Center
Phone: 800.547.4303

MGC 10-15 MISC2540-w