TRENDS IN MICRO- AND NANO SOFTWARE

ir. N.Olij (Niels), ir. A.F. Bakker (Arjen), dr.ir. H.H. van den Vlekkert* (Hans)
PhoeniX B.V., P.O. Box 545, 7500 AM Enschede, The Netherlands
http://www.phoenixbv.com, info@phoenixbv.com, Phone: +31-53-4836460
*LioniX B.V,P.O. Box 456, 7500 AH Enschede, The Netherlands
http://www.lionixbv.com, info@lionixbv.com, Phone: +31 (0)53 489 3827

Abstract
Micro Systems Technology (MST or MEMS) and Nano technologies are compared with Integrated Circuit electronics (IC) over a broad range of topics. A lot of similarities but also crucial differences exist between these fields, involving the applied machines and the basic technologies. The ultimate difference between them lies in the flexibility required in the MST. When IC builds the products based on the standard libraries that are applied numerous times, a typical MST product is unique.
This major difference eliminates the effective used of the IC's sophisticated and mature software tools in the MST field. This paper lists the current situation on the software tools and the trends that can be observed in the development of the software for MST. The requirements for the next generation software tools in the micro and nano systems technology area are given.

Introduction
Micro systems technology[1] (MST) was evolved from Integrated Circuit electronics (IC). Once started as an academic variation of a technology that was widely applied to create electronic devices, MST has evolved to a separate technology branch with IC. Some of the IC technology roots can still be found in MST, such as lithography, evaporation etc. Nevertheless, when IC mainly evolved towards downscaling of the feature size, MST is going towards the technology diversification. This is one of the reasons that the mature software available for IC is hardly applicable for the MST. This paper evaluates the differences between the MST and IC fields, lists the predicted trends and their implication on the software for MST. The result is a list of requirements that can be set to model the design software that is suitable to support professional MST development and manufacturing.

Physical domains
A difference between MST and IC is the domains in which their typical devices operate. While IC serves the electrical domains; MST serves a wide variety of physical domains, in which each market of the products is much smaller than IC. The physical domains that are typically used in MST are the mechanical, chemical, magnetic (including RF), optical, and the combinations of those domains. The combination among those domains, in addition to the electrical domain, increase the complexity of the design and manufacturing of the MST devices.

Technology
Another difference between the IC and MST is the technology applied to fabricate devices. The scope of the technology defined here is somewhat broader than only the actual processing on the machines. The technology here also includes the design, the documentation, the measurements and the process steps to fabricate the devices within the physical domains.
Most of the equipments used by the IC facility are also used by the MST, but MST uses these machines for different purposes with the IC. While IC pushes towards downscaling the feature's size and standardization, MST generally pushes towards broader applications of the equipment that can sometimes lead to a larger / thicker features (when homogeneity is the major challenge).
Since IC deals only with one physical domain (the electrical domain), the fabrication of the devices

---
[1] In this paper MST is used to what may also be referred to as MEMS, micro machines, etc. Also their Nano equivalents are captured in this for simplicity.

has been standardized optimally. Most of the processes are fixed because most of the developed devices are similar. MST is not a standard, neither in technology nor in materials due to the diversity of the involved physical domains. The processes to create an MST device can be described as "free", not bound to a standard set of actions.

Types of devices

One of the IC's key-success is that only a small amount of geometries involved, which can be assumed to be fixed and fully optimized. Typical involved components are transistor, resistor and capacitor. The fixed geometries are captured in the library of the standard elements that will build the final system/device. The functionality of an IC device is a function of the library elements, in which the actual geometry of the underlying components is irrelevant.

In MST, the geometry of the underlying components is a relevant topic. The functionality of an MST device depends on its geometry. The measurement range of a pressure transducer depends on the membrane size, the membrane thickness, the material constants, the sensing technology (piezo resistive, capacitive, oscillation).

This implies capturing MST devices in a library element as if it were for the IC devices practically impossible. MST requires much more flexibility to describe its ever unique functional characteristic: its geometry.

Documentation

One implication of IC's being build up from library elements is that the documentation becomes a routine job. The processes are fixed; the library elements are fixed; and thus documentation is a matter of describing the relation and the connection among the elements. This sort of documentation is optimally automated which in turn requires hardly any human interaction anymore.

The requirement for the MST documentation is flexibility, because the geometry is “free” and the process of creating the geometry is free. An automated MST documentation system should be very flexible and facilitate the freedom the designer requires to create the devices. The documentation system should, preferably, be able to use parameterization in combination with the geometry and technology.

Modeling and measurements

The trend in IC is to standardize as much as possible, to reach a high integration of models with detailed physical and functional aspects and to move to a high abstraction level. Abstraction reduces the calculation time for the ever increasing detail of the models. In MST the trends are also in this direction, but are much harder to achieve. Calculation time increases drastically by the increasing details of the models, as well as the increasing range of physical domains involved. Using IC tools to implement the MST models means a standardization of the MST technology, which by nature making it suitable only for limited application areas and thus market.

Measurement in IC is most of the time the standard measurements of variables, for which specialized equipment is used, that fit well to the models used. The amount of data is huge, but with the detailed and advanced models this can be reduced to a small set of parameters of a higher order model. At MST the relation to the actual processing is much closer. This means that a lot of "small things" have to be logged: small amounts (up to 10 points per measurement) of non-accurate values are obtained. Model parameters that should be extracted from the data extend far beyond the existing models for MST. Therefore, a suitable software package for MST should have the possibility to define parameters and to feed these back into the design.

The message of this chapter is that applying an IC software solution to a small MST market implies the costs of doing this job will never be earned back. Suitable software for MST should be flexible in technology as well as in the application area.

Design and fabrication trajectory

Numerous differences between IC and MST is a direct consequence of the fact that MST is hard to standardize: the design and the production process of IC is clearly separated, whereas in MST most of it is concentrated in only two or three people.

In IC a *system engineer* determines the function by combining standard components into a new component. These cells have been designed by a *component designer* who uses library elements, which in turn have been optimized by a *library designer*. The library designer is supported by *clean room specialists*, assisted by *clean room operators*. Since these functions are standardized, it is even possible to separate these functions beyond company borders. Fab-less design houses are common in IC.

Due to the high variation in applications and the lack of integrated design tools for designing, the design and the fabrication process in MST mostly do not extend further then a designer and a clean room specialist (sometimes assisted by a clean room operator). In MST a trend towards separation in the design process is preferred, but this fails often because of the complexity of the numerous process's variation. An MST design tool therefore should be flexible to handle the variations in the design processes, but also should be able to incorporate the high flexibility of the fabrication process into the design. The fab-less model is not so common[2]. Design houses that offer foundry services face the task to bring their designs fast into production at the foundry itself, because there is a lack of overall design systems and good functional models. Foundries that do bring products into production now need to invest a lot of effort to scale the design into a volume production with compatible version. The ideal MST design system should facilitate flexible communication using standardized documentation between foundry and design house to facilitate such a fab-less business model.

Quality and Costs

Because IC is a highly separated business, all segments of the chain can be optimized separately, which leads to a low pricing. Even when low amounts of products are needed there are specialized multi project wafer services (MPW) that can deliver within a reasonable budget. MPW services for MST are not so common; the MUMPS® process of MEMSCAP and the foundry service of Sensonor are good examples of MPW services. Their process is highly standardized and therefore the amount of geometries that can be obtained from such services is limited. Since the functionality of an MST structure is a function of its geometry, the amount of devices in which these services can be applied is also limited. Having software tools that facilitate communication from design into production and back will increase the utilization of such foundry services. Since a lot of IP can be involved, protecting this IP is a requirement in these communications and the foundry service range of services is therefore dependent on such protection.

Documentation is an integrated part of the design process, so most companies implement their own documentation system. Usually, the interfacing among the elements in this form of documentation is captured in system procedures. Because of the complexity, this form of documentation is typically has a form of "Quality by Inspection". Ideally good quality control should be "Quality assured through controlled processes". Advantages are:

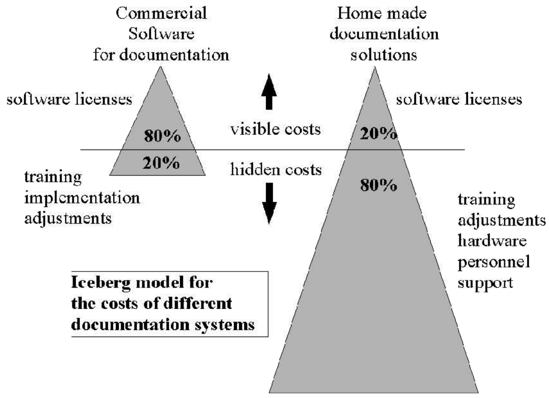
- better predictability
- better understanding
- better reference for process assessment
- a clear distinction between owners and users
- a resulting decrease in reject rates

Another aspect of documentation systems is the cost structure: the only real visible costs of a "home build" software system are the direct costs for software, which might even be an ordinary office-

---

[2] See also SmallTimes July/August 2003 p.43, Monteith G. Heaton and John S. Foster, "MEMS companies don't need foundries, they need full manufacturing partners"

suite. But there are a large amounts of hidden costs which are generated for implementation and adjustments, hardware, personnel, support and training. The visible costs are only a tip of the iceberg. Having a software package that implements the document system might have no change in the visible costs, but reduces the hidden costs considerably.



Requirements for software tools

Summarizing all points mentioned above, the next list of requirements can be generated for the next generation MST software tools

- The software should be able to **handle flexibility** in components (thus geometries), as well as in applying modeling tools to the geometries and in the application area in which the software can be used.
- The software should support **multi user cooperation and communication**. This means that more people should be able to work at the same time on the same system, but also several people with different level of abstraction should be able to work at the same design.
- **Documentation** is an integral part of the design process. This requires all technologies (thus also new and not known at this point) should be able to be incorporated into the system. Small amounts of non-accurate **measurements** during development, as well as high amounts of standard measurements with large data files during production, should be integrated in one software tool.
- All data should be **re-usable** to enable the organization to learn. One way of doing that is making the software environment parameterized.
- The software should be able to **link actual process data** to process flows to develop the design rules for a particular design.

Conclusion

There are a lot of similarities between integrated circuit electronics and micro system technology but the differences ensure that the software tools for the integrated circuit electronics cannot be applied effectively in the micro systems technology area. Next generation software tools for micro system technology should facilitate flexibility instead of fixation of technology. This is the model that makes the cost structure of micro system technology acceptable. The next generation software tool for micro system technology should facilitate communication to make the design house / foundry model work effectively.
Supplying these software tools to the MST community is the mission of PhoeniX.