Fault simulation and modelling of microelectromechanical systems

by R. Rosing, A. Lechner, A. Richardson and A. Dorey

High-reliability and safety-critical markets for microelectromechanical systems are driving new proposals for the integration of efficient built-in test and monitoring functions. The realisation of this technology will require support tools and validation methodologies including fault simulation and testability analysis and full closed-loop simulation techniques to ensure cost and quality targets. This article proposes methods to extend the capabilities of mixed signal and analogue integrated circuit fault simulation techniques to MEMS by including failure mode and effect analysis data and using behavioural modelling techniques compatible with electrical simulators.

Microelectromechanical systems (MEMS) find their applications mainly in smart sensing and actuation where low cost, high reliability, high accuracy, programmability and small size are critical specifications. Current and future application examples are pressure sensing in aircraft engines, automotive braking, steering and control, vessel pressure in reactors and medical implants. Typically these devices will have a non-electrical interface or input and limited controllability and accessibility, creating new challenges for production test, self-test and online data validation techniques. Similar motivations are currently driving the introduction of mixed signal fault simulation environments and modelling techniques.

The generation and validation of efficient test strategies for integrated circuits, whether for production or in-the-field monitoring utilises fault simulation techniques. This process supports the design and test engineers in the optimisation of test programmes for maximum fault coverage and identification of testability problems early in the design and manufacture cycle. To enable fault simulation, simulator compatible models of failure mechanisms and defects have to be generated. Prior to a fault simulation, a realistic fault list must be extracted from the design data at the layout or schematic level. The corresponding models of these faults are sequentially inserted into the fault simulation source file. Computation of the test coverage will then allow optimisation of testability and identification of potential sources of test escapes.

For microelectromechanical transducers, integration complexity, low cost and small size is stimulating new markets, many of which require high reliability and dependability. High-quality test procedures, both online and offline must be developed, supported by a test validation methodology. This article proposes methodologies to include microelectromechanical transducer elements within a fault simulation process to support efficient computation of test coverage for MEMS. Methods of modelling both the electromechanical and electrical elements within the same simulation environment to support efficient injection and analysis of faults are discussed. Behavioural modelling techniques compatible with electrical simulators are used to implement these models. In most cases, the fault simulation process must be carried out in close-loop configuration to allow the inclusion of non-idealities that can affect fault coverage, such as process variations, noise, mode coupling and resolution limitations. A list of realistic faults for the microelectromechanical transducers has to be generated and modelled to enable this process.

The proposed methodologies are based on preliminary work[1] and build on component level modelling used in design and test studies at both Robert Bosch GmbH[2,3] and Carnegie Mellon University.[4]

In the next section, a methodology for deriving a complete fault list for microelectromechanical transducers is described through combining failure modes obtained from a failure mode and effect analysis (FMEA) and defect-related and parametric faults extracted in a similar way as for hard and soft faults in mixed-signal circuits.[10] There follows a discussion of strategies for modelling both these faults and fault-free transducers at different levels of hierarchy. A possible built-in self-test technique for an example sensor is then described.

Previous work

The FMEA$^+$ concept

FMEA$^+$ was first proposed by Olbrich.[5] The technique relies on integrating qualitative failure analysis and quantitative fault simulation. FMEA[6] is well accepted by the system design industries whereas fault simulation tends to be restricted to low-level components (unless behavioural models are used). To illustrate the need for the integration of the two methods, a brief analysis of the causes of faults that can occur in MEMS devices reveals the following categories:

- local defects
- global and local parameters out of tolerance
- wear (especially in devices with movable parts)
- environmental hazards
- problems due to imperfection in the design process (i.e. design validation poor compared to mixed-signal designs)
- mode coupling/structure oscillation in incorrect modes
- system level faults (for example crosstalk between signals of different modules)

For CMOS circuits, defect-related and parametric faults are typically taken into account during a fault simulation and can be expanded to include in-field failures, such as gate oxide breakdown or open faults caused by electromigration. FMEA can be used to compile fault lists

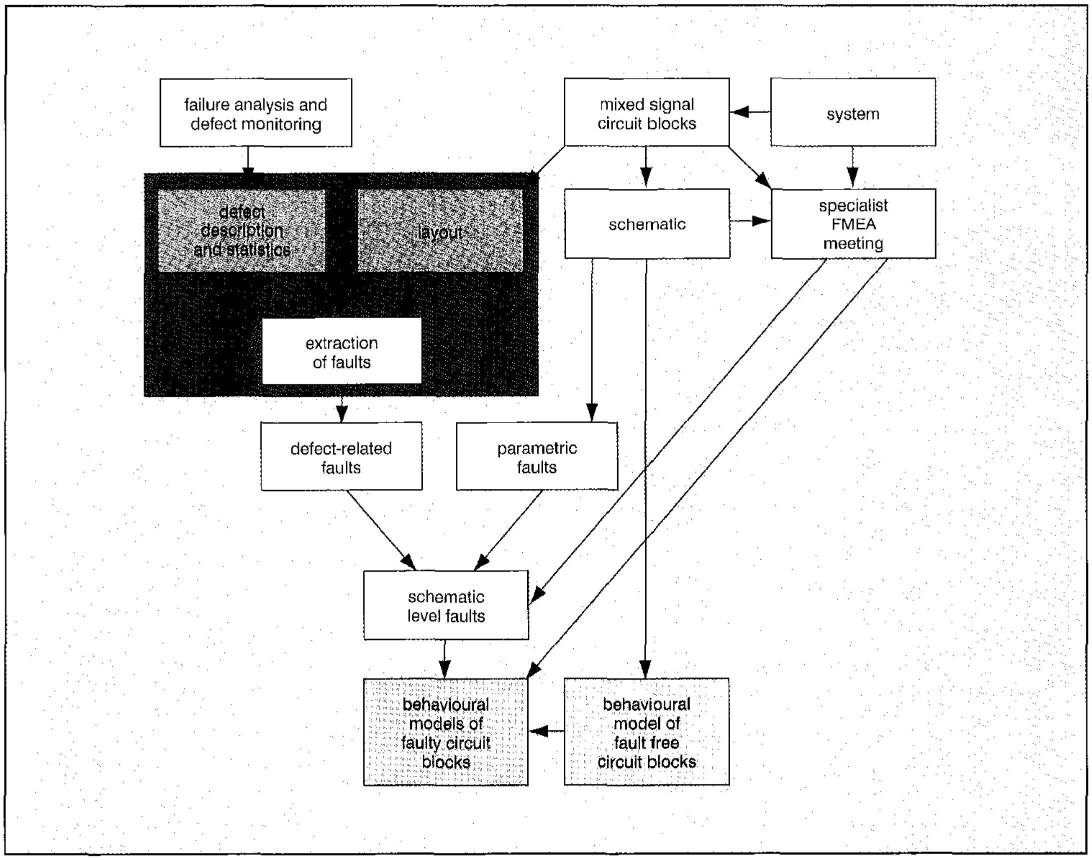


Fig. 1 Generation of the fault list and behavioural models for mixed-signal circuitry in the FMEA$^+$ approach

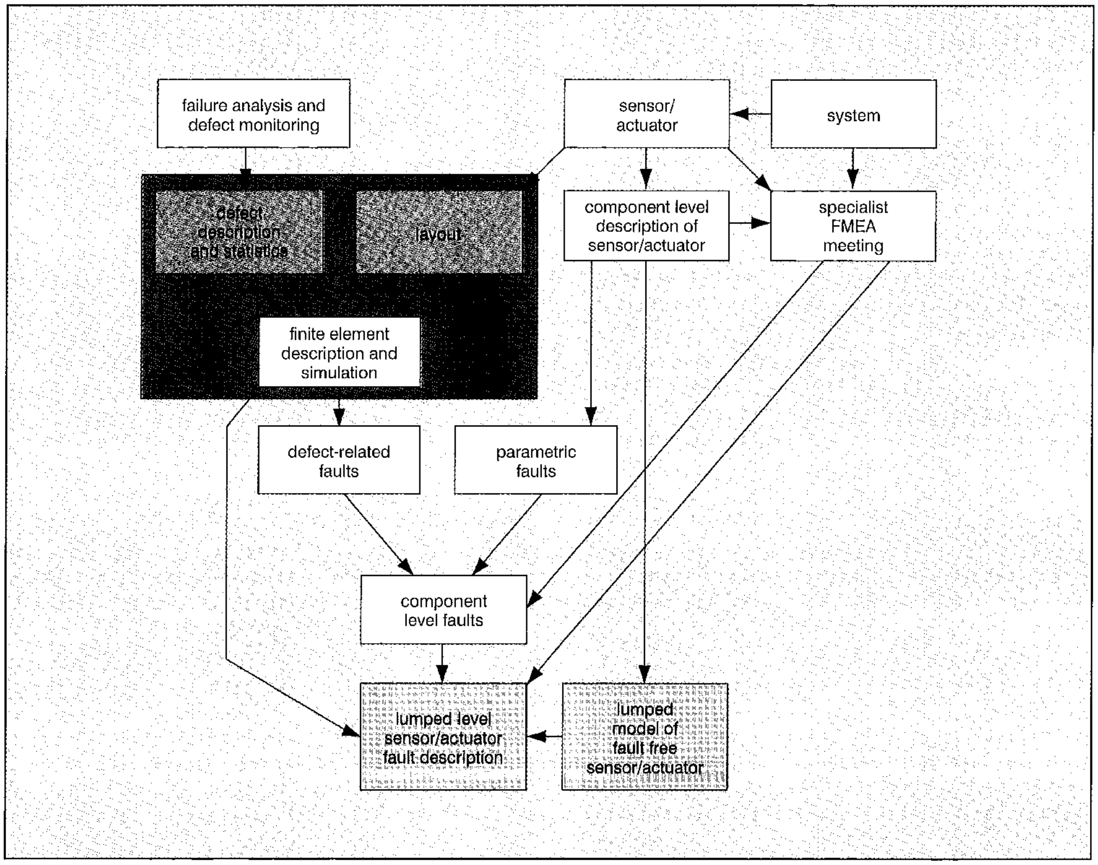


Fig. 2 Generation of the fault list and behavioural models for sensors/actuators in the FMEA⁺ approach

related to the remaining fault categories. The procedure involves:

1 Identification of all the functions of the system at all levels of hierarchy (component, macro and system) to be analysed.
2 Anticipation and description of how the parts at the different levels of hierarchy can fail (failure mode).
3 Assumption that failures have occurred and description of effect(s).
4 Identification of every possible fault cause for the failure mode.

In a failure mode, effect and criticality analysis (FMECA)[6,7] a ranking system is used to express the severity of the failure mode, its chance of occurrence and the likelihood of it being detected. FMECA will simply be denoted as FMEA throughout this article.

FMEA is ideally performed by a team of specialists involved in the design, test and fabrication of the system. Since the analysis is performed at different levels of hierarchy, failure modes can be predicted at an early stage of the design. It will, however, not be possible to predict all failure modes and their effects accurately in an FMEA meeting. Furthermore, the ranking of the different faults is based on the subjective judgment of the FMEA team members.

Another disadvantage of the FMEA method is that it is not automated. The analysis can be automated using numerical simulations, expert systems[8] or causal reasoning.[9] In all cases, information compiled from previous analysis of similar systems is used. This reduces the time to generate a list of failure modes. However, both expert systems and causal reasoning suffer from a subjective evaluation of the effects of the failure modes. Furthermore, the methods that are used to evaluate these effects are not compatible with the numerical method used in fault simulation.

Many of the disadvantages mentioned above can be overcome when the faults predicted during an FMEA meeting are modelled within a hierarchical fault simulation of the microsystem. This requires the modelling of these faults in a form compatible with the simulator.

Analogue and mixed-signal fault simulation

The methodology generally used to generate the fault list for analogue and mixed signal fault simulation is shown in Fig. 1. The following procedures are core to the success of the methodology:

1. Inductive fault analysis (IFA).[10,11] Here a defect-related weighted fault list is extracted from the layout, taking defect statistics like densities and defect diameter distributions into account.
2. Parametric faults due to process variations and device-oriented faults, i.e. gate oxide shorts, can be added to the fault set from the schematic level. The probability of gate oxide shorts can be predicted from the dimensions of the transistor in which it occurs. Layout information is not required in this case.
3. The fault set is typically simulated at the schematic level.
4. The use of behavioural models for fault free and classes of faulty circuit blocks will enable a more efficient hierarchical fault simulation of an entire system.

Sensor and actuator fault simulation

To generate a fault list and both fault-free and faulty behavioural models for a sensor or actuator, a similar procedure as that for mixed-signal circuits is feasible. This is shown in Fig. 2.

The sort of defects that can occur in these kinds of structures can be determined by failure analysis. In the work by Castellejo,[12] the most typical failure mechanisms in a bulk micromachining process are identified for each processing step together with the faults and deviating parameters caused by those mechanisms. Kolpekwar[13] on the other hand investigates the effect of one category of defects, particulate contamination, by inserting them into a mesh description of the structure and performing finite-element simulations using a Monte Carlo approach. For each fault, the change in resonant frequency of an example resonator is observed. Since this is not a parameter of either the lumped model or the component level model, the effect on the closed-loop system behaviour can not be simulated.

To be able to handle a fault in a closed-loop system simulation, it has to be modelled at either the component level or the lumped level (it is discussed below how this can be achieved). Previous work[14] by the authors shows the results of a fault-free and a fault simulation (for an example fault) on the closed-loop system are shown. The sensor was modelled at the lumped level and the electronic circuit blocks at the behavioural level.

The results from the fault generation process shown in Figs. 1 and 2 can be used to perform a hierarchical and statistical fault simulation on the entire microsystem. The results of this simulation identify difficult-to-test or detect faults and test escapes, and can also be used to apply design-for-test (DfT) optimisations or to implement built-in self-test (BIST) structures. This is shown in Fig. 3*a*. In Fig. 3*b*, hierarchical simulation of the system block is achieved by describing faults at either the component level or through faulty behavioural models. This system block can be either an electrical circuit block or a sensor/actuator.

Achieving test support using FMEA[+]

To be able to include failure modes from an FMEA into a hierarchical fault simulation of the microsystem, a simulator compatible fault model is needed. A behavioural or schematic level description of the transducer can be implemented in an electrical simulator, which supports a behavioural language including the

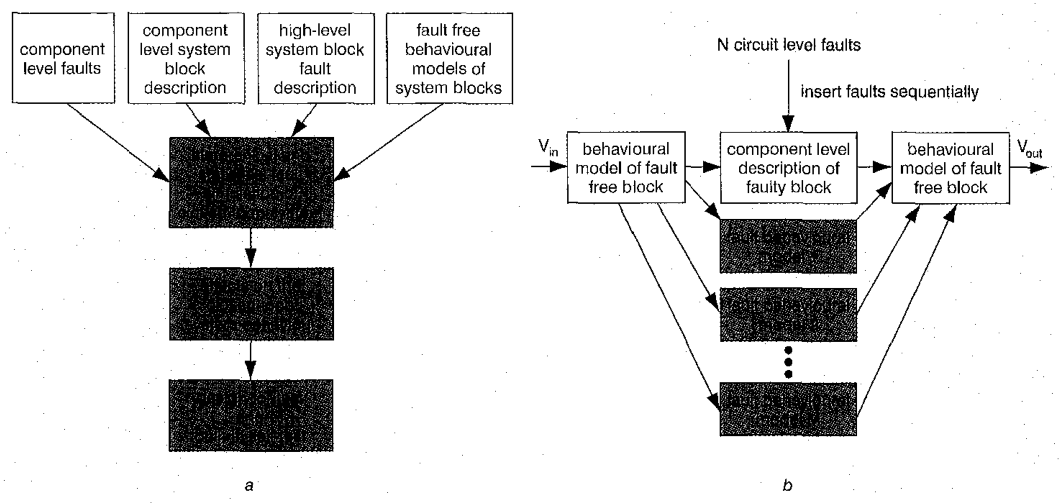
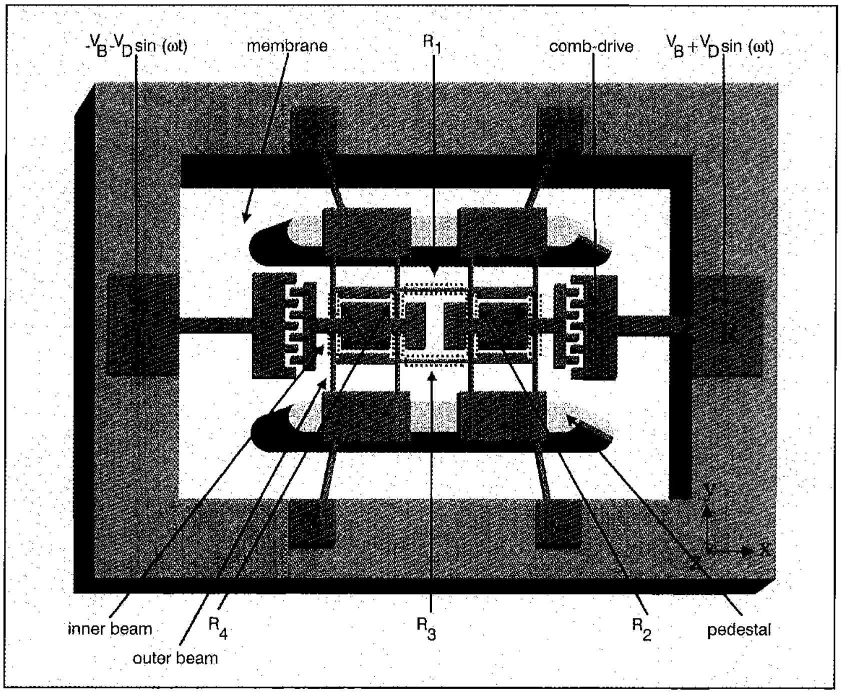


Fig. 3 Achieving DfT or BIST for microsystems

Fig. 4 Pressure sensor



use of non-electrical variables. This enables the combined simulation of electrical circuitry and the transducer, and therefore microsystem fault simulation. Examples of programs that enable these simulations are the combination of ELDO and VHDL-AMS, Saber with its behavioural language MAST, and SMASH with its behavioural languages ABCD and VHDL-AMS.[15] Examples of tools that support hierarchical fault simulation are MiST PROFIT,[16] Faultmaxx[17] and GDSFaultsim.[18] Some of these tools combine hierarchical and statistical mixed signal fault simulation.

Modelling of transducers can occur at two levels of hierarchy, as will be shown in the next two subsections. Models are preferably generated using analytical solutions. The results of finite-element simulation can be used either to refine the analytical models or to generate models for elements that do not have an analytical solution.

Component level modelling

The description of sensors and actuators at a component level is based on the fact that these structures can be described as an interconnected set of elements. A microelectromechanical structure can, for example, be described as a composition of elements such as masses and beams. When behavioural descriptions are generated for those elements, the structure can be modelled at a schematic level. This is similar to an electrical schematic, which consists of a number of electrical components, for which behavioural descriptions (for example the equations modelling a transistor) are derived.

To form a component level description of the entire transducer, the behavioural models of the separate components have to be linked using 'through' and 'across' variables. In the electrical domain these variables correspond to currents and voltages, respectively. Kirchhoff's current law states that, on each node of the schematic, the sum of currents equals zero. In a closed-loop within the schematic the sum of voltages equals zero (Kirchhoff's voltage law). These relationships hold in general for the corresponding through and across variables. For mechanical structures the through variables are the forces and moments and the across variables are the displacements and rotations.

Use of through and across variables is supported in a number of behavioural languages, used within circuit simulators. In the work by Vandemeer[19] a simulator, referred to as a nodal simulator, which links descriptions of MEMS components together, is described. Every beam of the design is described as one element characterised by its stiffness matrix and an effective mass matrix. The simulator can therefore be regarded as a finite element simulator in which every component is modelled by just one element.

This type of (linear) simulation will give good results for small displacements. Furthermore, on each component, the relationships between forces and displacements in the different directions should be approximately independent. If these demands do not hold, either non-linear analytical solutions have to be used or the results from low-level finite-element simulations have to be mapped into the behavioural model.

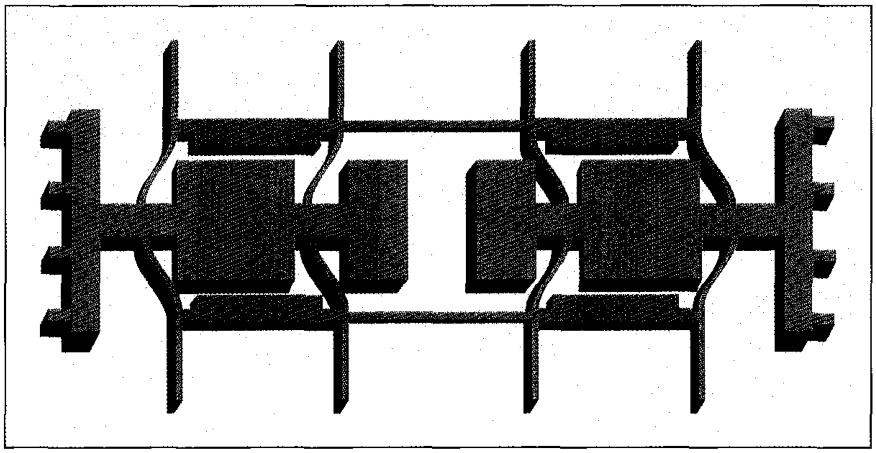


Fig. 5 Movement of the sensor

Modelling of beam elements is described by Neul.[20] The natural frequencies of the beams are simulated by splitting the beam into more than one element. In the work by Teegarden[21] a microgyroscope system is modelled. Finite-element simulations are used to derive descriptions of components for which it is difficult to find an analytical solution.

In Fig. 3 it is shown that both the component level description and the behavioural description of a transducer are required in a hierarchical fault simulation. A further explanation for the need to use both approaches is given in the following subsection using an example system.

Lumped modelling

In Fig. 4 a sketch of an industrial micromachined resonant pressure sensor is shown. Electrostatic forces within the comb-drives cause the two movable structures to oscillate in opposite directions. The structures therefore separate and then close. Due to the stiffness of the piezoresistors (R_1 and R_3) connecting the two movable parts, the movement of the outer beams is negligible compared to the movement of the inner beams. An exaggerated illustration of the movement of the sensor is shown in Fig. 5.

A glass cover over the sensor (not shown) maintains an approximate vacuum, which minimises the viscous drag and therefore maximises the Q-factor of the system. The pressure difference between the upper side of the substrate, exposed to the low pressure in the cover, and the bottom of the substrate (forming part of the bottom of the chip), exposed to the measurement pressure, causes the substrate to bend. Therefore the pedestals move apart and cause a tension in the beams that form the spring. This tension causes the spring stiffness and therefore the resonant frequency of the system to change.

A high-level description of this sensor is generated in the behavioural language ABCD within the simulator 'SMASH'. The sensor is implemented as a set of equations in a subcircuit of a system netlist. The movable part of the structure is simply described by its mass. The four beams connected to each movable part are modelled as one spring. An analytical solution is used to model the relationship between the stiffness of the beams (spring

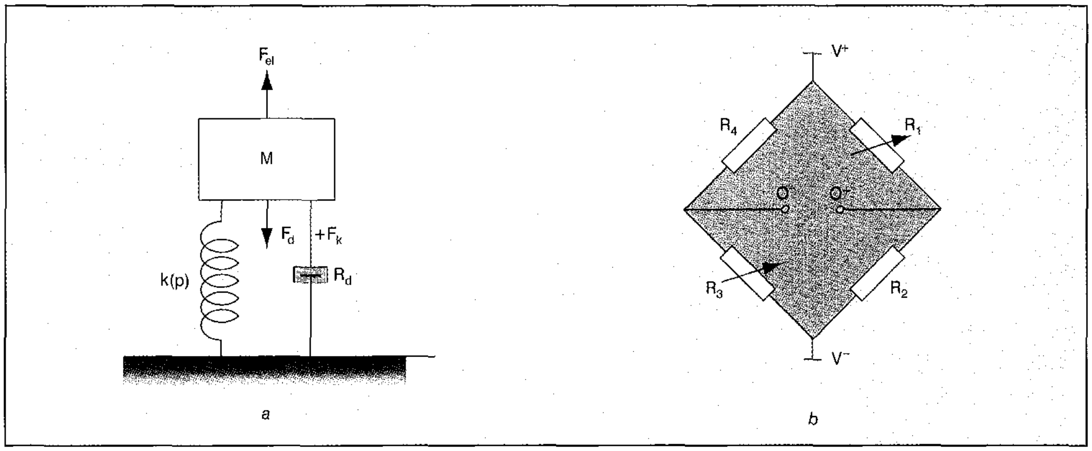


Fig. 6 Behavioural modelling of the sensor

'constant') and the tension in the structure. An equivalent mass of the beams is calculated and added to the mass of the movable part. The mechanical behaviour of the transducer can therefore be modelled as a mass-spring-damper system at the lumped level, as illustrated in Fig. 6*a*.

Analytical solutions exist for bending of membranes under different boundary conditions. However, the fact that, in this sensor, pedestals are fixed on the membrane makes it very difficult (if not impossible) to calculate the bending of the membrane and the resulting distance that the pedestals separate. Therefore, an analytical relationship between the measurement pressure and the tension in the beams forming the suspension cannot be derived. In this case, finite-element simulations were used to determine the relationship between the measurement pressure and the separation of the pedestals. The relationship is found by way of performing a best polynomial fit through a set of finite-element results for different pressures. An analytical solution was used for the relationship between this separation and the tension in the beams forming the suspension.

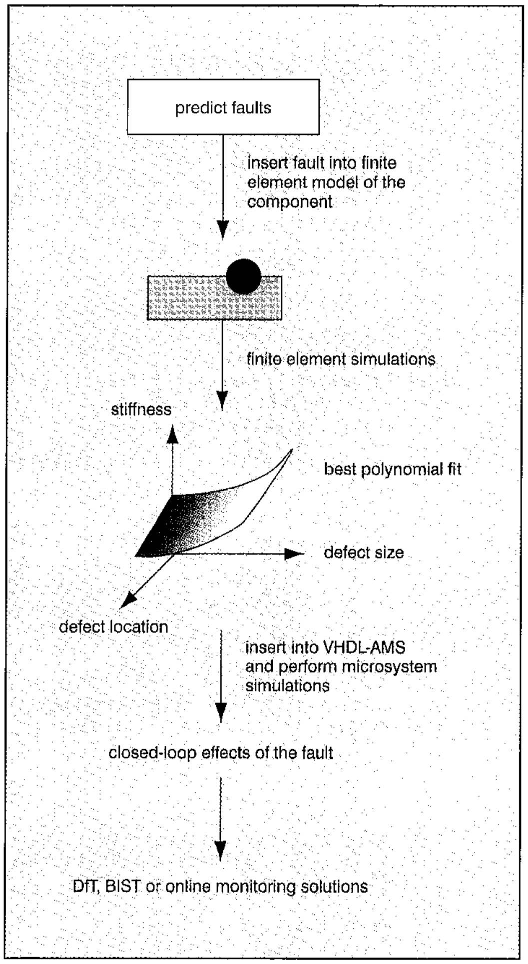


Fig. 7 Modelling and simulation of defect-induced faults

An analytical solution is used to model the relationship between the applied drive voltage and the electrostatic force. The electrical behaviour of the comb-drive has been modelled as a varying capacitance under influence of movement.

The piezoresistors R_4 and R_2 (both consisting of four resistors) will not change value during movement, since the effect of part of each resistor being compressed is cancelled out by the other part being extended. The piezoresistors R_1 and R_3 change value at a frequency equal to the resonant frequency of the structure. The applied force to and the relative change of resistance value of these piezoresistors are related by the piezoresistance coefficient π. From Fig. 4 it can be seen that the piezoresistors form a Wheatstone bridge, as is schematically modelled in Fig. 6*b*. The output *V(O+; O-)* of this bridge is the sum of a sine wave voltage at the resonant frequency and a bias voltage.

This example illustrates that lumped modelling of transducers is similar to behavioural modelling of electrical circuit blocks. In both cases only the functionality of the circuit is described. The elements of which the circuit is constructed are not individually modelled (all the beams in the sensor are modelled as one stiffness value).

Fault modelling

As was explained previously, to enable a hierarchical fault simulation, sensor fault models must be compatible with the simulator. Faults that influence the behaviour of the entire sensor have to be modelled at the behavioural level. By adapting the equations in the nominal behavioural model, a faulty behavioural model can be generated.

Faults affecting a single element result in asymmetric behaviour and need to be modelled at the component level. At this level, every component is described as a parameterised cell. Parametric faults can be described by adapting the behavioural description of the component.

Defect-related faults can be modelled by either adapting the behavioural description of the component or adding an extra component. Finite-element simulations can be used to quantify the defects effect on the functionality of the component in which it occurs. To implement this, a list of defects must be generated. This can be done by examination of the process and defect statistics. The strategy to extract, model and simulate this type of failure is shown in Fig. 7. For every type of component this series of simulations is performed once, after which a fault model for this component can be generated.

The above process will create a library of component level fault models, containing both defect-induced and parametric faults, which can be generated and reused in other designs. Furthermore, as with gate oxide shorts in electrical circuitry, the likelihood of occurrence of each fault can be predicted from the dimensions of the structure the fault is located in.

Fault modelling at the lumped level occurs in a similar fashion, by changing the parameters of the lumped model, adapting the behavioural description of part of the model or adding an extra element to the lumped model.

The previously described model can be adapted to generate a component level model. This level of simulation is possible using SMASH.

To enable modelling of FMEA failure modes, it is necessary to categorise these failures to the level of modelling they require. The following categorisation is proposed:

- Component failures, such as elements affected by environmental effects, wear or other device-oriented faults, as mentioned earlier in the article.
- Failures that can be easily modelled at the lumped level. Modelling at this higher level of hierarchy will speed up fault simulations.

Following these steps, a thorough investigation of fault effects on the closed-loop system behaviour can be achieved, which paves the way for structural DfT, BIST or online test solutions.

DfT and BIST approaches for MEMS

One built-in test approach for the MEMS structure illustrated in Fig. 4 is currently under investigation. The aim is to adopt the integrated diagnostic reconfiguration (IDR)[22] method initially developed for electrical components for use in the pressure sensor.

An electrical circuit design using the IDR technique is composed of a number of interchangeable circuit blocks. During normal operation these circuit blocks can be dynamically interchanged, though the overall circuit function stays the same. When a fault affects one block the circuit function changes when its blocks are interchanged and can be easily identified at the system level. By observing the profile of changes, faulty components can be identified and exchanged leading to a degraded circuit function.

It can be shown that the technique has specific applications to devices where an array of identical sensors is implemented. For the pressure sensor, which contains internal redundancy, the technique is currently under evaluation as a built-in test methodology for production and for in-field verification. During the test program, the sensor test stimulus will be fed to one comb drive only (L in Fig. 4) and the output $V(O-;O+)$ (Fig. 6*b*) will be monitored. In the following stage the test stimulus will be fed to the alternative comb drive (R in Fig. 4) and its output will be compared to the initial system response. A relative measurement is therefore taken. An absolute measurement (simply measuring the resonant frequency) contains less information due to the unknown input pressure. The aim is to extend the test methodology to allow a dynamic exchange of comb structures to ease the test response analysis. It is expected that the technique will support full on-chip self-test, provided that test stimulus generation and response analysis can be implemented in the interface electronics.

Conclusions and future work

This article has proposed the combination of failure mode and effect analysis (FMEA) and hierarchical fault simulation techniques for testability computation and optimisation of MEMS. This strategy has been referred to as FMEA$^+$. An analysis on a pressure sensing system has been initiated through the use of behavioural models of the sensor components including electrical feedback circuitry.

The new technique can be automated by classification of FMEA failures supported by a fault simulation model library and easily integrated into commercialised simulation environments.

Further research addresses a deeper investigation into MEMS failure modes (or classes) and hierarchical behavioural fault-free and fault modelling techniques. Furthermore, the component level description of the pressure sensor and fault models at both the lumped and the component level will be generated.

A possible BIST solution, which uses reconfiguration techniques to exploit the redundancy within an industrial pressure sensor is proposed in this article. Further studies will pave the way for the successful DfT optimisation and integration of on-chip test support and BIST for this and other types of MEMS.

References

1 ROSING, R., RICHARDSON, A., DOREY, A., and PEYTON, A.: 'Test support strategies for MEMS', Proceedings of the 5th International Mixed Signal Testing Workshop, Whistler, Canada, 1999, pp.345-350

2 TEEGARDEN, D., LORENZ, G., and NEUL, R.: 'How to model and simulate microgyroscope systems', *IEEE Spectrum*, July 1998, pp.66-75

3 NEUL, R., BECKER, U., LORENZ, G., SCWHARZ, P., HAASE, J., and WUNSCHE, S.: 'A modelling approach to include mechanical components into the system simulation', Proceedings of DATE, 1998, pp.510-517

4 MUKHERJEE, T., FEDDER, G. K., and BLANTON, R. D.: 'Hierarchical design and test of integrated microsystems', *IEEE Design & Test of Computers*, 1999, **16**, (4), pp.18-27

5 OLBRICH, T.: 'Design-for-test and built-in self-test for integrated systems', PhD Thesis, Lancaster University, Lancaster, UK, September 1996

6 British Standard Institution, BS5760: 'Reliability of systems, equipment and components', part 5. 'Guide to failure modes, effects and criticality analysis (FMEA and FMECA)', 1991

7 BOWLES, J. B.: 'The new SAE FMECA standard', Proceedings of the Annual Reliability and Maintainability Symposium, 1998, pp.48-53

8 RUSSOMANNO, D. J., BONNELL, R. D., and BOWLES, J. B.: 'A blackboard model of an expert system for failure mode and effects analysis', Proceedings of the Annual Reliability and Maintainability Symposium, 1992, pp.483-490

9 BELL, D., COX, L., JACKSON, S., and SCHAEFER, P.: 'Using causal reasoning for automated failure modes and effects analysis (FMEA)', Proceedings of the Annual Reliability and Maintainability Symposium, 1992, pp.343-353

10 OLBRICH, T., PEREZ, J., GROUT, I., RICHARDSON, A. M. D., FERRER, C.: 'Defect-oriented vs schematic level based fault simulation for mixed-signal ICs', Proceedings of ITC, 1996, pp.511-520

11 WALKER, H., and DIRECTOR, S. W.: 'VLASIC: a catastrophic fault

yield simulator for integrated circuits', *IEEE Transactions on Computer-Aided Design*, 1986, **CAD-5** (4), pp.541-556

12 CASTILLEJO, A., VEYCHARD, D., MIR, S., KARAM, J. M., and COURTOIS, B.: 'Failure mechanisms and fault classes for CMOS-compatible microelectromechanical systems', Proceedings of ITC, 1998, pp.541-550

13 KOLPEKWAR, A., KELLEN, C., and BLANTON, R. D.: 'MEMS fault model generation using CARAMEL', Proceedings of ITC, 1998, pp.557-566

14 ROSING, R., RICHARDSON, A. M., and DOREY, A. P.: 'A fault simulation methodology for MEMS', Proceedings of the International Conference on Design Automation and Test in Europe, Paris, 27th-30th March 2000, pp.476-483

15 SMASH Reference Manual, Dolphin Integration, Grenoble, France, 1997

16 VOORAKARANAM, R., CHAKRABARTI, S., HOU, J., GOMES, A., CHERUBAL, S., CHATTERJEE, A., and KAO, W.: 'Hierarchical specification-driven analog fault modeling for efficient fault simulation and diagnosis', Proceedings of ITC, 1997, pp.903-912

17 HAMIDA, N. B., SAAB, K., MARCHE, D., and KAMINSKA, B.: 'FaultMaxx: a perturbation based fault modeling and simulation for mixed-signal circuits', Asian Test Symposium (ATS '97), November 1997

18 EBEN AIMINIE, Y., RICHARDSON, A., DESCLEVES, C., and SOMMACAL, K.: 'GDS FaultSim-a mixed-signal IC computer-aided-test (CAT) tool', Proceedings of DATE, Munich, March 1999

19 VANDEMEER, J. E.: 'Nodal design of actuators and sensors (NODAS)', MSc Thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, May 1998

20 NEUL, R., BECKER, U., LORENZ, G., SCHWARZ, P., HAASE, J., and WUNSCHE, S.: 'A modelling approach to include mechanical components into the system simulation', Proceedings of DATE, 1998, pp. 510-517

21 TEEGARDEN, D., LORENZ, G., and NEUL, R.: 'How to model and simulate microgyroscope systems', *IEEE Spectrum*, July 1998, pp.66-75

22 SHARIF, E., RICHARDSON, A., and DOREY, A.: 'An integrated diagnostic reconfiguration (IDR) for fault tolerant mixed signal microsystems', Proceedings of the International Conference on Electronics, Circuits and Systems, 1998, **1**, pp.413-416

The authors are with the Centre for Microsystems Engineering, Lancaster University, Lancaster LA1 4YR, UK.

BOOK REVIEW

Computer architecture and implementation

Harvey G. Cragon

Cambridge University Press 2000, 318pp., £24·95 ISBN 0 52165 168 9

If Goethe is justified in his metaphor of architecture as 'frozen music', modern computer architecture should be described as frozen cacophony. Indeed, despite the synchronous nature of digital logic, different units inside the computer follow different 'rhythms' (e.g. memory is more than an order of magnitude slower than the processor). Despite the sequential nature of computer programs, operations are often performed out of order or in parallel, or even speculatively, in anticipation of certain events that may or may not occur later. A computer is, as it were, a large orchestra, which does not quite follow its conductor, the master clock, nor its scores, the machine code. It frequently stalls and replays its 'music' out of sync, with many additional voices or none at all.

The reasons for such complex behaviour are not always fundamental. Many factors are historical, stemming from the desire to preserve compatibility with earlier hardware products, existing software or both. Consequently, to make sense of the intricate methodologies and design trade-offs that are laid in the foundations of modern computer organisation, one needs some historical perspective.

Here Harvey Cragon's book 'Computer architecture and implementation' comes in handy, as the author is not forgetful of the works of the past that continue to influence the technology of the present. His analysis goes back to the first widely used commercial mainframes, IBM System 360-390. At the same time, the state-of-the-art architecture of Intel x86 up to the Pentium Pro with MMX extensions is employed as a contrasting example. These two reference points delimit almost 40 years of design evolution, which is pretty much the whole modern history of computer engineering.

Perhaps more importantly, the book introduces performance analyses wherever it touches upon structures with vague temporal behaviour, such as pipelines, hierarchical memory or speculative execution. Those help to make the presentation in the book quantitative and concrete, showing the strengths and weaknesses of any design trade-offs. The author rightly believes that the ability to make quantitative predictions of product behaviour is a hallmark of good engineering as well as being an effective pedagogical device.

The book can serve as a simple introduction to modern computer organisation. As such it is of considerable value to electrical engineers wishing to understand the innards of computers well enough to be able to read specialist literature on the subject.

PROF. ALEX SHAFARENKO
University of Hertfordshire