Layout Verification and Correction of CMOS-MEMS Layouts

Bikram Baidya (bbaidya@ece.cmu.edu), Kai He (khe@ece.cmu.edu)
and Tamal Mukherjee (tamal@ece.cmu.edu)
Carnegie Mellon University, Pittsburgh, PA 15213, USA

ABSTRACT

The advent of CMOS micromachining has introduced new design rules for fabrication of integrated CMOS-MEMS devices. This paper presents a context dependent DRC algorithm to handle the issues related to pre-fabrication verification of such layouts. In addition, problems related to density control, specific to CMOS-MEMS designs, are discussed. An automatic slotter which introduces MEMS-compatible slot holes is presented and its capability demonstrated. Having such verification and correction tools which address the needs of integrated CMOS-MEMS designs will help reduce integrated MEMS design time.

Keywords: DRC, slotting, CMOS micromachining, microloading, slot hole shields, density requirement

INTRODUCTION

Layout verification in the form of Design Rule Checking (DRC) has long been known in VLSI [1] and more recently in MEMS [2]. With the advent of integrated CMOS-MEMS [3], new issues related to CMOS micromachining have surfaced necessitating the need for new DRC algorithms. The design rules for MEMS areas of such layouts differ significantly from those of the circuit areas. Hence the DRC algorithms for integrated designs need to intelligently apply design rules to different parts of the layout by automatically differentiating the MEMS and circuit areas. The use of the top metal layer to define MEMS structures and protect circuit areas in CMOS micromachining [3] results in large metal areas which need slotting to minimize fabrication abnormalities [5]. Unlike slotting in VLSI, the slot holes introduced in such integrated designs need to be properly shielded by underlying metal layers to ensure that the circuits are protected from the micromachining.

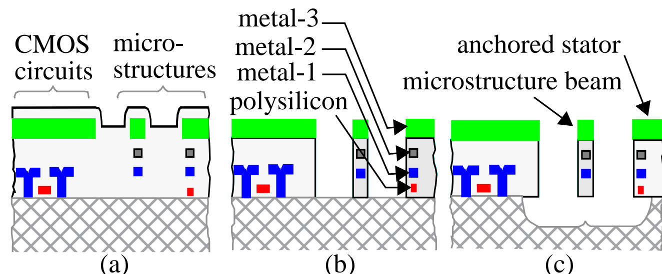


Figure 1: Cross-section of CMOS micromachining process [1]; (a) after standard CMOS process, (b) after anisotropic etch, (c) on final release using isotropic etch

The prototype DRC and slotting algorithms discussed here are designed for the high-aspect-ratio CMOS micromachining process developed at Carnegie Mellon University [3]. The process flow is shown in Figure 1. The standard CMOS process is followed by two etching steps to release the microstructures.

MEMS-SPECIFIC DESIGN RULES

Varying mechanical and electrostatic requirements in MEMS result in a wide range of structure widths and inter-structure gaps which are etched differently, due to microloading [6] [7] [8]. It was observed that the lateral etch depth was considerably lower for holes in plate areas compared to beam-beam gaps in the layout. A graph of undercut vs. inter-structure space is shown in Figure 2(a). Figure 2(b) shows a beam which could not be released because of small inter-beam gap. Similarly, the plate in Figure 2(c) could not be released because of the small etch holes while Figure 2(d) shows a plate that was released.

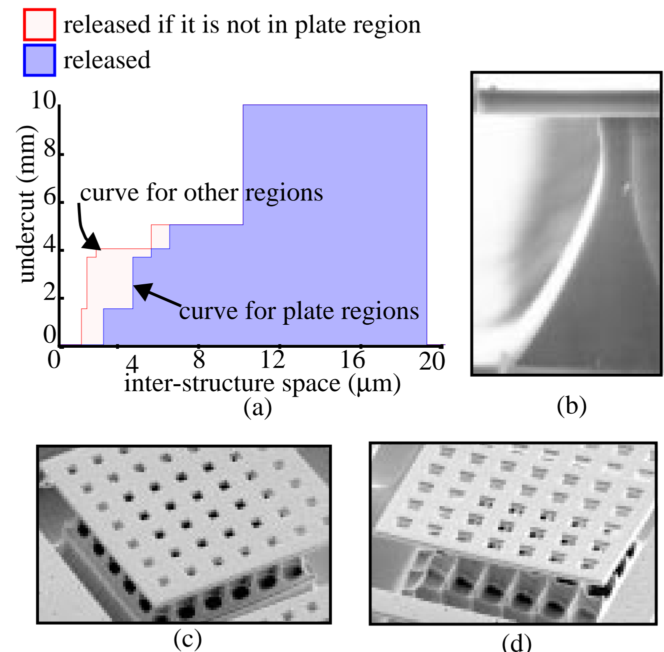


Figure 2: (a) Graph showing undercut widths for various inter-structure spaces, (b) beam not released due to small spacing, (c) plate not released because of small etch holes, (d) released plate

The graph in Figure 2(a) captures the context-dependent design rules (Figure 3) for maximum structural width (A) and minimum inter-structure gap (B). The minimum structural width (A) arises from the fact that ion milling erodes the top metal mask and effectively increases the variation in the over etch of the beam. The minimum structural metal extension rule (C) is needed to guarantee proper metal cover over the regions to be protected from etching. This accounts for the mask alignment errors. A minimum spacing to edge (D) is needed for the polysilicon in the microstructures to prevent it from being exposed to the etchants. A minimum circuits to edge spacing rule (E) is needed to protect the circuits from being etched away due to lateral etching. A maximum beam length rule (F) helps reduce the effects of curl [9] on the microstructures.

The most challenging aspect of design rule checking for MEMS is its context dependent nature. While strict conservative rules can be used to avoid this dependence, it would seriously limit the freedom of the MEMS designer. The prototype DRC algorithm discussed here liberates the designer to optimize the desired structure at the cost of algorithmic complexity of the design rule checker.

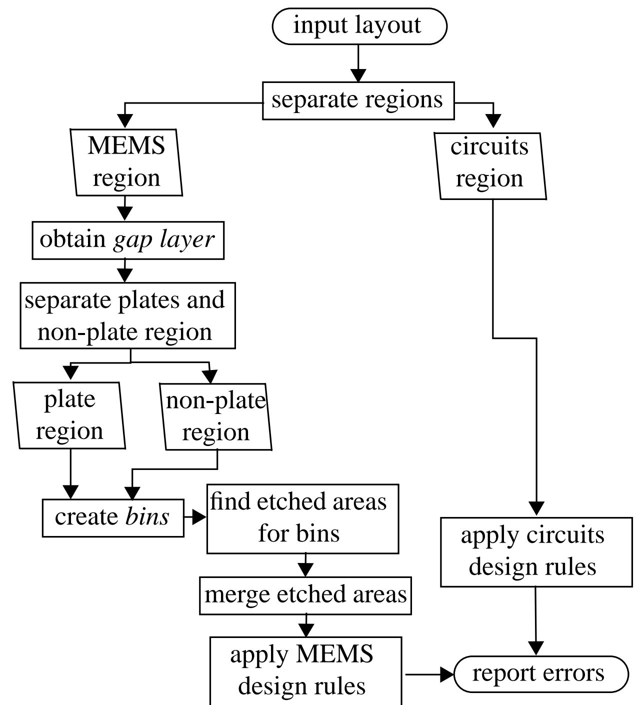
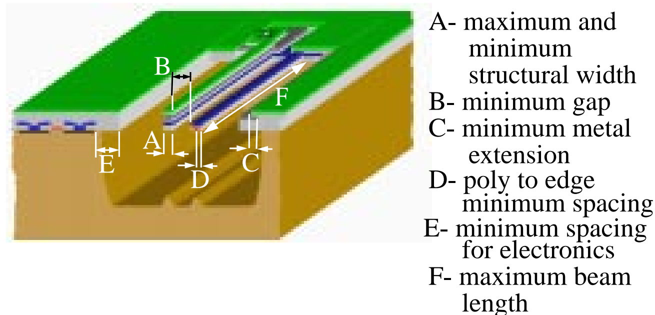


Figure 4: Flow diagram for DRC algorithm

DRC ALGORITHM

The release etch is dependent on the size, shape and spatial distribution of the gaps. Though ideally the etching curve in Figure 2(a) should be continuous, we model it using step functions. Implementation of such step DRC rules are common in VLSI where rules differ for narrow and wide metal areas. The geometry is divided into different areas depending on the step curve, using *shrink* and *bloat* operations, and each partition is saved into a separate derived layer (for e.g. the non-plate regions for the discussed process can be partitioned into *small* for width < 1.2 µm, *medium* for width between 1.2 µm and 1.5 µm and so on. There are a total of six partitions as shown in Figure 2(a)). The derived layers are then checked using the parameter corresponding to its size on the step curve.

MEMS DRC has a further complication in having two step curves which apply to the same layout. The choice of the curve depends on whether the empty regions being checked lie in a plate region or a non-plate region. The DRC algorithm first separates out the MEMS area from the circuit area of the layout using the maximum etch criterion defined by the etching curve in Figure 2(a) (20 µm for the process discussed). It then generates a *gap layer* by a geometric *not* of the MEMS area. This layer stores the area in which the etchant will release the MEMS structures. The algorithm emulates the etching process by expanding the geometry of this layer in accordance to the curves in Figure 2(a). The layer is partitioned into plate and non-plate areas using feature recognition like that in the extractor described in [10] [11]. The heuristics takes into account the geometrical parameters of the gaps and those of the neighboring filled areas. The two areas are further subdivided into regions (referred to as bins) corresponding to the different steps in the graph in Figure 2(a). The geometry in each of these bins is then expanded to obtain the area that the etchant entering it will release. The total released area is then obtained by a geometric *or* of all such areas and is compared with the original layout to verify the MEMS related design rules. The flow of the DRC algorithm is shown in Figure 4. The DRC algorithm uses generic geometry manipulation procedures like geometric *and*, *or*, *shrink*, *bloat*, etc., which are available in any commercial VLSI-CAD tool. Since the complexity of such procedures are always $O(nlgn)$, the total time complexity of the DRC algorithm is also $O(nlgn)$, where n is the size of the geometry being considered.



A- maximum and minimum structural width
B- minimum gap
C- minimum metal extension
D- poly to edge minimum spacing
E- minimum spacing for electronics
F- maximum beam length

Figure 3: Different design rules for CMOS micromachining

SLOTTING

Deep submicron CMOS processes require slotting of the metal layers [5]. This impacts the use of top metal layer as a

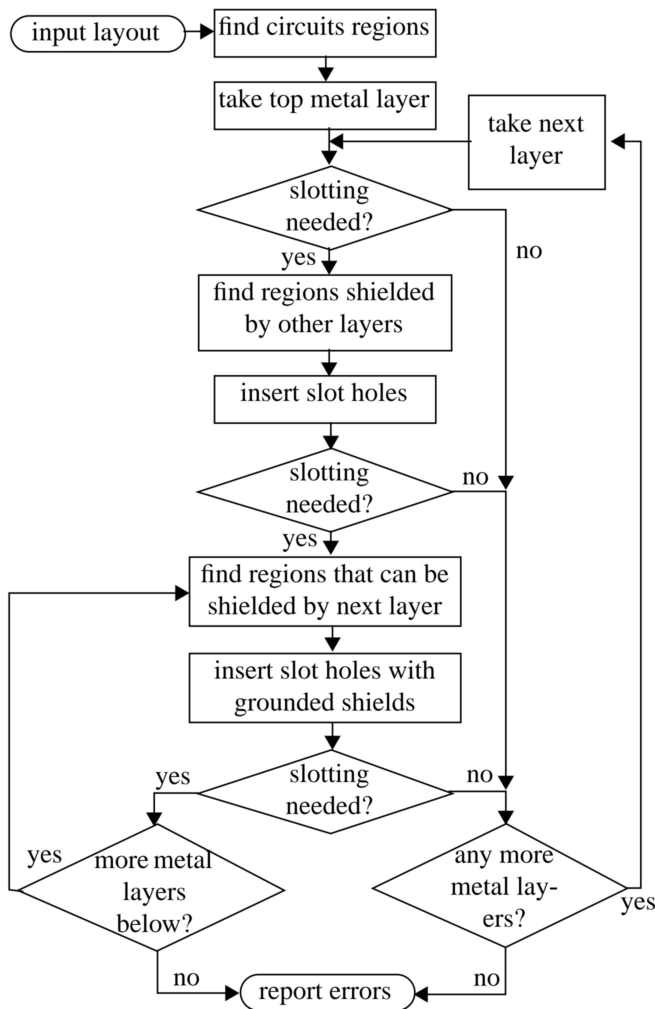


Figure 5: Flow diagram for slotting algorithm

shield for the circuits in the CMOS-MEMS process [3]. A greedy automatic slotter has been developed which reads in the DRC rules and creates a minimal number of DRC-correct slot holes to satisfy the density requirements of the process. Metal shields are inserted under the slot holes in the top metal layer to ensure that any circuitry under the slot remains protected from the micromachining. The shields are also electrically connected to the top metal layer by vias to prevent charge trapping.

The slotting algorithm uses predetermined grid to insert slot holes. The size of the grid is defined by the density requirement for the process. Hence the algorithm attempts to insert slots whenever any metal area sufficiently overlaps a grid element. The minimum area that is needed for overlap will depend on the type of slot hole the slotter is trying to implement. If the overlap area is rectangular in size then the slot hole together with the under-metal shields, if necessary, are inserted in the center of the overlap area. If the area is polygonal in shape, a finer grid, which divides the original grid into four equal subparts. Next, the overlap area with the finer grid is obtained and if sufficient overlap is present then a slot hole is inserted in the center. The density parameter and other design rule parameters required by the slotter are read in from an user-defined technology file. Each of the metal layers are slotted in a different loop starting from the topmost metal layer (metal3). In the process of slotting one layer, geometry is added to the underlying metal layers in the form of slot-shields. The shields inserted are shorted to the top ground metal layer in order to avoid floating metal areas. This helps in accurate estimation of the extra parasitic capacitances added in the process of slotting.

For the metal layer being considered (say metal3), the slotter first recognizes the potential areas for slotting by using generic procedures like *shrink*, *bloat*, etc. It then tries to insert slot holes in areas which are already covered by underlying metal layers. A density analysis is now used to find the remaining areas that still need slotting. DRC separation rules are then used to find the areas which can be slotted such that a slot-shield can be added using the next metal layer. Since the shield added needs to be connected to the top metal layer, there should be enough area to insert the via on the metal shield. In the final iteration, slot holes with shields using the last (metal1) layer are inserted. Such shields need to be connected to the top metal layer through the middle metal2 layer and hence sufficient overlap with this shield and the other metal layers are needed to insert connecting vias. If such overlap area is not available, the slotter tries to create the overlapping metal2 areas. Finally the areas that the slotter fails to slot are reported as DRC errors. These errors can be easily fixed by increasing the separation between the transistors in the circuit and rerunning the slotter. The overall flow for the algorithm is shown in Figure 5.

RESULTS

Example demonstrating the DRC algorithm

The context dependent nature of the DRC algorithm is demonstrated in the example shown in Figure 6. The errors are reported by the black lines. The DRC algorithm reports unreleased plate area ((a) in Figure 6) because of small hole size.

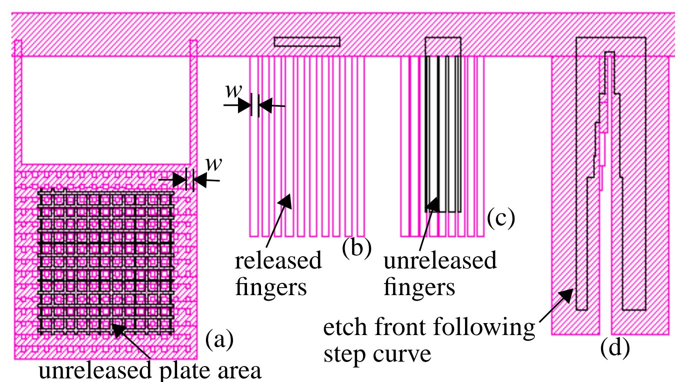


Figure 6: Example demonstrating MEMS DRC cases

However, it correctly predicts release of the fingers ((b) in Figure 6) having gaps of same width as the holes in the plate (*w*). The second set of fingers ((c) in Figure 6) were reported to have DRC errors because the gaps had a width less than the minimum required gap size (1.2 µm). The structure (d) in Figure 6 shows the step nature of the etch front. The amount of etching predicted by the DRC algorithm is dependent on the gap size in accordance to the graph in Figure 2(a).

Example demonstrating the slotter

Figure 7(a) shows a layout of a buffer circuit. The corresponding slotted layout is shown in Figure 7(b). Three cases of slot holes are used. First slot holes are added where other metal layers exist to act as shields (Figure 7(c)). Next, slot holes with metal2 under shields are added (Figure 7(d)). Finally, metal1 is added to act as shields for more slot holes (Figure 7(d)). The added metal1 is connected to top metal via a sandwiched metal2 layer. Figure 7(d) also shows slotting done in metal1.

CONCLUSION

A context dependent DRC algorithm has been presented which addresses the issues of MEMS DRC. A greedy slotter has also been presented which uses minimal MEMS-compatible slot holes to address the density requirement of deep submicron processes. The capability of the DRC algorithm and the slotter have been demonstrated. Such a set of layout verification and correction tools will enable error-free designs thereby avoiding loss of money and time in fabricating erroneous layouts.

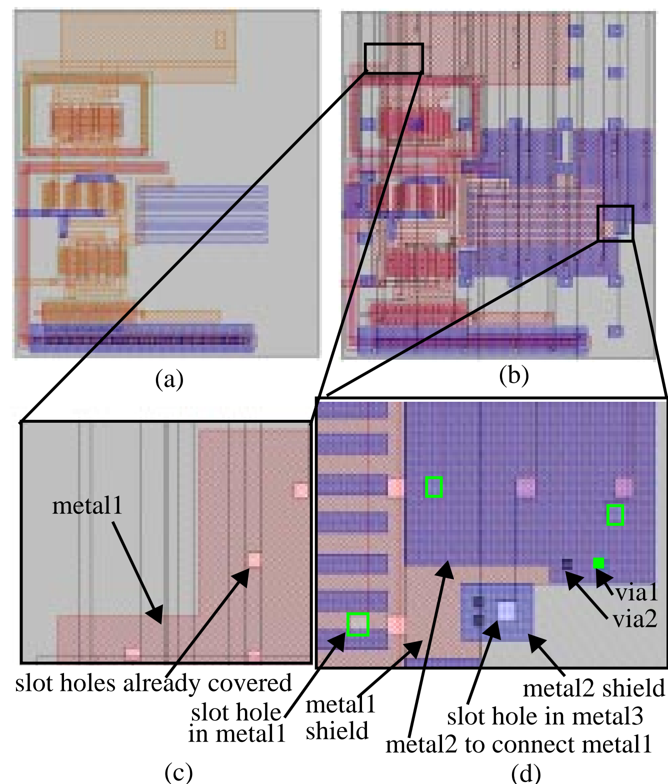


Figure 7: Slotting for a buffer circuit; (a) original layout, (b) slotted layout, (c) slot holes in already shielded areas, (d) metal1 and metal2 shields for slot holes

ACKNOWLEDGEMNTS

This research effort is sponsored by the Defence Advanced Research Projects Agency (DARPA) and U. S. Air Force Research Laboratory, under agreement number F30602-97-2-0323 and F30602-99-2-0545 and in part by the National Science Foundation award CCR-9901171. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA, the U. S. Air Force Laboratory, or the U. S. Government.

The authors also wish to acknowledge Mr. Xu Zhu for providing the SEMs.

REFERENCES

[1] T.G. Szymanski, C.J.V. Wyk, “Goalie: A Space Efficient System for VLSI Artwork Analysis,” *IEEE Design & Test of Computers*, vol.2, no.3, June 1985, pp. 64-72.

[2] N.R. Swart, “A Design Flow for Micromachined Electromechanical Systems,” *IEEE Design & Test of Computers*, vol. 16, no. 4, Oct.-Dec. 1999, pp. 39-47.

[3] X. Zhu, D.W. Greve, G.K. Fedder, “Characterization of Silicon Isotropic Etch by Inductively Coupled Plasma Etch in Post-CMOS processing,” *Proc. IEEE MEMS*, Miyazaki, Japan, 23-27 Jan. 2000, pp. 568-73.

[4] Cronos web page, *http://www.memsrus.com*.

[5] A.B. Kahng et al, “Filling and slotting: analysis and algorithms,” *Proc. 1998 Intl. Symp. Physical Design*, Monterey, CA, USA; 6-8 Apr. 1998, pp. 95-102.

[6] Marc Madou, *Fundamentals of Microfabrication*, CRC Press, 1997.

[7] A. Misaka, K. Harafuji, “Simulation Study of Micro-Loading Phenomena in Silicon Dioxide Etching,” *IEEE trans. Elec. Dev.*, vol. 44, no. 5, pp. 751-760.

[8] M. Armacost, et. al., “Plasma-etching processes for ULSI semiconductor circuits,” *IBM Jnl. of Research and Development*, vol. 43, no. 1-2, pp. 39-72.

[9] Sitaraman Iyer et al, “Macromodeling Temperature-Dependent Curl in CMOS Micromachined Beams,” *to be published in MSM 2001,* Hilton Head, South Carolina, Mar. 19-21, 2001.

[10] B. Baidya et al, “MEMS Component Extraction,” *Proc. MSM ‘99,* San Juan, Puerto Rico, Apr. 19-21, 1999, pp. 143-6.

[11] B. Baidya, T. Mukherjee, “Challenges in CMOS-MEMS Extraction,” *to be published in MSM 2001,* Hilton Head, South Carolina, Mar. 19-21, 2001.