MEMS Component Extraction

B. Baidya (bbaidya@ece.cmu.edu)*, S.K. Gupta (skgupta@eng.umd.edu)**
and T. Mukherjee (tamal@ece.cmu.edu)*
*Carnegie Mellon University, Pittsburgh, PA 15213, USA
**University of Maryland, College Park, MD 20742, USA

ABSTRACT

Surface micromachined structures are composed of *atomic elements* like anchors, beams and fingers, which can be further grouped into *components* like springs, comb drives and plates. Automatic recognition of these elements and components is crucial for a structured design methodology in MEMS (Microelectromechanical system). As MEMS design tends to be layout-centric, design evaluation requires *extraction* of the atomic elements from the layout. Furthermore, MEMS *component extraction* reduces the size of the simulation problem, enabling efficient design evaluation. An improved extraction module has been developed for component extraction that generates the netlist of the schematic corresponding to the layout. An ordinary differential equation solver combined with component models can then be used for efficient functional verification of the layout by simulating the extracted netlist. The utility of the extractor is demonstrated for a variety of MEMS devices composed of different types of springs and electrostatic actuators and sensors. Simulation time for the extracted netlist decreased by a factor of 10 when component extraction and component models were used compared to a netlist of only atomic elements.

Keywords: MEMS, canonical representation, component extraction, comb drives, springs, lumped parameter model

INTRODUCTION

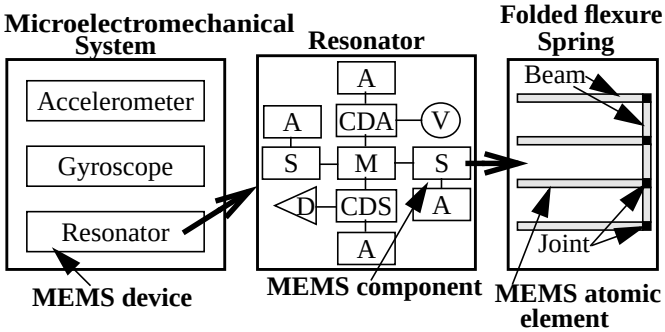
The acceptance of microelectromechanical systems (MEMS) in industry and the advent of stable fabrication process has recently led to the development of complex MEMS designs. A structured design methodology based on design hierarchy [1][2] has been proposed to handle such MEMS designs. The design methodology starts with a schematic design which is followed by a transcription of the design into a layout description. Currently the only means to verify the layout is to develop a mesh for the design and then perform finite element simulation. For complex designs this is prohibitively slow and interpretation of the results is very tedious. To overcome this problem we can make use of design hierarchy, to help ensure that the layout description is a correct spatial realization of the schematic, via the extraction of the atomic micromechanical elements that comprise the layout. The efficiency can be further increased by recognizing the MEMS components in the layout. The recognition of atomic elements on the basis of their shape, size and position was reported in ASME DETC '98 [3]. The elements are classified into anchors, plate masses, beams, cantilever beams, joints and holes. In this paper, we describe the extraction of components such as springs and electromechanical comb transducers after briefly discussing the algorithms that were used for recognition of atomic elements.

MEMS DESIGN HIERARCHY

As MEMS designs grow more and more complex, a need for hierarchical design representation (Figure 1) [1][2] is needed. A complex suspended microelectromechanical system is composed of a number of *devices* like resonators, accelerometers and gyroscopes. Each of these MEMS devices are in turn composed of *components* like mass, springs and comb drives. The components can be broken down into much more fundamental or *atomic elements* like beams, joints, anchors, plate masses and gaps. The advantage of such a hierarchical representation is that models [4][5] can be derived at each level of the hierarchy for a more efficient simulation. Hence extraction is needed at each level of the hierarchy. Following such a hierarchical design methodology also allows us to modularize a complex design by using the components as building blocks. This makes it possible to extract and simulate the entire design by extracting each subpart separately instead of the whole design at one time. In this paper we show our efforts towards extraction at a component-level.

DETECTION OF ATOMIC ELEMENTS

Recognition of atomic elements is needed prior to component-level extraction. This section describes the algorithms



A: anchor; CDA: comb drive actuator; S: spring; M: mass; CDS: comb drive sensor; V: source; D: detection electronics

Figure 1: Hierarchical MEMS design

used for recognition of the atomic elements. The first step towards the recognition process is to convert the given layout information into a set of rectangles which is unique for the design. This is the *canonical representation* of the layout and is defined as the representation which uses the minimum number of rectangles to cover the given layout area, such that infinitesimal outward extension of an edge of any rectangle never intersects with the interior of the layout area. The rectangles in the canonical representation have at most one neighbor per edge. This canonical set is developed incrementally by transferring rectangles, one at a time, from the input set of rectangles to the output canonical set. The output set is always maintained in a canonical condition and any additions that disturb this equilibrium results in a series of partitions such that the equilibrium is restored.

The recognition process first uses information of the nonstructural layers. Box overlap checking between these layers and the structural layer results in hints about the location of plate mass and anchor areas. This is followed by the recognition of beams and fingers. A beam element is defined to be a rectangle which has neighbors only at its two shorter edges. Cantilever beams are defined as fingers. This is followed by the detection of physical holes in the structural layers. These are filled up so as to reduce the number of nets in the final netlist. These hole areas are also marked as locations for plate masses. The hints for plate masses and anchors are then used to mark the rest of the rectangles by recursively expanding from these seed rectangles. After the entire layout has been recognized merging is used to reduce the number of rectangles and hence the number of nets in the final netlist. The merging routine finds out the best out of maximally vertical or maximally horizontal representation for each of the connected set of similar type and replaces them with the best representation. For example, a plate mass may be partitioned into very small rectangles during the canonical representation which can be merged back at this step in order to reduce the number of nets in the final netlist. The connectivity between the recognized rectangles can now be used to generate the netlist or they can be used for component-level recognition which results in a simpler netlist.

COMPONENT EXTRACTION

The atomic elements can be grouped to define MEMS components. This section describes the algorithms used for extraction of electromechanical comb transducer and spring components. A simple comb drive consists of interdigited comb fingers (Figure 2(a)) placed on electrically disconnected rotor and stator forming a capacitive arrangement used for sensing or actuating mechanical motion [6]. Fingers with pedestals (Figure 2(b)) are sometimes used to improve sensitivity without increasing number of fingers [7]. A modified comb drive structure with alternate polarity at every stationary drive finger (Figure 2(c)) is often used to reduce the levitation problem faced by the above comb drives [8]. Such a structure is also used to sense transverse motion via differential sensing of the two sets of capacitances.

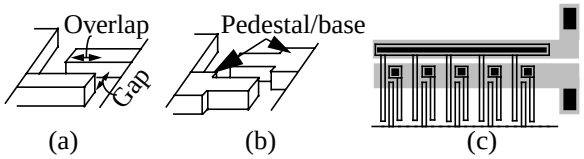


Figure 2: (a) linear comb fingers; (b) fingers with pedestal; (c) differential comb drive

Springs are composed of beams and joints and connect the suspended plate mass to the anchors. A fixed-fixed flexure (Figure 3(a)) has a very stiff spring constant because of extensional axial stress in the beams. Crab leg and U-spring (Figure 3(b) & (c)) are modifications to the fixed-fixed beam so as to reduce peak stress in the flexure at the cost of reduced stiffness in undesired directions. A folded flexure (Figure 3(d)) also reduces axial stress and gives more compliance while occupying lesser area. A meander spring (Figure 3(e)) is a modified version of a fixed-fixed flexure which helps achieve more compliance using less space.

Comb drive extraction

Comb drive extraction starts with a connectivity analysis of the set of recognized fingers. Fingers having electrical connectivity are given same connectivity number. Fingers are then sorted into buckets based on their orientation. Each such bucket is then checked for uniformity of the fingers with respect to region of occurrence, length of fingers, width of fingers and inter-finger gap. If the fingers have pedestals then the region of occurrence, length and width of the pedestal, inter-pedestal gap and the relative position of the pedestal with respect to the thin cantilever finger are also checked. The buckets are partitioned whenever any nonuniformity is found in any of these parameters. A box cover of each of the buckets is then created. The box covers are checked mutually for overlap using box overlap rules. Whenever an overlapping pair is found between two buckets having different connectivity numbers they are matched in size and combined to form a comb drive. The matching function takes care that there are no uncoupled comb fingers in the final comb drive. If one of the overlapping sets have comb fingers which do not couple capacitively with any of the fingers of the other set, then the

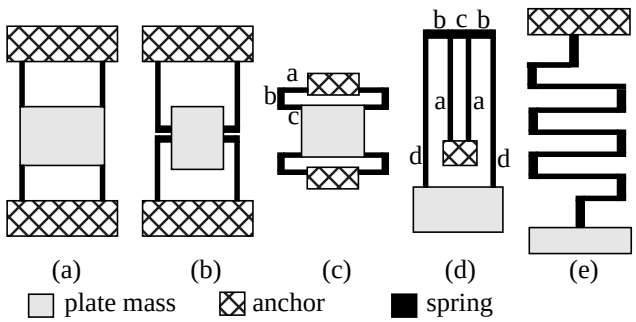


Figure 3: (a) fixed-fixed flexure; (b) crab leg; (c) U - spring; (d) folded flexure; (e)meander spring

finger set is partitioned so that the final pair only has all coupled fingers. Overlapping triplets are also detected and checked to see whether they form differential finger comb drives. In such a case the rotor set is matched with each of the stator sets to get the final triplet which are then merged to form a comb drive.

Spring Extraction

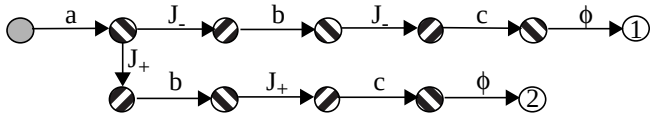
Spring detection is done using a Finite State Machine (FSM) based algorithm. Unlike FSMs the algorithm may have more than one final state whose simultaneous satisfaction is necessary for the outcome of the recognition to be true. The FSM can be defined by M = {S, L, U, G, F}, where
S = start state;
L = language = {joint, beam, NULL};
U = transition states which are either joint-state (states which accepts either joints or NULL) or nonjoint-state (state which accepts only beams);
G = the set of rules for the FSM; and
F = set of final states.

Table 1: Dictionary of joints

Joint name	m - param	t- param	ports	example
J_+	+1	0	2	
J_-	-1	0	2	
J_{T0}	0	0	3	
J_{T+}	+1	+1	3	
J_{T-}	-1	+1	3	
J_0	0	+1	4	

A joint is defined to be a node having one input node and at most three output nodes and is labelled using the 'm' and 't' parameters. The t-parameter is 1 only if there is an output port along the direction of the input port. An output port at right angles to the input port contributes a +1 or -1 to m-parameter depending whether the twist direction is anticlockwise or clockwise. The six types of joints possible using such a convention are shown in Table 1. The set of beams for the language depend on the spring to be detected. For example, a U-spring requires three beams (Figure 3(c)) which may or may not be equal in dimension, while a folded flexure requires four type of beams which must be arranged as shown in Figure 3(d).

The FSM for each of the springs is created by reading in the description of the FSM from library. The connected sets of



start state; joint state; non joint state;
alternative final states; a,b,c: beams as defined inFig3c

Figure 4: FSM for a U-spring

beams and springs obtained after the atomic recognition is then passed through each of these FSMs to recognize their type. For each such set the input is started from anchor in the spring netlist.

RESULTS

This section shows a select few results to demonstrate the capability of the current component extractor which implements the algorithms discussed above. The current implementation is for Manhattan-style MUMPS [9] designs. Improvement of the algorithms to handle different processes and non-Manhattan designs is currently underway.

Folded Flexure Resonator

Figure 5 shows a folded flexure at various stages of the recognition process. The extracted netlist was simulated using lumped parameter models [4][5] for the components and the simulation result is shown in Figure 5(e). The simulation was found to be more than 10 times faster than when only atomic elements were used in the simulation [4][5].

Accelerometer

Figure 6(a) shows an accelerometer which uses a meander spring and a differential comb drive. Figure 6(b) shows that the components were correctly recognized.

Gyroscope

Figure 7(a) shows a three-fold symmetric gyroscope which uses U-springs and beams for its suspension mecha-

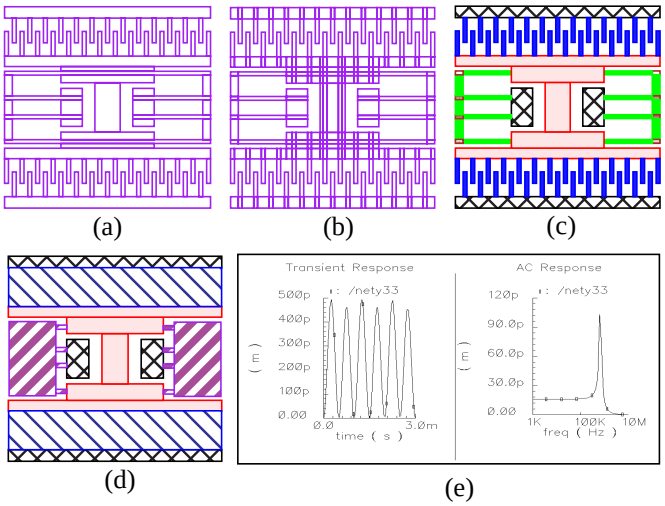


Figure 5: folded flexure resonator; (a) layout, (b) canonical representation, (c) recognized layout, (d) component extraction, (e) transient (1KHz source) and ac (resonant frequency = 691.8KHz) analysis of extracted netlist

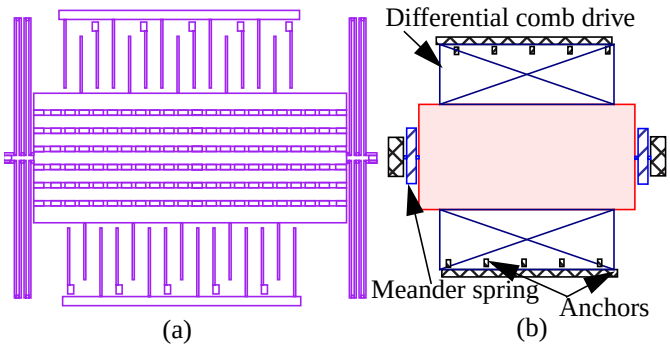


Figure 6: accelerometer using a meander springs and differential comb drive; (a) input layout, (b) extracted layout

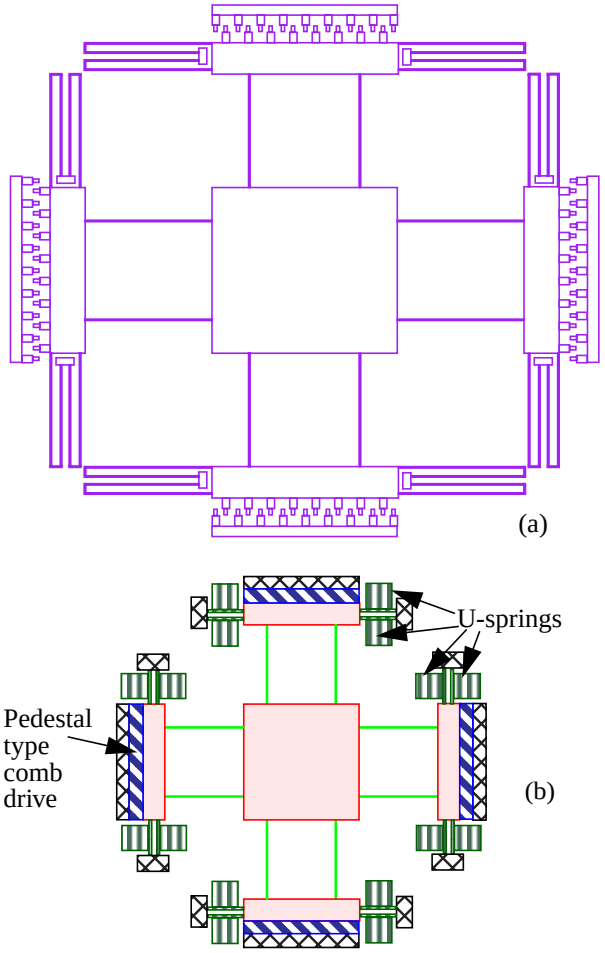


Figure 7: three-fold symmetric gyroscope using U-springs and pedestal type comb drive

nism and uses pedestal type fingers in its comb drive for increased actuation. Figure 7(b) shows the final extracted layout.

CONCLUSION

Component recognition during layout extraction leads to fewer elements in the simulation netlist. A design methodology that couples component-level recognition and component simulation models can lead to an iterative design cycle based on interactive feedback about device quality. An extraction module which achieves component-level recognition has been shown in this paper.

ACKNOWLEDGEMENT

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. 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. Sitaraman Iyer and Ms. Qi Jing who helped by providing necessary models for the lumped parameter simulator.

REFERENCES

[1] J.E. Vandemeer, M.S. Kranz, G.K. Fedder, "Hierarchical Representation and Simulation of Micromachined Inertial Sensors," *Proc. of MSM*, Santa Clara, LA, Apr. 6-8, 1998 pp. 540-545.

[2] G. K. Fedder, "Structured Design of Integrated MEMS," *Proc. of MEMS '99*, Orlando, Florida, Jan. 17-21, 1999, pp.1 -8.

[3] B. Baidya, S.K. Gupta, T. Mukherjee, "Feature-Recognition for MEMS Extraction," *CDROM Proc. 1998 ASME DETC*, Atlanta, GA, Sept. 13-16, 1998.

[4] G.K. Fedder, Q. Jing, "NODAS 1.3 - Nodal Design of Actuators and Sensors," *IEEE Transactions on Circuits and Systems (II) Special Section on BMAS* (to be published).

[5] S. Iyer, Y. Zhou, T. Mukherjee, "Analytical Modeling of Cross-axis Coupling in Micromechanical Springs," *MSM '99*, San Juan, Puerto Rico, April 19-21, 1999. (to be published).

[6] W.C. Tang, T.-C.H. Nguyen, M.W. Judy and R. T. Howe, "Electrostatic-comb Drive of Lateral Polysilicon Resonators," *Transducers '89*, Vol. 2, pp. 328-331, June 1990.

[7] T. Hirano, T. Furuhata, K. J. Gabriel, and H. Fujita, "Design, Fabrication, and Operation of Submicron Gap Comb-Drive Microactuators," *J. of MEMS*, Vol. 1, No. 1, March, 1992.

[8] W.C. Tang, M.G. Lim and R.T. Howe, "Electrostatically Balanced Comb Drive for Controlled Levitation," *Technical Digest IEEE Solid-State Sensor and Actuator Workshop*, pp. 23-27, June 1990.

[9] D.A. Koester, R. Mahadevan, K.W. Markus, *Multi-User MEMS Processes (MUMPs) Introduction and Design Rules*, available from MCNC MEMS Technology Applications Center, 3021 Cornwallis Road, Research Triangle Park, NC 27709, rev. 3, Oct. 1994, 39 pages.