A Top-Down Microsystems Design Methodology and Associated Challenges

Michael S. McCorquodale, Fadi H. Gebara, Keith L. Kraver, Eric D. Marsman, Robert M. Senger, and Richard B. Brown
Department of Electrical Engineering and Computer Science
Solid State Electronics Laboratory
University of Michigan
Ann Arbor, Michigan 48109-2122

ABSTRACT

An overview of microsystems technology is presented along with a discussion of the recent trends and challenges associated with its development. A typical bottom-up design methodology is described and we propose, in contrast, an efficient and effective top-down methodology. We illustrate its implementation with the development of a microsystem design that has been completed and fabricated in CMOS technology. Gaps in the tool capabilities are identified and suggestions for future directions in CAD tool support for microsystems technology are presented.

1. Introduction

Microsystems technology is defined as an intelligent miniaturized system comprising sensing, processing and/or actuating functions where two or more of the following technologies are combined onto a single or multichip hybrid: electrical, magnetic, mechanical, optical, chemical, or biological [1]. Building a complete microsystem involves several challenges as these designs include not only a union of the analog and digital circuit domains, but also the magnetic, mechanical, biological, chemical, or electrical domains. Moreover, the design constraints associated with systems such as these can be as specific as the material properties of a layer that defines a microstructure to as broad as an abstraction of the embedded processor that supports the firmware for the microsystem. A variety of tools for completion of such designs exist, but as yet, there is no complete end-to-end framework for development. In this work, we have aimed to leverage advances in integrated circuit CAD tools and couple them with trends in microelectromechanical systems (MEMS) and mixed-signal circuit design to address the challenges associated with the development of microsystems. We propose an efficient and effective design methodology for such development, while identifying gaps that call for new design automation developments.

2. Design Trends and Challenges Associated with Microsystems Technology

Fig. 1 illustrates a generalized end-to-end wireless integrated microsystem. Here the various technologies along with the typical design tools for each are illustrated. Tremendous breadth exists when developing such systems. For example, MEMS components are often developed with finite element (FE) tools that simulate mechanical response to an applied stimulus, while the microprocessor section is almost completely synthesized with some form of hardware description language (HDL). Digital IC design tools are now ubiquitous and offer the designer tremendous flexibility through system abstraction. Only recently have such trends developed in the analog and mechanical domains.

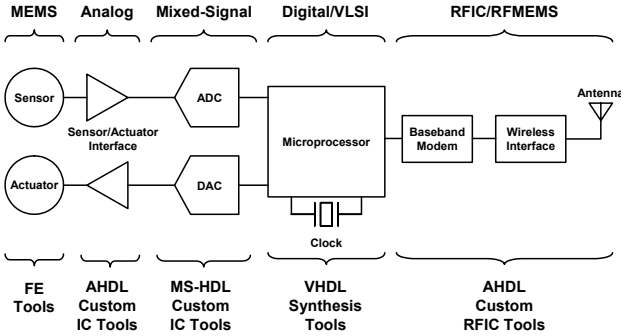

Fig. 1: The anatomy of a generalized wireless integrated microsystem. Key technologies and associated development tools are shown.

There are several existing, and likely future, MEMS technologies that warrant integration with CMOS or a related process technology. Indeed a great deal of research has been underway in this field including activities in monolithic MEMS-based oscillators [2], accelerometers [3], and switches [4], to name just a few. Only recently have such subsystems been developed and thus ambiguity in a wholistic design flow and gaps in the related CAD framework are not surprising. Clearly, a design methodology for such systems that addresses the challenges associated with the convergence of these technologies is required if complete systems that implement these research breakthroughs are to ever come to fruition.

Current and past development approaches have been typically ad-hoc and bottom-up in nature. This design methodology is an outgrowth of both the disparate nature of the technology and the process by which this technology has developed. Much MEMS work to date has been focussed on device development. Once device performance is optimized, supporting electronics are added incrementally. Therefore, a bottom-up development approach is rather natural. However, now these devices are appearing within much larger systems and the typical

development strategy is to partition sections of the microsystem into the mechanical, analog, or digital domains. Design activities become disjoint and ad-hoc from here, where each subsystem is designed with a separate tool suite and with little, if any, cross-domain verification. As discussed previously, MEMS technology has been designed almost exclusively with finite element tools, however the majority of these simulators do not support an interface with a standard IC framework. Therefore, in almost all applications, some level of model extraction and abstraction is required for simulation of the MEMS component with the supporting analog electronics. Often, this extraction is custom tailored to each component and it must be completed by the designer without the aid of design automation. Additionally, several of these systems require logic for programming or trimming, and in many applications a complete embedded processor is required to support the system. Here the standard tool suites allow designers to synthesize digital logic and physical design from a hardware description language, but typically verification is not performed with the analog and MEMS devices integrated into the microsystem.

As Fig. 1 and the previous discussion illustrate, several disparate tools are required for the development of microsystems. Specific design challenges involve management of these tools as well as system verification across these various design platforms. Clearly, the number and complexity of tasks involved in the development of microsystems are significant. Attention to design methodology has become increasingly important in order to develop systems efficiently and close the design gap between manufacturing and design capabilities [5].

3. Typical Design Methodology: Bottom-Up

A bottom-up design methodology involves the development of each block from the device to system level. Devices are combined to form blocks, which are then combined to complete and verify the system. In [6], the problems associated with a bottom-up design methodology are addressed. They include lack of architectural study and optimization, costly redesign effort associated with iteration through the flow, and significant processing time for system-level simulation, if it is even possible.

Fig. 2 illustrates this typical design methodology as applied to microsystems technology. Here a system specification is translated into a specification for three domains: digital, analog, and mechanical. Design activities ensue from the device to block level and from the block to system level. A macro is delivered from each domain and the system is assembled with an automatic place and route (APR) tool. The system is then verified and only at this point are problems addressed. Therefore, time-consuming redesign effort is required back at the device level. Moreover, additional iterations are also common with the APR tool in order to optimize macro placement.

Although methodologies such as these have been employed in the past, they are clearly insufficient for complex microsystems. As the field matures, it is likely that microsystems will contain several, if not hundreds, of magnetic, mechanical, optical, chemical, or biological components along with the supporting analog and digital devices. A proper, efficient, and exhaustive design methodology and framework is obviously required.

4. Proposed Design Methodology: Top-Down

The proposed design methodology has been implemented in the development of a complete microsystem, pictured in Fig. 3. The microsystem is comprised of five major subsystems: a microprocessor core, memory, peripherals, an analog front end (AFE), and a mechanically-tunable MEMS-based clock generator. The design was fabricated in *Taiwan Semiconductor Manufacturing Company's* (*TSMC*) 0.18µm mixed-mode CMOS process and consists of approximately 1.5 million transistors that occupy a silicon area of 10.24mm^2.

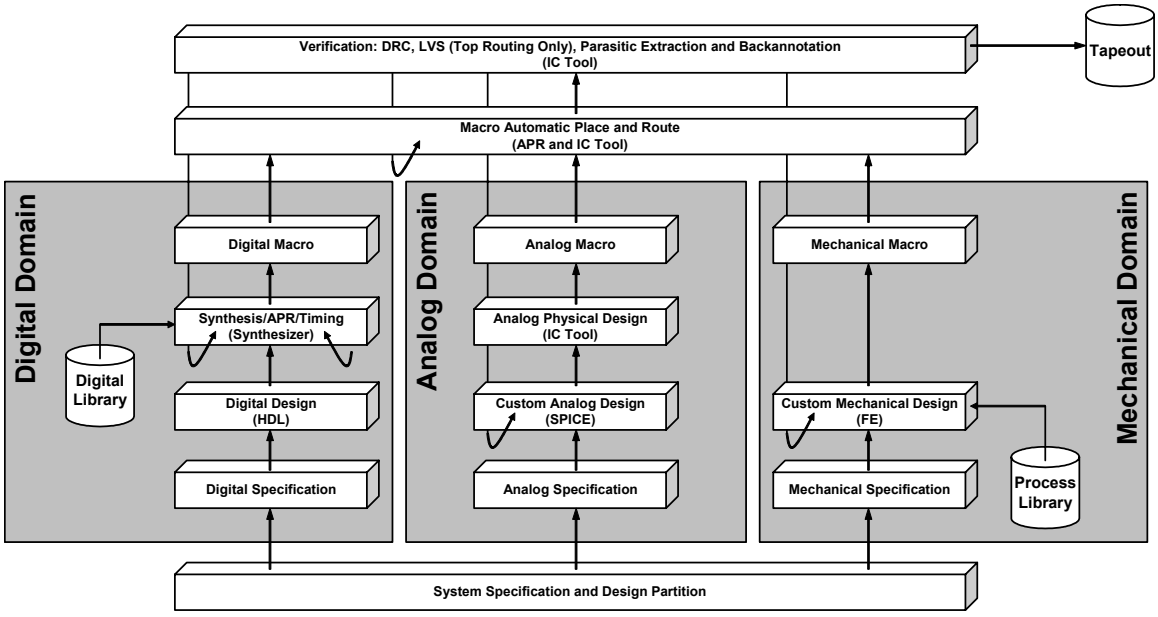

Fig. 2: Typical ad-hoc and bottom-up microsystems design methodology

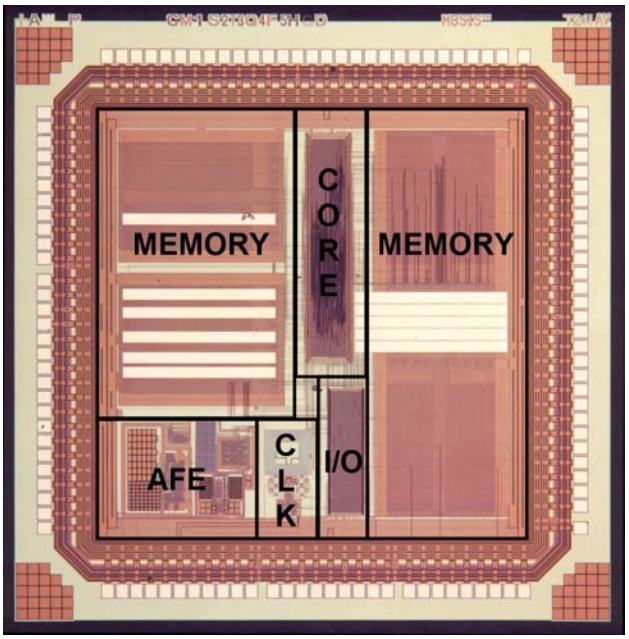

Fig. 3: Die micrograph of the fabricated microsystem in *TSMC*'s 0.18μm mixed-mode process. MEMS components are released by post processing.

4.1 A Top-Down Approach

In a top-down approach, development would proceed from the system to device level. The system could be studied and optimized with a mixed-signal HDL (MS-HDL) from which the abstract circuit blocks are derived. Device-level designs would then be completed, and achieved performance could be benchmarked against the original specification using the abstract blocks and system model. Throughout a top-down design methodology, cross-domain verification at various levels should be budgeted. This reduces the likelihood of time-consuming redesign effort at higher levels of system assembly.

In conjunction with the top-down approach, several additional development requirements were considered while constructing the design methodology. An environment that supports hardware abstraction and cross-domain simulation for MEMS, analog, and digital electronics was required. The environment also had to support simulation of abstract hardware with device primitives in order to accurately model digital programming of analog and mechanical components without synthesis of these digital devices. A model that could be modified easily for system verification based on the realized subsystem performance was desirable, as was cross-domain verification at every level of abstraction. The complete tool suite had to support low-level simulation including FE and basic transistor-level analysis, as well as non-linear RF and noise analysis. Support for HDL synthesis, timing verification, and APR was mandatory for digital design and final chip assembly.

Although no single design framework met all of these design requirements, we have found that the *Cadence AMS* environment is well-suited to achieving many of these goals for system-level development of microsystems technology. Our attention was first brought to this framework for the support of *Verilog-AMS*, an analog and digital HDL which is a superset of the *Verilog* and *Verilog-A* languages. We have found that *Verilog-AMS* is also ideal for behavioral modeling of mechanical devices. Prior to the emergence of *Verilog-A*, many MEMS engineers had been using device level models, including primitives, for MEMS component modeling. Clearly, the *Verilog-A* language is a significant improvement over this technique as it provides added modeling flexibility while it minimizes complexity.

Additional tools used in this work included *Spectre* for analog subsystem and transistor-level design, *Coventorware* for FE analysis of MEMS components, *Synopsys* for digital synthesis, *Cadence Silicon Ensemble* for APR, and *Mentor Graphics Calibre* for DRC and LVS. The requirement of such an extensive and disparate tool suite is a significant challenge faced in the development of microsystems technology.

4.2 Methodology

With a framework in place, a design methodology was determined, as illustrated in Fig. 4. *Verilog-AMS* was employed to realize the system specification. MEMS and analog components were modeled in *Verilog-A*, while the microprocessor core and peripherals were modeled in *Verilog*. From this system model, a natural partition of top-down subsystem design activities followed. Each block was specified with an abstraction for the hardware. In parallel with behavioral verification of the digital section, the blocks in the mechanical and analog domains were developed and performance metrics were determined. Updated *Verilog-A* was developed to model achieved performance from FE simulation in the mechanical domain while device-level design and analysis using *Spectre* led to achieving the analog specification. The digital electronics were developed such that a complete behavioral description of the hardware was realized. At this point, the first cross-domain verification of the system was achieved. Once the HDL from each domain had been updated with the achieved performance, verification of the system model was trivial. In the *Cadence AMS* environment, HDL and primitives may be mixed and critical subsystem performance metrics can be determined quickly with a detailed model for the subsystem and an abstract model for the remainder of the system. This was particularly significant when considering analog and MEMS device-level performance that required digital programming which was described only by HDL.

A system-wide simulation was completed and iteration in the mechanical and analog design activities continued, dependent upon system performance. This first cross-domain simulation offered significant benefits over the bottom-up methodology described previously. First, design effort had not been expended synthesizing the digital electronics. Second, iteration in the design of the MEMS and analog circuits occurred early in the design flow. Last, the system simulation was fast as it was described by behavioral HDL, not a complete device-level netlist. However, simulation was also timely in the case of a primitive-level subsystem simulation as the remainder of the system is described by HDL and only the critical blocks were modeled at the device-level.

System development continued with a typical physical design methodology. The digital sections were synthe-

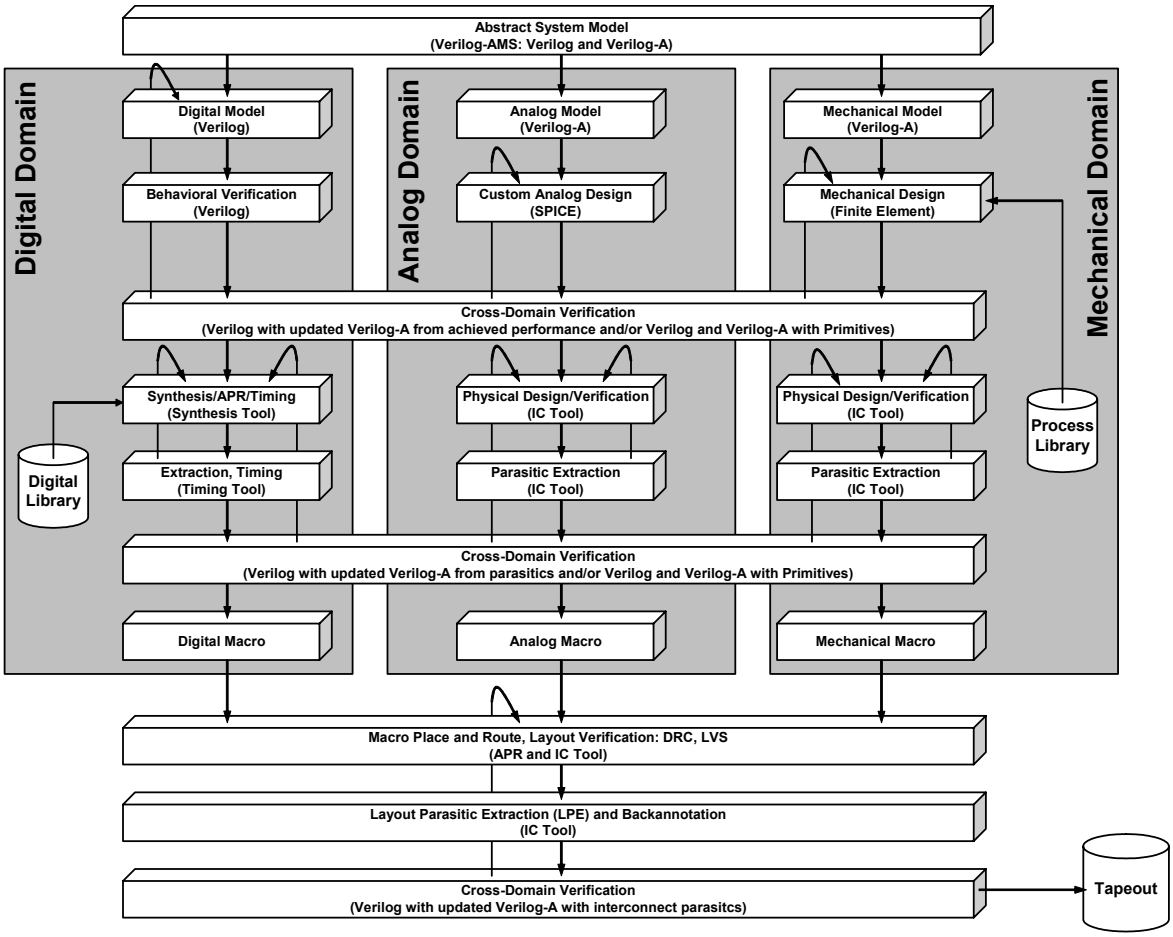

Fig. 4: Proposed top-down microsystems design methodology

sized and the mechanical and analog sections were custom designed. Timing information from the synthesis tool was used in an iteration to achieve timing closure for the digital section. Similarly, parasitic extraction and backannotation afforded an iterative process in completing the mechanical and analog sections. Once timing closure was reached in each domain, a second cross-domain simulation was executed for system verification based on physical design. Again, the HDL for the subsystems was updated and system simulation was timely and accurate. Physical design iteration continued until timing closure was achieved for the complete system. The domain-specific design activities completed with the delivery of a hard macro.

The final system development activities included APR, physical design verification (DRC, LVS), layout parasitic extraction (LPE), and backannotation. A final cross-domain verification was completed once parasitic extraction data for the interconnect between macros was determined. APR iteration was also necessary.

5. Gaps in the Tool Suite

Several CAD-related shortcomings were encountered throughout the development of this microsystem. All of these gaps are associated with the development of MEMS and analog electronics. First, we have noted that the design tool suite used lacks support of a MEMS behavioral model that is automatically extracted from FE simulation. Likewise, an analog behavioral model that is automatically extracted from SPICE simulation is not supported.

Throughout the physical design flow we encountered a lack of physical verification of the MEMS devices in primitive form. Synthesis libraries for MEMS and analog components from behavioral or topological models were also unavailable. As a result, porting capability of microsystem designs between process technologies through synthesis is not achievable.

We have developed several custom and elementary patches for these design gaps in order to automate design flow. For example, a custom CV model for the tunable component contained within the on-chip clock generator was developed, and results from FE electrostatic simulation are ported to this model. Additionally, the lack of MEMS device verification was overcome by custom modification of the DRC and LVS decks. Trivial fixes include *Verilog-A* models that were updated by hand from achieved performance in *Spectre*. However, this process becomes time-consuming when considering complex analog subsystems. Other similar patches were employed and we believe that automation of patches like these would greatly facilitate microsystem designs of the future.

Solutions to the other identified gaps are more diffi-

cult. Analog and MEMS synthesis from HDL or a topological model is indeed a significant endeavor. Nonetheless, some tools for this very purpose [7] are available today and active research in this field in underway such as work shown in [8]. With these synthesis capabilities, the design would have been completed in a more timely manner.

6. Conclusion

In this work we have presented an overview of microsystems technology and the trends and challenges associated its with development. A typical bottom-up design methodology was outlined and the associated shortcomings were addressed. By leveraging advances in mixed-signal and digital IC design tools, we have proposed an efficient and effective top-down design methodology. The methodology has been employed in the development of a complete microsystem. Gaps in the tool suite were presented as well as suggestions for future directions in CAD development. The authors aspire to promote appropriate tool development and integration so as to foster additional research activity in the field.

7. References

[1] European Union 4th Framework, *ESPIRIT Workprogramme*, Sept. 1996.

[2] D. J. Young, *et al.*, "A Low-Noise RF Voltage-Controlled Oscillator Using On-Chip High-Q Three-Dimensional Coil Inductor and Micromachined Variable Capacitor," *Solid-State Sensor and Actuator Workshop*, pp. 128-131, 1998.

[3] S. Lee, *et al.*, "Two-dimensional position detection system with MEMS accelerometer for mouse applications," *Proc. Design Automation Conference*, pp. 852-857, 2001.

[4] G. M. Rebeiz and J. B. Muldavin, "RF MEMS Switches and Switch Circuits," *IEEE Microwave Magazine*, vol. 2, issue 4, pp. 59-71, Dec. 2001.

[5] K. Kundert, *et al.*, "Design of Mixed-Signal Systems on Chip," *IEEE Trans. on CAD*, vol. 19, no. 12, pp. 1561-1572, Dec. 2000.

[6] K. Kundert, "A Formal Top-Down Design Process for Mixed-Signal Circuits," *Advances in Analog Circuit Design*, April 2000.

[7] D. Maliniak, "Embedded Design Challenges hold center stage at the 39th DAC," *Electronic Design*, pp. 55-60, June 2002.

[8] A. Doboli, *et al.*, "Behavioral Synthesis of Analog Systems using Two Layered Design Space Exploration," *IEEE/ACM DAC*, June 1999.