Interfacing requirements for MEMS components in system-on-chip methodologies

Neil W. Bergmann[*]
School of Computer Science and Electrical Engineering, The University of Queensland

ABSTRACT

Modern VLSI design is moving towards a System-on-Chip design paradigm, where chip design involves the integration of separate macrocells from different manufacturers. This paper explores the obstacles to adopting this same methodology for systems incorporating MEMS components. These obstacles include the technology specific nature of most MEMS devices, interference between MEMS sensors, and the limited electronics device density of mixed MEMS/Microelectronics technologies. It is conjectured that one fruitful avenue for further work is the development of MEMS interface circuits which can be incorporated into a single SoC along with other electronics macrocells, and which then connect to discrete MEMS sensor chips.

Keywords: MEMS, System-on-Chip, VLSI, interfacing

1. INTRODUCTION

In the past, electronic systems have often been built from a range of separate integrated circuits, discrete components, sensors, and actuators. For specialized, high volume applications many of the electronics components are integrated onto a single ASIC (Application Specific Integrated Circuit). As integration levels increase to the hundreds of millions of transistors, it is now becoming impractical for ASICs to be designed as a single unit by a single design team. Rather, the ASIC design process is moving towards a system-level-integration paradigm. Instead of a system being built by selecting compatible components from different manufacturers and integrating them on a printed circuit board, a System-on-Chip (SoC) is built by selecting compatible macrocells or IP blocks from IC design vendors, and integrating them onto a single IC [1]. The chip design process becomes a system integration problem, but one with many unique challenges and difficulties. To provide a complete SoC methodology, it is useful to be able to integrate sensors and actuators also, and this introduces the problem of MEMs System-on-Chip.

This paper first examines the major well-known issues in microelectronic system-on-chip design methodologies, and then addresses these problems again, but this time looking at the integration of MEMS components within a System-on-Chip methodology.

2. VLSI SYSTEM-ON-CHIP

2.1 Overall methodology

In a system-on-chip methodology, there are two main aspects to the design. Firstly, there is the design of the individual macrocells, such as processors, memories, arithmetic units, peripherals, and I/O Pads. Secondly there is the specification, integration, verification, fabrication and testing of the system as a whole. Successful system integration depends on macrocell designs which are compatible with the particular system integration methodology being used.

2.2 Macrocell design capture

In order for a macrocell to be used on a chip, the mask layout for the macrocell obviously needs to be inserted into the final chip mask layout. Given that the final macrocell mask layout will be unique to a particular fabrication technology,

[*] n.bergmann@csee.uq.edu.au, phone +61 7 3365 1182, fax +61 3365 4999, CSEE, The University of Queensland, St Lucia 4072, Australia.

Electronics and Structures for MEMS II, Neil W. Bergmann, Derek Abbott, Ahsan J. Hariz,
Vijay K. Varadan, Editors, Proceedings of SPIE Vol. 4591 (2001) © 2001 SPIE · 0277-786X/01/$15.00

there is a strong incentive for the designers to capture the IP content of the macrocell in a technology independent manner which can be customized for a range of current and future fabrication technologies.

For macrocells where geometric arrangement has a critical impact on circuit size and performance, this geometrical arrangement can be captured by designing the cell at symbolic (or "sticks") level [2], with simple automated conversion to mask layout. Examples would be memory array design, where efficient layout requires a rectangular grid of cells. Such a symbolic layout necessarily limits the target fabrication technologies to those with the same functional mask layers as the symbolic layout. For example, if the symbolic layout assumes 3 layers of metal interconnect, then the target process must be able to provide this.

Capturing the design at logic level, perhaps in synthesizable VHDL [3] or Verilog [4], extends the range of target technologies considerably, by allowing the design to be synthesized to layout on a case-by-case basis. Synthesis can be done either by the macrocell designer, by the SoC integrator, or by a trusted third party such as a design house. The disadvantage of capturing the IP at logic level is that layout efficiency depends on the synthesis tool. There is also the problem of post-layout functional verification, since synthesis can introduce new critical paths and race conditions. Nevertheless, the advantages of a technology independent, long lifetime design suggest that the majority of IP blocks are likely to be captured and maintained at logic level, with case-by-case synthesis to mask layout.

2.3 Macrocell encapsulation

What does an SoC integrator get when they purchase a macrocell from a third-party vendor? Usually they are purchasing the rights to use that macrocell design in one or more of their IC's. There may also be a royalty fee for each chip sold using the macrocell. Much like the purchase of a software program, the item which is purchased in the form of some digital files is really the purchase of the rights to use some piece of Intellectual Property. Since this IP constitutes considerable work on the part of the macrocell designer, that designer will hope that the macrocell can be licensed many times, and protection and tracking of IP is important.

Particularly in the case where synthesis is undertaken by the SoC integrator, that integrator effectively has access to the source code of the macrocell. In the commercial software world, end user access to source code is rare. One would expect the same to be true in the SoC world. We might expect many macrocell designers will be individual designers or small companies who might lack the resources to maintain synthesis capability to a wide variety of target processes, and we can expect to see the emergence of trusted systems integration houses who will do the synthesis task on behalf of the macrocell vendors.

Because VLSI circuits must be thoroughly verified before fabrication, there is a necessity to provide a simulation model for each macrocell in addition to the physical layout. This is also problematic, since in many cases the synthesizable VHDL or Verilog code, with post layout delays added, provides the most straightforward accurate simulation model. Supply of this model again equates to releasing source code. One solution is to provide behavioural level simulation code, although these programs may also reveal much of the internal macrocell structure. Another solution is again to use a trusted system house to provide either simulation services (so that SoC integrator only gets I/O information on macrocell blocks). Still another solution is the use of some type of "compiled code" simulation strategy, where the SoC integrator gets executable but not viewable models of macrocell operation.

Protecting macrocell IP while still allowing the detailed simulation necessary to ensure whole chip verification remains a major obstacle to the widespread adoption of SoC technology. It is seen that system design houses, in the role of trusted IP broker, are likely to be one route to achieving this compromise. CAD tool vendors such as Cadence have set up major design facilities, such as the "Project Alba" campus [5], to facilitate this broker role.

The SoC industry is obviously keenly aware of these issues, and the Virtual Sockets Industry Alliance [6] [7] has done a lot of work towards suggesting workable solutions. VSIA have produced a number of specifications for data exchange between macrocell designer and SoC integrator, covering issues such as physical interfaces, test-sequences and simulation models.

2.4 System interconnection strategy

In addition to IP blocks on an SoC, there needs to be some interconnection strategy (such as a bus) and some interconnection logic (such as address decoders).

In conventional board level design, interconnection logic is greatly reduced if all the components adhere to the same interconnection strategy. For example, the PCI standard [8] allows processor-peripheral interconnection across a broad range of computer systems.

The integration of SoC modules is also greatly eased if all modules are designed for the same interconnection strategy. There are several vendor standards for SoC interconnection strategies. One popular example is ARM's AMBA on-chip bus standard [9], which specifies both physical and protocol characteristics for interconnecting AMBA-compliant macrocells. This bus protocol is processor-centred, and mostly used in ARM-powered SoC applications.

In a conventional computer system, one finds several different levels of interconnection. At the processor level, registers and ALU are interconnected using some high-speed embedded busses. At the next level is a high-speed processor-to-cache or processor-to-memory bus. At the next level, is the backplane interconnection bus, such as PCI. At the wider level is a Local-Area Network such as Ethernet.

In SoC, such hierarchies have already been proposed. The AMBA specification includes AHB (AMBA High-speed Bus) and APB (AMBA Peripheral Bus) busses which mimic those levels in conventional systems. Others have suggested that LAN techniques could well be appropriate in very complex, multiprocessor SoC applications. Protocols such as Ethernet might be just as applicable across a chip as across a campus, and can extend the reach of the Internet to individual sub-modules within a chip.

2.5 Testing

Any IC needs testing to ensure correct fabrication. Given a macrocell with limited internal visibility, the system integrator must rely on the test information provided by the macrocell designer, and must also devise strategies to undertake testing using the supplied test vectors, or else be able to invoke some internal self-test capability of the macrocells.

Board level test strategies such as JTAG [10] can be extended to within a chip. It is unlikely that a JTAG serial link between complex modules in a 100 million transistor SoC will provide an efficient test platform for these chips. More likely, each chip will have some specialized test interface module, connected to the system interconnection bus, and externally accessible by a JTAG interface. Increasingly individual macrocells will need to be responsible for generating test vectors, observing outputs, and reporting back appropriate test information.

2.6 Liability

With mask sets for state-of-the-art IC processes running into many hundreds of thousands of dollars, any required revisions to a circuit are expensive in the extreme. Given that there can be several macrocell vendors, a system integration house, and an overall SoC integrator, the source of blame for a faulty design may be difficult to identify. The need to have clear legal guidelines which represent the conditions under which a macrocell is provided, and the guarantees that accompany it are likely to be key components of any IP-based virtual component market place.

2.7 Summary

The market for virtual components which are available for integration into SoC products is still in its infancy. Many potential problems exist, although industry groups are working hard to overcome these. Of some concern is that these industry groups are driven by some of the major CAD vendors, and the solutions proposed are likely to require access to expensive CAD systems, which reduces the ability of small independent designers to participate in this marketplace.

Nevertheless, there do not appear to be any insurmountable obstacles which will prevent the more widespread acceptance of this System-on-Chip methodology for VLSI circuits over the next decade or so. We now look at the characteristics of MEMS devices to investigate their ability to participate in the same SoC methodology.

3. MEMS SYSTEM-ON-CHIP

3.1 Overall Methodology

MEMS SoC integrates MEMS-based sensors, actuators and other devices along with analog and digital electronics, including processors and telecommunications interfaces, to provide a complete system. While adding sensors and actuators further decreases system cost, it is still rare to be able to integrate a complete system onto a single chip. Delivering power to a chip usually involves additional circuitry and connectors, as do communications interfaces.

For our purpose SoC does not really refer to whether or not a single chip constitutes a complete system. Rather, it refers to a chip design methodology where the primary task of the chip designer is to integrate a number of separate macrocells from different sources onto a single chip. For this definition of SoC, it is less important whether the whole system fits onto one or many chips, rather it is important that the chip designer can choose from a wide range of electronics and MEMS macrocells and can use these to build individual chips.

In many cases, MEMS manufacture is incompatible with the efficient manufacture of complex (multi-million transistor) circuitry on the same substrate. For this reason, we will consider two different scenarios. The first scenario we call monolithic SoC, where all electronics (digital and analog) and also MEMS sensors and actuators are built on the same monolithic substrate. The second scenario we call hybrid SoC, where MEMS devices are built on one chip, and electronics are built on a second chip, each using a different fabrication flow. These chips could be mounted together on an MCM (Multi-chip Module), or else could be separate components on a PCB.

3.2 Macrocell design capture

For each MEMS sensor/actuator, we can expect there to be some interface electronics which supplies the necessary signal conditioning and data conversion. Therefore, each MEMS sensor/actuator will consist of two macrocells, one for the MEMS device itself (such as an accelerometer), and one interface macrocell consisting of analog and digital electronics.

Any digital electronics circuitry can be captured in Verilog/VHDL and compiled to mask layout for each different fabrication technology. Analog circuitry is less easy to capture in a technology independent manner. Mixed-signal enhancements to VHDL and Verilog go some way towards solving this problem. In general, analog circuits are relatively small, and we might expect that manual customisation of the analog interface electronics would be a necessary part of porting an interface circuit to new technology.

Another possibility is to define all interface circuits in terms of a set of “generic” analog primitives (such as op amps, mixers, multiplexers, switches, oscillators). Interface circuits could quickly be ported to any fabrication process which implemented the generic analog library. At this stage, the set of generic primitives is not sufficiently well understood to adopt this approach.

Current MEMS devices are typically designed for a single fabrication technology. There is not really any generic method for specifying a MEMS design which can be implemented in multiple fabrication technologies. In the near future, we would expect that MEMS devices will need to be manually redesigned for each new fabrication technology. Indeed many MEMS devices are intimately tied to their fabrication technology, and are customised for each new application.

For the next few years at least, we can expect specific MEMS macrocells to be manually customised for a specific technology. Each individual fabrication technology will have its own set of available macrocells. Technology independent macrocells will not appear for some time yet.

For monolithic MEMS SoCs, it can be expected that the implementation technology will be far short of the multi-million transistor leading edge CMOS processes. Rather, we can expect monolithic MEMS SoCs to incorporate electronics modules such as processors which have been specifically designed to be compatible with the limited fabrication technologies available.

3.3 Macrocell encapsulation

Because MEMS macrocells will be individually customised for each new fabrication technology, access to the macrocells will normally be in terms of the mask set (or other fabrication instructions) required to manufacture the device. The "source code" problem of IP access and protection will therefore be a major problem for MEMS SoC.

The solution will probably need to be the same as for VLSI SoC. Access to the mask layout will be restricted to a trusted system house who will do the final physical system integration.

The system designer will have access only to an interface model of the MEMS macrocells. Electrical and mechanical simulation information will also be provided for system designers. In the case of MEMS devices, it is less likely that these simulation models can be "reverse engineered" to capture the IP content of the device.

The MEMS community currently lacks both the dominating CAD vendors and system houses that characterise the SoC community in VLSI design. This means that early players in the MEMS SoC market have a real opportunity to define the way that this marketplace develops.

3.4 System interconnection strategy

MEMS sensors/actuators are currently designed as standalone components. To be part of an SoC methodology, considerably more attention must be paid to system issues. This tends to have most effect on the design of the interface macrocells.

Many SoC designs are processor-centred, and for this paper we will concentrate on such systems. In processor-centred systems, the typical interconnection strategy is bus-based. MEMS modules would typically interface at the peripheral bus level (such as AMBA APB). It would therefore make sense for MEMS interface logic to be bus-based. Each MEMS sensor would therefore require not just analog signal conditioning and power switching circuitry, but would also require a bus-based digital controller.

For hybrid MEMS circuits, there is a partitioning problem. On which chip should the bus-interface logic be placed? If placed on the MEMS-chip, then the connection from electronics chip to MEMS chip can just be a single bus which deals with all MEMS sensors on the MEMS chip. If placed on the electronics chip, then each sensor on the MEMS chip will have a separate dedicated connection between sensor on one chip and interface on the other. This partitioning will be one major design decision to be made.

Note that a single MEMS sensor could be provided with a wide range of interface circuits which match different interconnection strategies for different applications.

3.5 Testing

Because MEMS sensors/actuators deal with environmental rather than electrical properties, their performance can often only be evaluated through sensor-specific tests.

For SoC, it is ideal if macrocells can be self-testing, so that system-level test design can be reduced to sequencing a number of self-tests. Wherever possible, then, interface electronics will need to incorporate the ability to autonomously test sensor operation. Alternatively, each sensor might need to be provided with test code to run on the processor which verifies correct sensor operation.

3.6 Liability

Liability issues will also need to be addressed for MEMS SoC. Since there is probably an even greater chance of independently designed MEMS devices interfering with each other when placed on a single substrate, it is likely that MEMS macrocells will be supplied with limited guarantees about operation in a mixed system. System integrators will often look for a "prime contractor" system house to provide guarantees of correct operation.

4. CONCLUSIONS

To date, MEMS sensors and actuators have not been available as macrocells within a System-on-Chip methodology. As these MEMS devices become more mature and better understood, there will naturally be market pressure to be able to incorporate these devices in a system-on-chip technology.

The following observations are offered about how this might happen.

- Because MEMS-devices tend to be very much tied to a particular fabrication technology, it will be relatively rare for multiple different MEMS devices to be integrated on a single SoC. Rather, the first examples of MEMS SoC will incorporate a single MEMS sensor with other electronics.

- Monolithic MEMS SoCs will have a relatively limited set of available electronics macrocells, and these macrocells will have limited power. One might expect to see the development, for example, of a set of 8-bit microprocessor macrocells which are compatible with mixed MEMS/Microelectronics fabrication technologies.

- The majority of MEMS SoC will be hybrid SoCs. MEMS sensors will be fabricated and tested on individual chips. SoC design will then integrate appropriate interface circuits onto the electronics-only SoC. It is conjectured that the key step in making MEMS technology more available for low cost consumer appliances will be the availability of MEMS-interface macrocells which are compatible with existing SoC technologies

REFERENCES

1. J. Wilson, G. Silcott, N. Peterson, W. Peisel, K.L. Kroeker, “SOCs drive new product development,” *IEEE Computer*, 32 (6), June 1999.
2. N.H.E. Weste and K. Eshraghian, *Principles of CMOS VLSI Design*, Addison-Wesley, 1994.
3. P.J. Ashenden, *The Designer’s Guide to VHDL.* Morgan Kaufmann Publishers, 2001.
4. V. Sagdeo, *The Complete Verilog Book*, Kluwer Academic Publishers, 1998.
5. *The Alba Centre: Driving the Future of Electronic Design* at http://www.albacentre.co.uk/
6. M. Birnbaum and H. Sachs, “How VSIA answers the SOC dilemma,” *IEEE Computer*, 32 (6), June 1999.
7. *VSIA Fact Sheet* at http://www.vsi.org/about.htm
8. T. Shanley, D, Anderson, *PCI System Architecture*, Addison-Wesley, 1999.
9. *AMBA: The de facto Standard for On-Chip Bus* at http://www.arm.com/armwww.ns4/html/AMBA
10. *Standard Test Access Port and Boundary-Scan Architecture*, IEEE Standard 1149, IEEE New York, 1990.