SYSTEM ON CHIP DESIGNING CHALLENGES

Abstract:

System-on-chip has been a nebulous term that mystically holds out a lot of excitement, and has been gaining momentum in the electronics industry. While the potential is huge, the complexities are several, and countering these to offer successful designs is a true engineering challenge.

With rapid advances in semiconductor processing technologies, the density of gates on the die increased in line with what Moore's law predicted. This helped in the realization of more complicated designs on the same IC. Over the last few years, with the advent of bleeding edge technology applications like HDTV and 3rd generation mobile devices, an increasingly evident need has been that of incorporating the traditional microprocessor, memories and peripherals - or in other words the whole system - on single silicon. This is what has marked the beginning of the SoC era.

System on Chip (SoC) design in the forthcoming billion-transistor era will involve the integration of numerous heterogeneous semiconductor intellectual property (IP) blocks. This paper deals with the problems encountered while designing SoC, and possible solutions.

Keywords: *System on Chip, Design criteria.*

Conclusion: This concept may look novel, but in true sense the technology is leading the extent of achieving the System on Chip through ultra Large Scale Integration. It is the semiconductor technology through which we can achieve the concept practically in building a computer on single chip.

INTRODUCTION:

SoC is defined as a single chip solution of an embedded system which may consists of microprocessor(s) and custom logic. SoC solutions are used when performance, size, weight, or power consumption is an issue. Traditionally, the hardware parts of a system have been designed using a CAD tool, and the software parts with programming languages. SoC designs, however, introduce a set of new problems. In addition to the huge costs of a SoC, the debugging and HW/SW integration are also very difficult. This implies that new tools that support joint HW/SW specification, simulation, emulation, prototyping, and debugging, are needed.

The realization of complex Systems on a Chip (SoCs) consisting of billions of transistors fabricated in technologies characterized by 65 nm feature size and less will soon be feasible. Such SoCs imply the seamless integration of numerous IPs performing different functions and operating at different clock frequencies. The integration of several heterogeneous components into a single system gives rise to new challenges. It is critical that infrastructure IP (I^2P) be developed for systematic integration of heterogeneous functional IP blocks to enable widespread use of the SoC design methodology. The five key aspects of a successful design strategy are - Architectural strategy, Validation strategy, DFT Strategy, Synthesis & Backend strategy and Integration strategy.

1. Architecture Strategy

A key dependency that actually defines an SoC architecture is the kind of processor that one uses as the central processing element. The SoC's target application demands may necessitate the inclusion of DSP cores in the design consideration. Another design aspect to be thought through is the kind of processor and system BUS that is to be implemented within the system. The use of standard buses like AMBA AHB (Advanced High Performance bus) and APB (Advanced peripheral bus) is definitely advantageous vis-à-vis the use of proprietary buses. With standard buses, the integration task of putting together the SoC becomes easier, especially if the IP cores that are available support the chosen standard bus protocol.

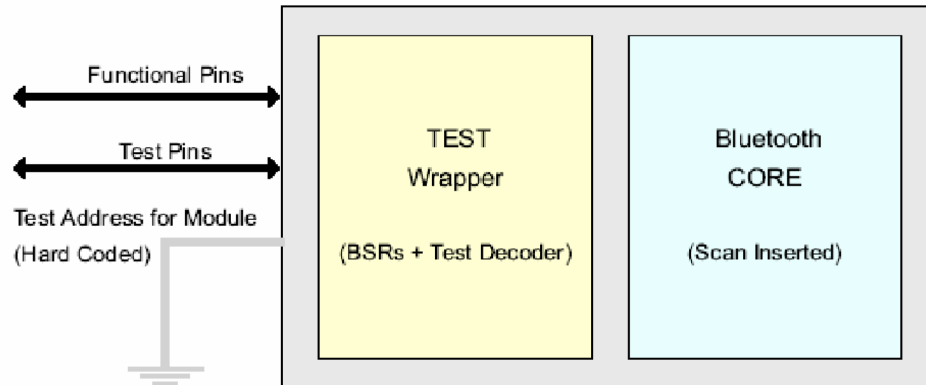
Two key design-for-integration techniques are required to address the challenges in integrating the IpP cores into a SoC

a) Parameterization, which is a key technique for customizing soft cores by using generics or parameters. With the help of this, one can select or eliminate commonly customized features during synthesis like the number of DMA channels in a DMA controller, the number of flash banks in a Flash controller, and so on.

b) Functional partitioning , where use of multiple hard, firm and soft cores is made instead of single monolithic hard cores to eliminate routing problems when the SoC is laid out on silicon. Timing critical components like the CPUs or function critical analog components are usually implemented as hard macros. Functions without any critical timing requirements but which require Intellectual property (IP) encryption fall under the definition of Firm macros. Finally the Functions which require no fixed layout or IP encryption and those which are frequently customized are called soft cores.

2. Design-for-Test Strategy

Verification forms such a crucial aspect of SoC designs, In this
Strategy, most common physical defects are modeled as faults and necessary circuits are included in the design to facilitate checking for these faults. These methodologies and automated processes that insert logic in order to increase the design testability form the essence of DFT.

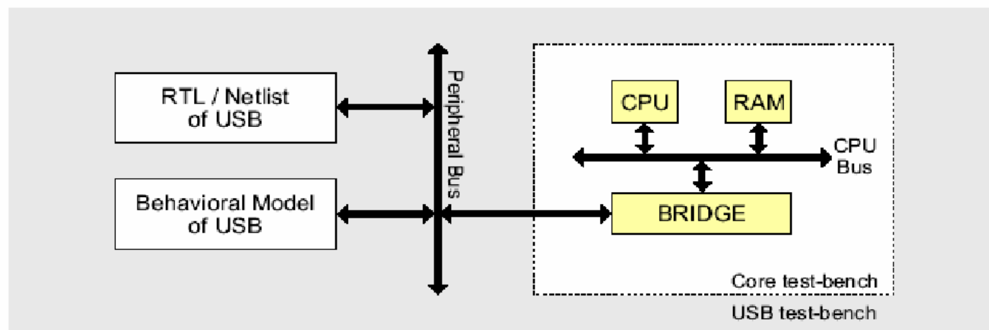


MODULE LEVEL SCAN METHODOLOGY

DFT is implemented using a full scan, muxed flip-flop style of scan insertion. Almost every node in the design is made controllable and observable. Multiple Scan chains are employed, one in each of the modules and these scan chains are stitched at the SoC Device level.

3. Validation Strategy

When several varied functions assemble on a single piece of silicon, verification can become an arduous activity. The verification progresses as the SoC unfolds i.e., at every level of the design flow - RTL, gate level and post layout gate level with timing. As much as 60% of the SoC development time is spent on verification. A high level of code coverage is mandated to determine that the design conforms to the specifications. Also, it is important to cover module testing as exhaustively as system testing is covered.



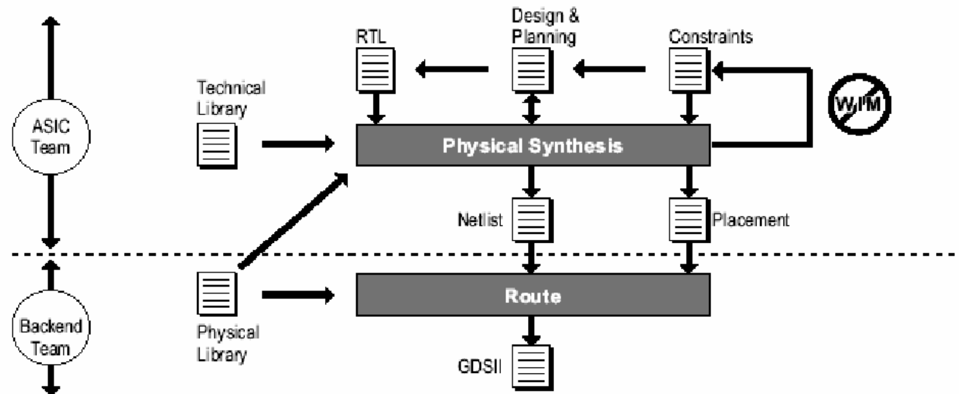
MODULE LEVEL VALIDATION SETUP

There are two issues to be considered when it comes to validating the SoC. First, we need to verify the IP cores thoroughly. Secondly, we need to verify their integration in the system. To meet the first requirement, IP cores, are subjected to extensive IP validation suites that include test benches and behavioral models. The IP's overall functionality is exhaustively verified thus, and the module level verification is effectively translated to the SoC device level by integrating the same test cases in the Test Bench. In order to verify the integration of IP blocks, a separate set of integration test vectors are created, based on typical scenarios. Code coverage and functional coverage tools are employed to trim the test vector set.

4. Synthesis and Backend Strategy

As Silicon processing moves into deep sub micron dimensions, physical effects become more prominent. Some of the common effects are Electro migration, IR drop, Cross Talk, 3D noise, antenna effects and EMI effects. As a result interconnect delays, power, signal

integrity, manufacturability, reliability all become prime objectives along with area and gate delays. Tackling these myriad issues requires chip planning, DFT planning, clocks planning, power planning and timing and area budgeting at a very early stage of the design cycle. The four-week SoC development cycle time, requires an early timing closure. Traditionally, for process geometries as low as 0.35u, standard logic synthesis was considered effective in synthesizing a large ASIC. With the transition to deep sub-micron geometries, painful multiple iterations are required to ensure performance.



PHYSICAL SYNTHESIS FLOW

As it were, the front-end designer creates a net list with limited concept of the physical world and hence performance/area tradeoffs are made with incomplete data. On the other hand the backend engineer has no idea of the designer's intent and hence "Flow creation" becomes more important than design.

5. Integration Strategy

Ultimately, all of the above aspects have to be coherently assembled to install a smooth spin-off flow strategy. This calls for automating the entire SoC Design Cycle in the form of scripts. Beginning with the SOC directory structure creation, extraction of the IP Cores and their validation files, creation of the device top Source file, Simulation, synthesis and ATPG generation, right up to signoff - the whole activity can be automated.

Problems Encountered:

Global wire delays: One of the major problems associated with future SoC designs arises from non-scalable global wire delays. Global wires carry signals across a chip, but these wires typically do not scale in length with technology scaling. Though gate delays scale down with technology, global wire delays typically increase and are controlled by inserting repeaters. Even after repeater insertion the delay may exceed the limit of one clock cycle. In ultra deep submicron processes, 80% or more of the delay of critical paths will be due to interconnect. Another important problem associated with global wires is that such wires are typically implemented in top-level metal layers, with the routing performed automatically in later stages of the design cycle. Such wires end up having parasitic capacitance and resistance that are difficult to predict a priori.

In the forthcoming technologies, global synchronization of all IPs will lose its significance, due to the impossibility of sending signals from one end of the chip to another within a single clock cycle. In fact, many large designs today use FIFO to synchronously propagate data over large distances to overcome this problem. This solution is ad-hoc in nature. Instead of aiming for global control, one attractive option is to allow self synchronous IPs to exchange data with one another through a communication-centric architecture.

Existing on-chip interconnect architectures will give rise to other problems. The most frequently used on-chip interconnect architecture is the shared medium arbitrated bus, where all communicating devices share the same transmission medium. The operating frequency of the shared bus depends on the propagation delay in the interconnection wires. This propagation delay, in turn, depends on the number of IP cores connected to the wires. Each core attached to the bus adds a parasitic capacitance, and therefore, the electrical performance will degrade with system growth. For SoCs consisting of tens or hundreds of IP blocks, such bus-based interconnect architectures lead to propagation delays that exceed acceptable limits as all attached devices share the same wires.
Another major problem encountered is the **On-chip isolation**.

On-chip Isolation: On-chip isolation is becoming increasingly important due to higher integration levels, higher frequencies, and tighter specifications for next generation products. Higher integration not only results in more transistor switching, and thus more noise creation, but it also puts noisy and sensitive components together on the same chip that were on separate chips in the past. At higher frequencies noise now couples more easily from place to place. Isolation provided by wells is reduced and package inductance becomes critical. The package at GHz frequency may cause on chip AC grounds to appear float. When the tighter specifications of next generation products are added, such as 3G cellular, are added to the picture, the RF designer must be both knowledgeable and creative to find an effective solution. An understanding of the impact of process technology, grounding effects, guard rings, shielding, decoupling, and package inductance is necessary to optimize isolation.

Solutions Proposed:

- **Global wire delays:**

Current SoC designs predominantly use shared-medium bus-based functional interconnects to integrate IP blocks. There are mainly three types of commercially used SoC interconnect specification, ARM AMBA bus, Wishbone and IBM
Core Connect. All of them suffer the drawback of nonscalability. There are several architectures available for measuring the global wire delays . A few are Bus based architecture and switch based systems.

Bus-based architecture is, in which the IP blocks are connected to the bus through specialized interfaces called agents. Each core communicates with an agent using the Open Core Protocol (OCP) .Agents communicate with each other using
TDMA (Time Division-Multiple Access) bus access schemes. These agents effectively decouple the IP cores from the communication network. The basic interconnect architecture is still bus-based and will hence suffer from performance degradation trends common for buses.

MIPS Technologies has introduced an on-chip switch integrating IP blocks in a SoC. The switch called *SoC-it* is intended to provide a high-performance link between a MIPS processor and multiple third party IP cores. It is a central switch
Connecting different peripherals, but only in a point-to-point mode.

Kumar [7] and Dally [8] have proposed mesh-based interconnect architectures. These architectures consist of an $m \times n$ mesh of switches interconnecting computational resources (IPs) placed along with the switches. Each switch is thereby connected to four neighboring switches and one IP block. In this case, the number of switches is equal to the number of IPs.

Saastamoinen [9] describes the design of a reusable switch to be used in future SoCs. The interconnect architecture is however not specifically discussed. Guerrier and Greiner [10] proposed the use of a tree based interconnect (SPIN) and addressed system level design issues.

- **On-chip isolation:**

Impact of Process Technology: Most Si based RF chips are fabricated in bipolar, Bi-CMOS, SiGe or CMOS process using a lightly doped bulk-type substrate.
Heavily doped substrates where latch up concerns dominate isolation and the extra wafer cost can be justified. But experienced designers are often able to achieve a few more dB isolation using a lightly doped substrate than a heavily doped substrate .Since lightly doped substrate is highly resistive with typical resistivity of 12 Ohm-cm . The channel stop region at the surface of the chip is approximately three orders of magnitude less resistive than substrate, so breaking the channel stop between two points will increase the isolation between them.

Grounding effects: It is a common practice among analog designers to have separate supplies for analog and digital sections of the chip to isolate the analog circuitry from the switching noise introduced on the digital supplies. The same technique is used to isolate different RF blocks. Dividing a chip into sections with different substrate grounds will attenuate noise coupling from area of chip to another.

Guard Rings: In a lightly doped substrate, as contrasted with a heavily doped substrate, guard rings provide an effective solution. (This is assuming that care has been taken to ensure that the guard ring is connected to a quite supply.)Guard rings around a sensitive circuit help to decouple noise from the circuit and ensure that noise will couple equally into both sides of a differential design. Guard rings around a noise source provide a low resistive path to AC ground for the noise and help minimize the amount of noise injected in to the substrate. The efficiency of the guard rings depends on the noise frequency and the package inductance. The inductive impedance causes the guard ring to “float” and become low impedance conductor for noise.

Shielding: Proper shielding for sensitive signal lines and passive components is an integral part of effective analog/RF IC layout. The challenge is to determine the appropriate shield layer, the bias potential and layout pattern. Sensitive signal buses are often laid out with alternating signal and shield lines to prevent cross talk through lateral and fringing electric fields. To isolate the signal lines from the substrate, the n well or the diffusion layers can be placed under the lines to prevent noise coupling. In general, Shielding increases parasitic capacitance since the field lines from the signal wires terminate at a closer distance on the shielding before they reach another signal line or substrate. The bias potential of the shield should be tied to the reference of the signals. The effectiveness of the shield also depends on the signal operating frequency. There is a tradeoff between a lower shield parasitic capacitance and a higher series resistance. As frequency increases, a large series resistance causes the shield to be ineffective.

On-chip Decoupling: Decoupling capacitance is not always thought of as something that increases the isolation between two points, but it serves the same role. Isolation is desirable to attenuate noise coupling from one portion of the chip to another. If decoupling capacitance can reduce the amount of noise created by supplying local charge for nearby switching and thus lowering the peak current drawn across the package inductance, careful use of it essentially isolates a sensitive circuit that no longer sees the same supply and substrate noise levels.