

# The Use of BEE2 in Addressing SOC Design Challenges (proposal for BEE2 related research)

**Dejan Markovic and Bob Brodersen**

The objective is to provide a methodology for rapid realization of power/area optimized SoC's. The ability to optimize SOC architecture without the need for excessive iterations through the backend flow is essential as designs become more complex. The goal is to integrate algorithmic, circuit, and architectural optimization procedures within an existing commercial backend design flow and also be able to shorten the design cycle. This work will extend our current research in energy-performance SoC optimizations and perform some of the key optimization steps on an FPGA array instead of simulation. An FPGA array based platform developed at the Berkeley Wireless Research Center, currently supporting a TeraOp/sec of real-time performance, will be used to greatly reduce the cost of SOC design and verification.

Following are key issues that will be addressed.

## **1. Design Reentry**

In the process of designing adaptive wireless communication systems, the design cycle traditionally requires reentering a design at various abstraction levels. The process of reentering design data uses a large amount of engineering resources since it typically involves the close interaction of multiple engineering teams spanning algorithms, architecture, and circuits. The situation is further complicated by the fact that these various teams use different design tools to verify their ideas. Algorithm designers use Matlab or C, while hardware designers like to work with RTL based tools from Synopsys or Cadence. This creates the scenario in which the same design needs to be reentered multiple times. The inability to effectively communicate design decisions across the boundaries of algorithms, architecture, and circuits, in addition to incompatibility between the tools, greatly constrains the implementation choices and increases time-to-market.

*Proposed approach:* An environment for FPGA design capture will be used to eliminate design reentry. An algorithm is entered once in a graphical block form, which provides timed data-flow representation of the design and abstract view of design architecture. With technology-specific data for speed, power, and area of functional blocks, algorithm designers can explore the implementation space while remaining in Simulink environment to verify the algorithm. This approach enables power and area optimizations across the boundary of algorithm, system architecture, micro architecture, and circuits, while taking into account unique design issues of deep sub-micron CMOS technology, such as leakage power and process variations. The hardware designer can use Simulink description to generate an HDL description automatically and be able to verify the results of hardware emulation using Simulink test vectors.

## **2. Algorithm Verification**

The complexity of integrated circuits (ICs) in wireless communication devices has been steadily increasing to support more functionality and higher performance. Functionality requirements can be quite drastic, especially in multi-antenna (MIMO) communication systems which use multi-dimensional signal processing algorithms. Due to complexity involved, detailed software simulations can be performed only on small parts of the design.

*Proposed approach:* In order to improve simulation, it is necessary to use hardware-assisted verification. With increasing cost of designing SOC's, design that would require re-spin has become unaffordable. The ability to verify the design in hardware before going to SOC is crucial in reducing this risk and bringing the cost down.

## **3. Word-length Optimization**

Utilization of hardware resources can be greatly improved by optimally selecting finite word-lengths for an algorithm. The challenge is to guarantee correct algorithm behavior while minimizing hardware cost. Simulation based approach is typically used since it enables word-length selection while observing error criteria. A large number of simulations are usually required until the word-lengths converge. For systems of large

complexity, simulation time becomes prohibitively long, or even the memory resources may not be adequate. The goal is therefore to speed up the word-length search procedure by avoiding time and memory intensive simulations.

*Proposed approach:* An approach of mapping the algorithm into an FPGA array to perform simulations in hardware will be used to quickly obtain word-lengths for very complex signal processing systems. The word-length search procedure will consist of 1) range estimation to find maximum and minimum values for each computational node, 2) defining starting word-length values, and 3) iterative search that adjusts word-lengths using a perturbation theory until specifications for a desired level of computational accuracy are met. Simulation-based procedure will be exported onto hardware platform. The main challenge is in avoiding the time consuming re-compile step every time word-lengths need to change in the process of computing sensitivities used in the perturbation theory method. An approach that requires single compilation step with run-time word-length programmability will be developed.

#### **4. Architectural Selection**

Architectural selection is a major time-consuming step in the process of mapping an algorithm to hardware. In a commercial environment, design architecture is typically ported to two or three successive technology generations by simply scaling down device dimensions. Such an approach works well in terms of power and area due to benefits of scaling, but the original architecture would not be the most power/area efficient in the new technology. With scaling of technology, speed-area-power characteristics of the underlying devices change, making the original architecture sub-optimal. In order to achieve the best power/area efficiency, the architecture has to track the energy-performance characteristics of the underlying circuits.

The easiest way to map an algorithm to hardware is by using direct-mapped parallel architecture approach. The direct-mapping often times lead to excessive use of resources, especially in cases where the same algorithmic operation needs to be computed on a multiplicity of independent data streams. Due to vast capabilities of scaled technology in terms of chip area, such inefficiencies could be easily overlooked.

*Proposed approach:* The graphical block-based design entry in Simulink and retargetable design flow (INSECTA) developed at the Berkeley Wireless Research Center will be used to track technology features in the process of architectural selection. The goal is to automate the step of architectural selection.

Starting from the direct-mapped parallel architecture, architectural transformations that minimize power and area will be performed automatically within a high-level block based Simulink environment. Area saving techniques such as time-multiplexing and data-stream interleaving will be studied first. Exploration of a number of algorithms will be needed to better define other commonly used techniques that need to be supported as well. The main challenge will be in retaining functional behavior of the algorithm after the transformations, which may require adding control elements to facilitate data-stream interleaving and proper initialization of the algorithm. As a first step toward full automation, architectural selection will be parameterized to allow exploration of varying degrees of time-multiplexing and interleaving, together with appropriate supply voltage scaling. Additional architectural techniques and higher levels of automation will be gradually developed.

#### **5. SOC Verification**

SOC test and debug can be significantly lengthened by creating test patterns by simulation and using this data for functional verification. Using FPGA to accelerate test procedure may lead to mismatch if SOC design is not fully compatible with the FPGA.

*Proposed approach:* The key is to maintain functional equivalency between FPGA model and an SOC. BEE2 then enables rapid algorithm validation early in the design cycle, real-time functional verification throughout the design process, and also SOC validation at the end. Test vectors generated within Simulink environment can be conveniently exported to BEE2 for SOC test, or generated in real-time on the BEE2, providing a real-time functional verification by hardware co-simulation, thereby greatly improving design productivity.