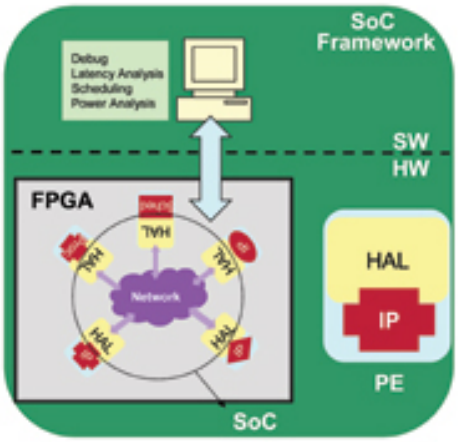
Building an FPGA-based SoC Framework for LTE Baseband Designs

By Akshaya Trivedi, Altera Corporation



Radio Access Networks (RANs) are undergoing an architecture change to an all-IP flat network to reduce network deployment costs as well as to offer rich mobile applications and services at a lower end-to-end latency. Consequently functionality spread across multiple boxes in the network has now primarily moved into the basestation (BTS) box or the eNodeB, making the BTS a virtual super intelligent wireless router. The OFDMA-MIMO LTE PHY functionality is done on the baseband card, also commonly referred to as the channel card. In addition, packet processing functionality from the MAC, Radio Link Control (RLC), Packet Data Convergence Protocol (PDCP), and Radio Resource Control (RRC) layers of LTE processing now also reside on the baseband card, driving the need for higher integration. Although the evolution of process geometry offers further integration, there is a need to support tools and SoC methodologies that assist wireless OEMs to significantly reduce their development, debug and integration time while minimizing their risk.

Wireless operators want to increase their average revenue per user (ARPU). To do this, the mobile operators need to significantly reduce their capital expenditure (CAPEX) and operating expenditure (OPEX) to hit the ARPU targets. As a result, wireless system vendors are faced with the following challenges:

Delivering equipment at significantly lower cost for any given site

* Reducing the number of sites by improving coverage
 * Introducing technologies that reduce OPEX
 * Delivering greener products by dramatically reducing power consumption
 * Providing a comprehensive product portfolio and end-to-end solution to support multiple deployment topologies

* Femtocells, picocells, microcells, and macrocells
* Offering multi-standard support with software upgrades
* Future-proofing architectures for capacity, performance, modifications and new features

Today's 40nm FPGAs and ASICs enable global wireless OEMs meet these mandates by offering high levels of integration. However integrating these complex systems on a chip requires a well-defined and reproducible methodology with a robust set of tools for design, debug and integration. Altera is offering a SoC Framework that is one example of such a methodology.

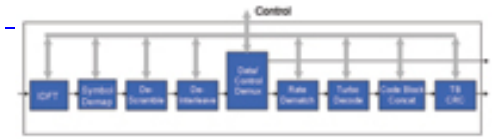
What Does a SoC Framework Provide?

FPGAs today can meet the high performance requirements of multi-sector OFDMA-MIMO systems. Addressing scalability, flexibility, design re-use, ease of debug, test and integration make FPGAs an attractive SoC development platform. The SoC Framework provides:

* A well-defined way to integrate a system
* A software-like design flow
* Early estimation of design performance metrics (Latency, Throughput, Resources, Power)
* Early HW/SW partitioning trade-off and analysis
* Superior system level debugging with a mature proven tool suite

The SoC Framework and Methodology

The SoC Framework for FPGAs and ASICs is comprised of a plug-and-play infrastructure where the system communication is abstracted from the processing elements (PE). A software task scheduler replaces hardware (RTL) state machines and application specific design templates help jumpstart the development providing a time-to-market benefit.



[1]

To successfully develop with an FPGA or HardCopy ASIC, SoC designers must rely on the ability to reliably interconnect existing components, including processors, controllers, and memory arrays, in a plug-and-play fashion. Modelling and developing such systems requires a systematic approach and a framework where the performance metrics are predictable early in the design cycle. As a systematic design methodology, the abstraction of communication network from the processing element (PE) simplifies the overall SoC by designing PEs and communication network independently. A unified messaging between system components provides the hand-shaking mechanism.

Figure 1 illustrates the SoC framework. The software framework incorporates

software models developed for latency and functional verification and software scheduling. The framework hardware consists of an infrastructure that supports heterogeneous SoC. The communication network is abstracted from the PE by the system abstraction layer (SAL). Together with the SAL, network fabric and communication protocol the SoC infrastructure is formed.

The salient features of the SoC Framework and methodology are:

* Software abstraction of control and debug simplifies hardware development
* Soft processing capability on the FPGA also retains flexibility in the ASIC
* Software task scheduler replaces RTL state machines
* SAL provides an API that can be used across global teams
* The SAL API approach provides plug-and-play capability
* Design re-use can be facilitated across global teams on multiple projects
* Add, delete, replicate components based on target requirements
* FPGAs become a platform for multiple projects
* Custom HW/SW partition
* Solution is mapped to the requirement rather than being mapped to available silicon
* Can be done during early system trade-off studies, prototyping stage or even at a later stage
* A well-defined and documented methodology on the interaction between the software task scheduler and the PE
* Identification of system bottlenecks and trade-off analysis for throughput vs. resources
* An early start to system integration even in the absence of certain PE blocks
* Analysis and specification of PEs
* Proven with the LTE PUSCH design template that demonstrates scalability, different number of users and throughput targets.

So How Do I Build a System?

The system can be built in several different ways. A couple of options are explored using the LTE PUSCH uplink bit-rate subsystem as an example:

* Option 1: Using the LTE PUSCH design template as a starting point (see Figure 2)
* Option 2: Using off-the-shelf LTE compliant IP/reference designs

* For example, Mixed-radix IDFT/DFT, CTC Turbo Decoder, Symbol Demapper (SDM), Rate De-matcher (RDM), etc.

In the first option, a PUSCH design is pre-assembled for a target configuration – this is called the design template. Designers can perform systems analysis to determine if the signal chain meets target specifications. They can then choose to replace some PEs, duplicate them or totally replace them based on the requirements. Designers can even add additional blocks that perform other functionality into the SoC framework and modify the task scheduling software to make a custom solution.

In the second option, designers can start from a blank slate and add individual IP

blocks that either are from the library provided by the silicon vendor or from their own internal development. Designers can then incorporate the SoC Framework methodology comprised of the SAL and the software task scheduler to stitch the system together.

Conclusion

Wireless OEMs are faced with challenges to provide a world-class solution for the diverse market needs of mobile operators. To respond to this challenge, there are three key technologies required:

* A broad portfolio of 40-nm transceiver-based FPGAs and ASICs
* Best-in-class software tools with the fastest compile times and other productivity enhancing features
* SoC framework methodology that eases complex system integration, facilitates design re-use and makes system debug easy thereby providing a quality user experience.

Akshaya Trivedi is a senior strategic marketing manager in Altera's Communications Business Unit. He has extensive experience in wireless infrastructure systems design and embedded DSP for wireless, image, video and speech. Mr. Trivedi holds a BSEE from University of California, Irvine and an MSEE from Georgia Institute of Technology.

Links:
[1] http://www.wirelessdesignmag.com/sites/wirelessdesignmag.com/files/legacyimages/1002/Altera2_lrg.jpg