

# Using configurable processors for high-efficiency multiple-processor systems

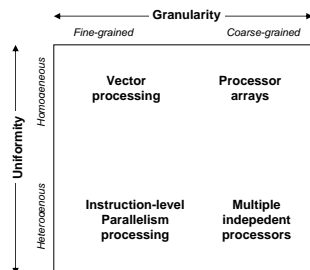
Chris Rowen  
 Tensilica, Inc.  
 3255-6 Scott Blvd  
 Santa Clara, CA 95054  
 +1 408 986 8000  
 rowen@tensilica.com

## ABSTRACT

This talk focuses the heart of the embedded system design challenge: How can you use multiple processors to efficiently implement complex concurrent system functions? To build this type of parallel system, the designer must work both at the architectural level—identifying the best partitioning and communication among sub-systems—and at the implementation level—creating the right interfaces, task software, and processor definitions to realize an efficient parallel system.

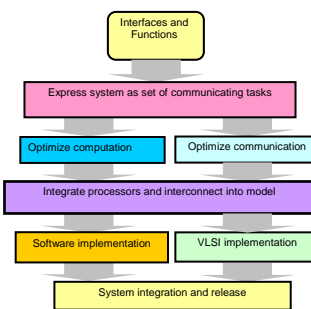
The parallel architecture challenge is three-fold: to take full advantage of the obvious concurrency between different independent chip functions, to identify and extract concurrency within tasks, and to simply express the communications as abstract tasks that enables efficient and scalable system performance. Similarly, the implementation challenge has three components: to translate the abstract communication into an efficient processor code implementation, to generate appropriate interconnect hardware supporting communication among processors (and memories), and to configure the processors with the right interfaces, memories and instruction set to allow low-overhead communication and efficient computation.

The combined hardware and software view explicitly builds connections between the chip designers concerns—chip cost, power dissipation, clock frequency, design hierarchy, and logical verification—and the software architect’s concerns—programming model, development tools, application throughput, real-time system response, code and data footprint, and upgradeability.

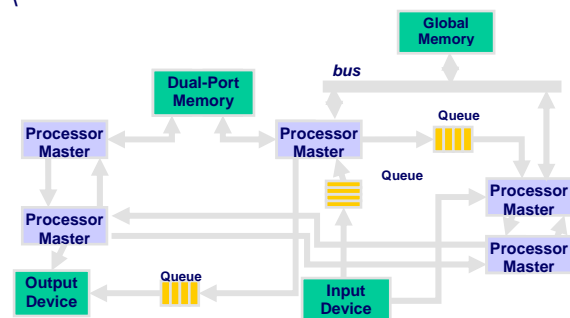


We can categorize forms of concurrency by looking at two workload characteristics: the granularity (scale at which we recognize that operations are independent) and the uniformity (the similarity of sets of independent operations) of the concurrency. Concurrency appears both at the fine-grained level of individual data operations—where basic operations on the data (loads, adds, compares) are independent of one another and can be performed at the same time, and at the coarse-grained level where entire procedures are independent of one another and can

execute at the same time without respect to ordering of operations between the two. Configurable processors form an essential building block for efficient multiple-processor system-on-chip design. Their flexibility in both communications interfaces and computation efficiency enables a new system design flow.



The multiprocessor design flow starts from the high-level requirements, especially the external input and output requirements for the new SOC platform and the set of tasks that the system performs on the data flowing through the system. The processors themselves are extended to fit the task computation profile. The communication software and interconnect hardware between processors are optimized based on communications traffic profiling. The tools of the flow create an accurate system model available early in the design schedule, so detailed VLSI and software implementations can proceed in parallel. Early and accurate modeling of both hardware and software reduces development time and expensive surprises late in the design cycle.



The computational extensibility of configurable processors allow a larger fraction of all chip functions execute efficiently as processors. These processors’ interface adaptability also creates a much wider range of feasible multiple processor topologies, combining shared buses, multi-ported memories, and point-to-point message-passing queues.