Enabling Communications by Enhancing the Multicore Ecosystem

Sven Brehmer
President and CEO
PolyCore Software
639 Greenwich Ln,
Foster City, CA 94404
+1 (650)-570-5942
sven@polycoresoftware.com

Kumar Venkatramani
Sr. Technical Consultant
SoftJin Technologies
2900 Gordon Ave,
Santa Clara, CA 95051
+1 (408)-773-1714
kv@softjin.com

ABSTRACT

The recently released ITRS 2005 Roadmap states that by year 2013, about 22% of a design will be done by an asynchronous global signaling technique. The same roadmap also states in 2005, software accounts for 80% of a system’s development cost.

Add to this, a growing trend to combine homogenous or heterogeneous cores on a single SOC, whether it is for reuse purposes or increased throughput, on-chip communication and the software associated with the communication (both drivers as well as virtualization) that need to manage the flow of control and data through the chip is becoming a crucial design step.

Addressing the challenge of harnessing this processing capability in a way that supports interoperability, and thus helps speed time to market is the Multicore Association's mission. This presentation will discuss the current status of the standards being implemented within the consortium's various working groups: Communications API (CAPI), Resource Management API (RAPI), Transparent Inter Process Communications (TIPC), and Debug all from within the context of embedded distributed systems

CAPI serves to capture the basic elements of communication and synchronization that are required for embedded distributed systems. RAPI will provide a standardized interface for the management of system resources on multi core architectures. This includes, but is not limited to the scheduling, and synchronization of work entities amongst cores or within cores and the management of non-uniform memory allocation and de-allocation. This presentation describes the RAPI, how it relates to CAPI, and how it delivers benefits to the development of multi core systems. The TIPC protocol is specially designed for intra cluster communication. TIPC's process will define higher-performance communication paths. Developers who need to debug applications running on multi core systems face numerous challenges, including: problems that require debugging multiple simultaneous threads of execution, such as synchronization bugs; differing user experiences for heterogeneous elements of a system; and mapping between the programming model, such as higher-level communication constructs, and the lower-level implementation, such as memory and registers. The Multicore Association is currently investigating how to help the industry provide better tools for application developers.