SYNOPSYS®
Silicon to Software™

Synopsys Hybrid Prototyping Solution

- Integrates Virtual and FPGA-Based Prototypes
- Start multicore SoC prototyping earlier
- Achieve high-performance execution of system-level models with real-world interface connectivity of hardware
- Partition SoC design to maximize overall performance
- Accelerate system prototype bring-up
- Improve software debug visibility
- Easily integrate ARM Cortex Processor models, AMBA protocol transactors and DesignWare IP into a single hybrid prototype

Start Multicore SoC Prototyping Earlier

Today, designers use two relatively well-established, independent methods for SoC prototyping: virtual prototypes or FPGA-based prototypes.

Virtual prototypes are fast, fully functional SystemC/Transaction-Level Model (TLM)-based models of SoCs under development that execute unmodified production code. Because of unparalleled software debug and analysis efficiency, virtual prototyping accelerates pre-RTL embedded software development, hardware/software integration, and system validation.

FPGA-based prototyping accelerates the creation of an ASIC prototype with high-speed hardware prototyping systems including a software flow for the conversion of ASIC RTL into one or more FPGA devices. FPGA-based prototypes provide cycle-accurate, near real time performance, and real world interface connectivity prior to availability of first SoC silicon.

Both virtual and FPGA-based prototyping approaches have unique benefits to the development team, but mixing SystemC and RTL model abstractions can be difficult. To accelerate time-to-market, more advanced ASIC prototyping techniques, such as a hybrid prototype are being adopted to enable designers to start multicore SoC prototyping earlier. Hybrid prototyping blends the best of what virtual and FPGA-based prototyping have to offer into a single environment.

Hybrid Prototypes Deliver the Best of Both Virtual and FPGA-Based Prototyping Approaches

The Synopsys hybrid prototyping solution combines virtual prototyping and FPGA-based prototyping to accelerate the development of SoC prototypes. By using Virtualizer™ virtual prototyping for new design functions and HAPS® FPGA-based prototyping for reused logic and new RTL, designers can start software development up to 12 months earlier in the design cycle. In addition, Synopsys' hybrid prototyping solution enables designers to accelerate hardware/software integration and full system validation, thus reducing the overall product design cycle.

Synopsys' hybrid prototyping solution enhances software stack validation through high-speed processor execution using abstract SystemC models. It allows connection to real-world I/O model interfaces via analog PHYs or test equipment attached to a HAPS FPGA-based prototype. In addition, designers can leverage new and existing RTL or IP in the FPGA-based prototype and pre-RTL portions of the design in SystemC transaction-level models (TLM), which are faster to implement and available much sooner in a project lifecycle.

> *"The rising complexity and software content associated with multicore SoCs means that systems engineers and software developers cannot wait for hardware to begin their work, so they increasingly utilize prototypes of their chips and systems. Synopsys' 'hybrid' approach allows developers to freely mix pre-RTL transaction-level models with RTL that already exists, giving design teams a significant head start on their hardware and software development."*
>
> **Chris Rommel**
> Vice President, Embedded Software and Hardware VDC Research

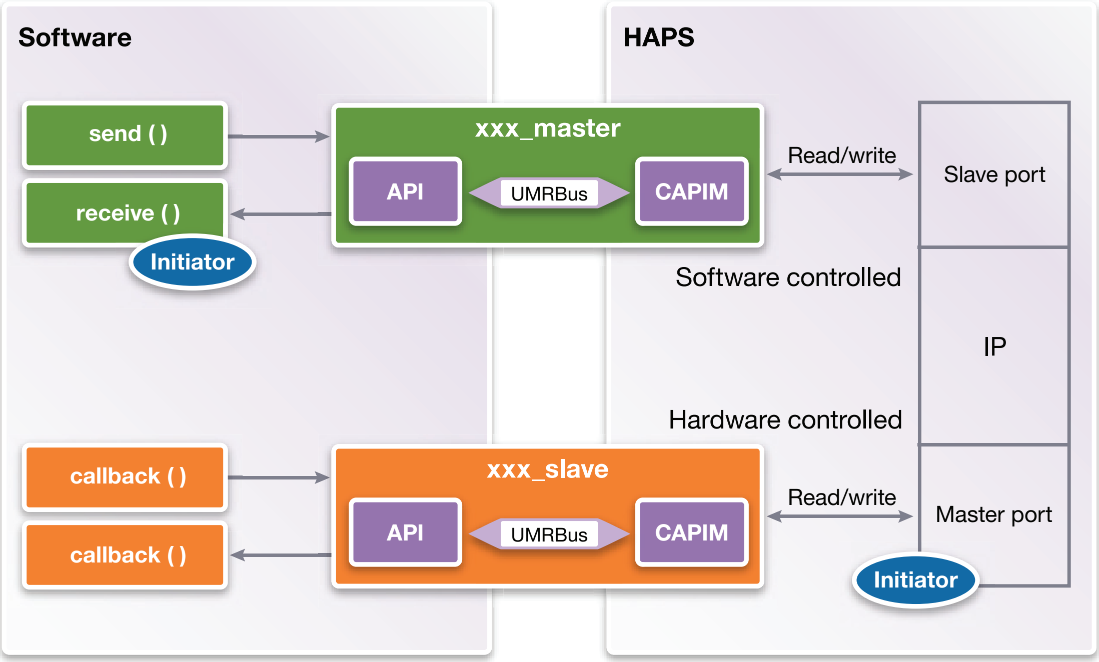

Figure 1: Transactors seamlessly translate between TLM and pin-level hardware events

Efficient Data Transfer Between Virtual and FPGA Environments

A hybrid prototype requires both a logical and physical link to combine the virtual and FPGA-based prototypes.

For the logical link, the Synopsys Transactor Library for the ARM® AMBA® protocol enables the data exchange between SystemC/TLM models executed by Synopsys' Virtualizer virtual prototyping and a HAPS Series FPGA-based prototyping system. In addition, a general purpose C++ transactor library with an application programming interface (API) is available for data streaming applications or integration between a HAPS Series system and custom C++ environments. The Synopsys Transactor Library supports a variety of ARM AMBA protocols, including AMBA 2.0 AHB™/APB™, AXI3™, AXI4™ and AXI4-Lite™.

The ARM AMBA master components can initiate a transaction from either the Virtualizer virtual prototyping or HAPS FPGA-based prototyping environments. In the virtual environment, the software API provides send, receive, and callback functions to communicate with the hardware domain. In the hardware context, the IP blocks communicate through the slave and master ports connected to the user IP.

The transactor scheduler supports de-synchronized operation of the loosely-timed model abstractions with cycle-accurate FPGA-based hardware for best overall system performance.

To ease physical connections, the HAPS Universal Multi-Resource Bus (UMRBus) interface kit includes a PCI Express® peripheral card for the host workstation and an interface pod for a HAPS Series system.

Naturally Partition With a Hybrid Approach

Consider the challenges of prototyping the individual components of a complex SoC design. While any block within a system could target a virtual or FPGA-based prototype, the modeling and integration effort can vary dramatically. Prototype attributes such as capacity, effort to author new blocks and real world I/O access can limit the degree of hardware/software development that can be accomplished with prototyping.

In the example SoC design illustrated in Figure 2, the CPU, level 1 and 2 cache memories as well as the display controllers are good candidates for a virtual approach since their models are easier and quicker to obtain than RTL and have excellent throughput for software execution. However, some of the peripherals such as the graphics engine and the radio interfaces for GPS and WiFi, lend themselves to prototyping in an FPGA, where test patterns and analog PHYs are easier to connect for sake of system validation. FPGA-based prototyping also has advantages when cycle-accurate performance and model fidelity are important.

Because of the advantages of hybrid prototyping, some design teams have taken on the difficult task of writing custom interfaces to connect the FPGA-based and virtual prototypes. This effort is often time consuming and error-prone. Synopsys' hybrid prototyping solution enables rapid deployment of full-featured hybrid prototypes.

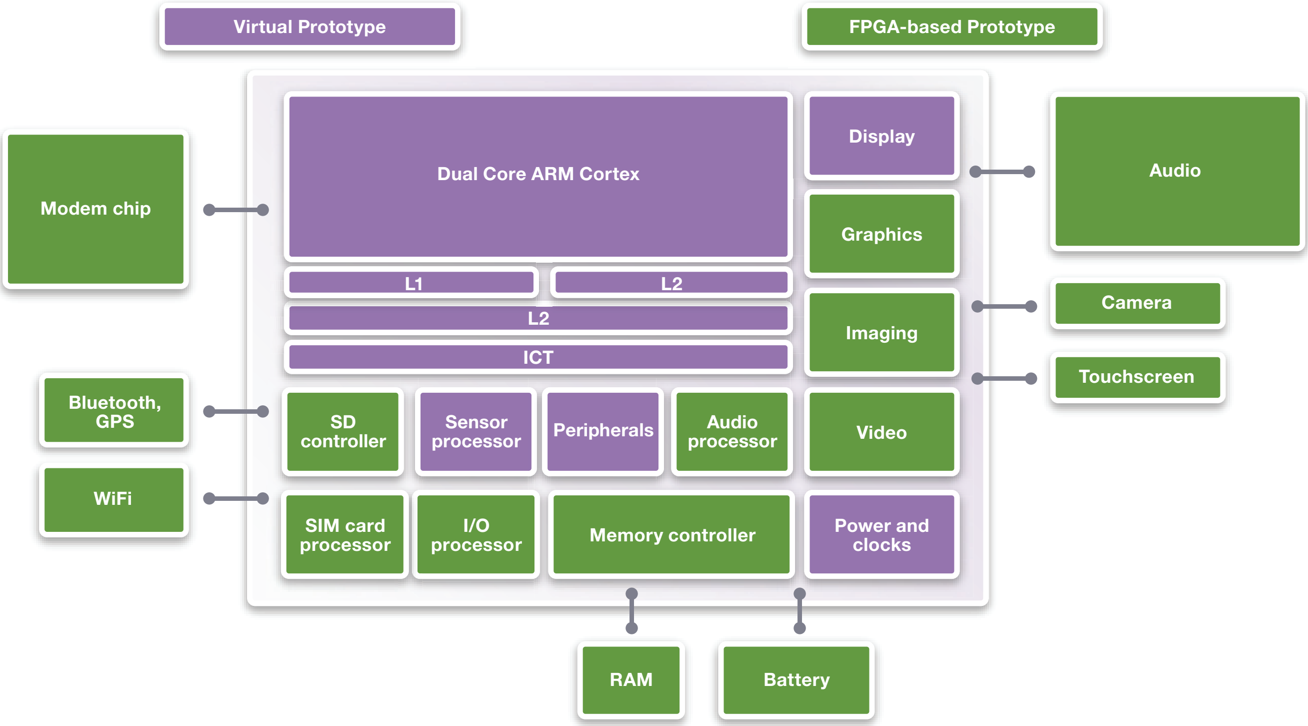

Figure 2: An example of SoC block partitioning between virtual and FPGA-based prototyping environments

Strength of HAPS FPGA-Based Prototyping

The hardware environment of the hybrid prototyping solution is based on Synopsys' leading HAPS family of FPGA-based prototyping products.

The HAPS Series of prototyping systems is designed to support all ASIC prototyping needs, including hardware/software co-development, proof-of concept studies, IP development and end-user evaluations. The flexibility of the HAPS Series allows the same system to be reused in several projects and/or various configurations by adding daughter boards containing I/O and custom subsystems.

The HAPS FPGA-based prototyping solutions consist of a suite of modular, easy-to-use products for ASIC prototyping that include HAPS hardware systems supported by an integrated tool flow with ProtoCompiler design automation and debug software.

The advantages of hybrid prototyping for the FPGA prototype designer include:

- Earlier bring-up: Start FPGA-based prototype before all RTL is available
- Faster utilizing a high performance CPU models
- Greater debug visibility of software under development
- Easier to validate IP in an SoC context

Strength of Virtualizer Virtual Prototyping

The virtual environment of the Synopsys hybrid prototyping solution is based on Synopsys' comprehensive Virtualizer virtual prototyping solution.

Virtualizer-based prototypes are used for software-driven verification, embedded software development, and integration and test. They are also used to improve communication within the supply chain because of the ease with which they can be deployed to hardware and software developers, regardless of location.

The Synopsys virtual prototyping solution offers a broad portfolio of models and reference designs, intuitive graphical prototype assembly and debug, and standards-based (SystemC/TLM) support and publishing capabilities.

Debug and analysis efficiency is increased through embedded system-level software debug and analysis tools as well as ease of integration with 3rd party software debuggers and other popular embedded software development tools.

Specific advantages of hybrid prototypes for the virtual prototype designer include:

- Reduced modeling effort: Bring in existing RTL, IP or subsystem earlier
- Quickly incorporate real-world I/O
- Model cycle-accurate hardware execution engines

Hybrid Prototyping Combines the Best of Both Worlds

Synopsys' hybrid prototyping solution enhances software stack validation through very high-speed execution of processors using a Virtualizer virtual prototype. It allows connection to real-world I/O model interfaces through analog PHYs or test equipment attached to a HAPS FPGA-based prototype. In addition, it allows ultimate flexibility to mix and match model abstractions to leverage legacy RTL with SystemC/TLM models that are faster to implement and available sooner in a project lifecycle.

The Synopsys hybrid prototyping solution is comprised of the following Synopsys products:

- HAPS-80, HAPS-70, or HAPS-DX Series System
- ProtoCompiler for fast implementation of FPGA-based prototypes
- Optional: Virtualizer (for SystemC support)
- Optional: VCS-MX (for co-simulation support)

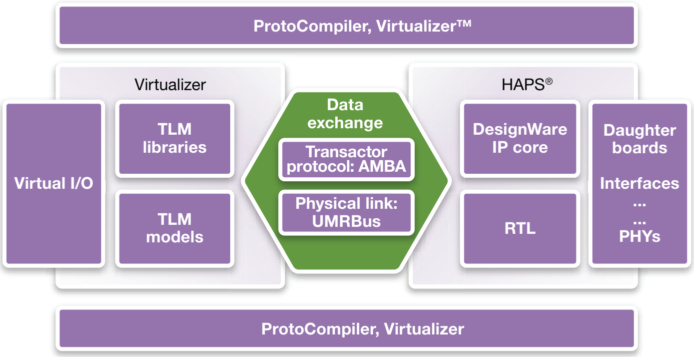

Figure 3: Synopsys hybrid prototyping solution seamlessly integrates Virtualizer virtual prototyping and HAPS FPGA-based prototyping

Synopsys, Inc. • **690 East Middlefield Road** • **Mountain View, CA 94043** • **www.synopsys.com**

08/15.RP.CS6368.