ESL Design Methodology in the Co-verification of an eMMC/SD system

By
Dennis McCarty

February 16, 2011

Arasan
The Bus Stops Here

Overview

Designers of complex systems that use software to control and configure hardware need a means of co-verifying the hardware and software operating simultaneously in order to discover errors that do not appear when simulating the hardware or software separately.

The Electronic System Level (ESL) design methodology focuses on the higher abstraction level and allows designers a means of system verification.

ESL is used to model the behavior of the entire system using a high-level language such as C++. Rapid implementation of the system can be automated using EDA tools such as high-level synthesis and embedded software tools, although much of it is performed manually today.

ESL can also be accomplished through the use of SystemC as an abstract modeling language. Electronic System Level is now an established approach that is used at most of the world's leading System-on-a-chip (SoC) designs companies. It is also being used increasingly in system design.

From its genesis as an algorithm modeling methodology with 'no links to implementation', ESL is evolving into a set of complementary methodologies that enable embedded system design, verification, and debugging through to the hardware and software implementation of custom SoC, system-on-FPGA, system-on-board, and entire multi-board systems.

ESL Verification

Co-verification is a means of verifying system architectures and of ensuring higher levels of RTL verification. Designers who wish to perform co-verification using ESL models must obtain a model of their system. Logic that is written in Verilog may be converted into SystemC for verification, but cannot be converted back to Verilog reliable when verification is complete. Moreover, one is at the mercy of the conversion tool to ensure the SystemC version is correct.

IP cores that have been licensed also suffer from the conversion problems. The solution is to license a SystemC version of the IP from the vendor.

ESL in the Arasan Total Solution

Arasan is breaking new ground offering ESL Models in SystemC that complement and allow co-verification of its digital and software IP products. The ESL models are combined with the software stacks and core IP products to perform Transaction Level Modeling using SystemC simulator. The high-level the software and hardware allows architectural exploration, performance modeling, independent software development, functional verification, and high-level synthesis. The models are part of the Total Solution offered by the company for implementing standards.

Arasan ESL IP Portfolio

The Arasan eMMC/SD ESL model combines SD2.0/SDIO2.0/eMMC4.41 Host Controller and an eMMC/SD card controller with a GREENSOCS bus interface at system side.

The product confirms to SD Host Controller Standard Specification Version 2.0 and eMMC standard specification version 4.3.

eMMC/SD Host in ESL

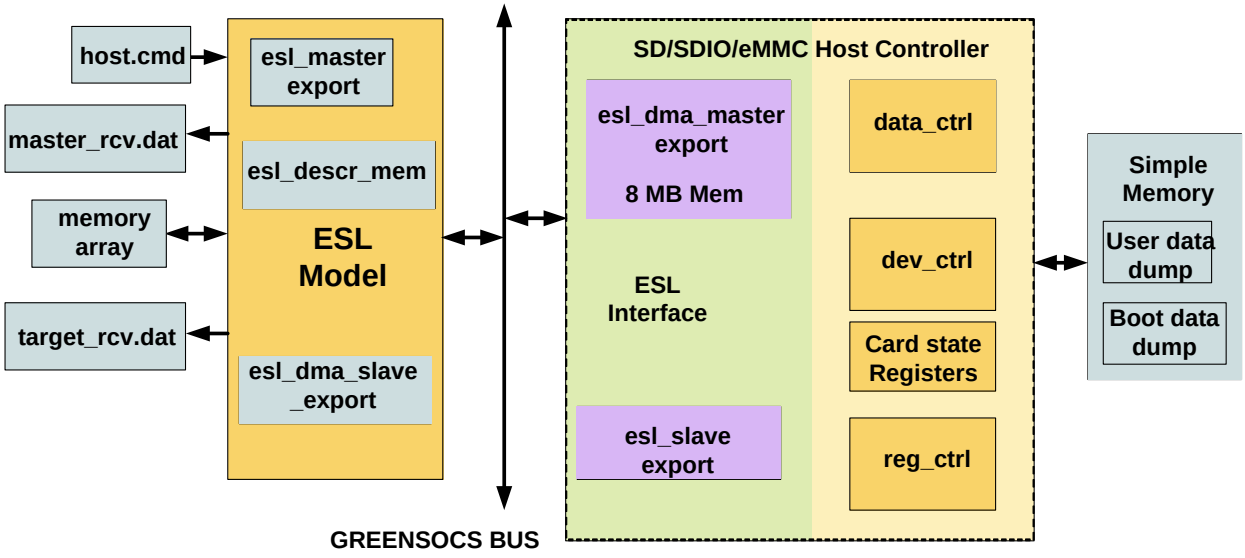

Figure 1: SD Host Simulation Environment

The eMMC/SD Host ESL is a high-level functional model of the SD/SDIO/eMMC Host RTL as shown in Figure 1. The model operates asynchronously and uses no clock. The model runs on the GREENSOCS simulator. The Host model consists of three main sections including the Host Controller itself, a Simple Memory that serves as the repository of SD device data transfers and the ESL model which is the test bench for the Host simulation.

The ESL Model is analogous to a test bench in that it reads commands from the host command (host.cmd) file and applies them Host model. Other files accessed by the ESL Model include the target receive (target_rcv.dat) file which stores data from target read commands and the master receive (master_rcv.dat) file that holds the data from write commands.

Elements in the ESL Model include the ESL descriptor memory (esl.descr_mem) block which is a memory that stores descriptors for virtual address translations. The ESL master export (esl_master_export) block reads the Host command file to execute the commands. It initiates Device data transfers.

The ESL DMA slave export (esl_dma_slave_export) block emulates the DMA in the Host controller that transfers data to the processor.

The Host model has a register control (reg_cntl) that emulates the configuration registers in the Host controller. These define the controller operation and are accessed by the processor. The Card State registers are read by the processor so that it will know the current state of the Host

controller. The data control (data_ctrl) block drives the command control signals to the ESL DMA Master Export and Register Control blocks. The device control (dev_ctrl) accesses the Simple Memory with Device data and holds Device-specific state and card information.

The block labeled as Simple Memory is actually acts as the SD Device data storage in the system. It has a User data dump that stores transfer data. The memory also has a Boot data dump that stores code for bootstrap loading.

Operations

The Host Controller provides Programmed IO and DMA data transfers. In programmed IO, the ARM processor transfers data using the Buffer Data Port register. Host controller support for DMA can be determined by checking the DMA support flag in the Capabilities register. DMA allows a peripheral to read or write memory without the intervention from the CPU.

The system address register points to the first data address and data is then accessed sequentially from that address.

PIO data transfers are performed on the ESL Slave interface and DMA transfers use the ESL Master interface.

The eMMC/SD ESL model interfaces with the ESL Driver model through two GREENSOCS interfaces

The ESL Slave Register interface ESL Driver model can read or write the eMMC/SD Host controller registers through the ESL Slave Register interface. The interrupt signal can be generated to the Driver model when any of the interrupt status bit is set.

The ESL Master DMA interface is used to read/ write the data from/ to system memory through ESL Master DMA interface. The interface performs DMA operations with the system memory and serves as the initiator of the GREENSOCS transactions. Register control and data control modules are control the ESL DMA Master.

Summary

ESD modeling offers designers a means of co-verifying hardware and software systems speeding both hardware and software development and debug. The Arasan ESD eMMC/SD model was used by a major semiconductor vendor to verify a full-custom SoC design. The product is but the first of the ESD models of its IP products to be offered by Arasan.

Fax: 408-282-7800
Email: sales@arasan.com

IP products, please visit: www.arasan.com