A SYSTEM-ON-CHIP COURSE USING ALTERA'S EXCALIBUR DEVICE AND QUARTUS II SOFTWARE

A. Bindal, *Member, IEEE* and S. Mann

ahmet.bindal@sjsu.edu

***Abstract*—**The authors have developed a senior-level undergraduate System-on-Chip (SoC) course at San Jose State University that emphasizes SoC design methods and hardware-software co-design techniques. The course uses a "real world" design project as the teaching vehicle, and it implements a SoC platform to control a five-axis robotic arm using Altera's state-of-the-art Excalibur chip. The Excalibur chip contains both an embedded ARM processor and a Programmable Logic Device (PLD) array. The course goes through a complete hardware-software co-design flow from implementing custom hardware devices on a PLD to developing an embedded algorithm in a state-of-the-art design environment for a complete SoC solution. Students are introduced to "real world" design methods by acquiring their design data from the Internet in the form of current data sheets and application notes and using them in conjunction with the Altera EPXA1 board and Quartus II design environment to create their design. Students learn to use the design environment by examining the sample design files in the development kit and following the step-by-step instructions towards creating a simple embedded application. After this familiarization step, students define the architectural specifications of a memory-mapped servo controller, implement it in the Excalibur's PLD array and interface this device with the ARM processor's internal bus (AMBA) in order to control each robotic arm servo. Functional regression tests and post-synthesis timing verification steps are applied to the servo controller following the implementation phase. Subsequently, students integrate the servo controller with the rest of the system and perform board-level functional verification tests to observe whether or not the robotic arm can move an object from a source to a destination point accurately. Students also develop an embedded algorithm, which translates user inputs in Cartesian coordinates into robotic arm movements in spherical coordinates during laboratory sessions.