MPhil in Advanced Computer Science
System On Chip Design and Modelling

Leader:	Dr David Greaves
Timing:	Lent
Prerequisites:	Some experience with VHDL/Verilog RTL, Assembler, C++
Structure:	8 lectures + 8 practicals/coursework: and research project option.

AIMS

A current-day system on a chip (SoC) consists of several different microprocessor subsystems together with memories and I/O interfaces. This course covers SoC design and modelling techniques with emphasis on architectural exploration, assertion-driven design and the concurrent development of hardware and embedded software. This is the 'front end' of the design automation tool chain. (Back end material, such as design of individual gates, layout, routing and fabrication of silicon chips is not covered.)

SYLLABUS

All candidates must have a basic knowledge of programming, digital hardware and assembly language programming. Experience with C++ is also highly useful.

1. **Verilog RTL Design with examples.** Basic RTL to gates synthesis algorithm.

2. **Further examples.** Event-driven simulation cycle. Using signals, variables and transactions for component inter-communication.

3. **SystemC overview.** Verilog synthesis and high/low-level mapping examples.

4. **High-level modelling in SystemC.** Bus and cache structures, DRAM interface. Design exploration.

5. **Transactional modelling (ESL).** Electronic systems level design. IP-XACT.

6. **Processor Modelling.** Instruction set simulators, cache modelling and hybrid models.

7. **Assertions and Monitors.** System Verilog brief tour. PSL/SVA assertions. Temporal logic compilation to FSM. Assertion-based design.

8. **On Chip Interconnect.** Busses (OPB (BVCI) and AXI). Glue logic synthesis. Transactor Synthesis. Network on chip.

OBJECTIVES

On completion of this module students should:

- be familiar with how complex gadgets, containing multiple processors, such as an iPod or a sat-nav is designed and developed.
- understand the hardware and software structures used to implement and model inter-component communication in such devices,
- be familiar with SystemC and PSL assertions.

COURSEWORK

The students will attend eight afternoon sessions where they first repeat demos developed in the lectures and later develop their own version of a component in a larger, class-based project. A collaborative element is that the blocks developed by each student can be used as different parts of a global project, held in a central workspace. If there are insufficient students for a good class-based project, we will arrange integration with project work in related M Phil modules.

RESEARCH PROJECT

The coursework itself is largely practical project work, but to take this further, there are a number of open-source projects where a student can expect to make a contribution. A typical example might be the recreation of an early microcomputer, including most of its peripherals. Practical work could instead involve interfacing a variety of peripherals to FPGA cards, including car-area-network devices to gain experience with industrial command and control. Alternatively, the project might implement a logic synthesis algorithm recently reported in the literature.

ASSESSMENT

- The lectured component will be formally reviewed with the student using a tick system, that ensures all the associated exercises are completed (25%).
- The course work should be written up and put on a private or public web site in the form of a hypertext dissertation with downloadable examples (75%).

RECOMMENDED READING

'*System Design with SystemC*', Springer. Grotket, Liao, Martin and Swan.
'*SystemC tutorials and whitepapers.*' Download from OSCI www.systemc.org
'*Essential Issues in SoC Design: Designing Complex Systems-on-chip*', Spinger. Youn-Long Steve Lin (Editor).

Last updated: January 2009