Course guide
2301216 - AIPCD - Advanced IP Core Design

Last modified: 04/04/2024

Unit in charge: Barcelona School of Telecommunications Engineering
Teaching unit: 1022 - UAB - (ANG) pendent.

Degree: MASTER'S DEGREE IN SEMICONDUCTOR ENGINEERING AND MICROELECTRONIC DESIGN (Syllabus 2024). (Optional subject).

Academic year: 2024 **ECTS Credits:** 4.0 **Languages:** English

LECTURER

Coordinating lecturer: Castells Rufas, David

Others: Moretó Planas, Miquel
Genovese, Ignacio

PRIOR SKILLS

Digital design based on an RTL-level hardware description language (VHDL, Verilog, ...).
Design and simulation of basic digital systems: combinational and sequential logic functions, arithmetic functions and finite state machines.
Implementation and debugging of basic digital systems on reconfigurable devices (FPGAs).
Development of software applications based on a microprocessor/microcontroller.
C/C++, Python programming language.

TEACHING METHODOLOGY

Lectures.
Laboratory practical work.
Individual work (distance).
Final Exam.

LEARNING OBJECTIVES OF THE SUBJECT

Identify different repertoires of instructions and select the most appropriate to implement a system.
Be able to estimate the parallelism potential of algorithms to guide platform selection or design.
Identify performance and energy bottlenecks.
Be able to create specific hardware (IP Core) with quality assurance methods that allow its easy integration in systems.
Be able to develop software code optimized for the best use of the IP cores to minimize execution time and energy consumption.
Learn to design a computer system combining available complex IPs (such as processors, memories, coprocessors, etc.) to undertake a specific function.
Acquire knowledge and hands on practice on FPGA prototyping and software-based verification methods.

STUDY LOAD

Type	Hours	Percentage
Hours small group	12,0	12.00
Self study	70,0	70.00
Hours large group	18,0	18.00

Total learning time: 100 h

CONTENTS

Types of IP Cores

Description:
Processor IP cores (microprocessors, microcontrollers, DSPs). Memories, RAM (SRAM, SDRAM, …), ROM, Cache, NVM. Memory Controllers. Communication Interfaces. Point to Point Interfaces (Serial/Parallel, Unidirectional/Bidirectional) Multi-endpoint Interfaces (multi-master, multi-slave).
Memory Mapped interfaces (Buses, AXI), Networks on Chip. Integration of interface IP cores in electronic systems (Generators, Platform Builder, ….). Specialized Cores. (cryptographic, image processing, …). Customization and adaptation of specialized cores.

Full-or-part-time: 2h
Theory classes: 2h

IP Core Verification

Description:
Importance of Verification. Ensuring correctness, reliability, and functionality of IP cores. The verification process (flow from design to testing). Systematic testing methodologies. Verification Methodologies. Simulation-Based Verification. Event-driven simulation vs. cycle-based simulation (QuestaSIM vs Verilator). Formal Verification. UVM (testbenches, agents, sequences, …). Cocotb, CI/CD (Flows). FPGA Emulation. Real-World Case Studies: Lessons learned, and challenges faced in real-world projects.

Full-or-part-time: 2h
Theory classes: 2h

Metrics

Description:
Importance of Metrics. Types of Metrics (area, performance, energy consumption). Definition of Area: chip area, gate count, resource utilization. Methods to measure area. Tools (Synthesis…) Performance Metrics. Delay (setup time, hold time, propagation delay), and Fmax. Execution Time. Bandwidth. Roof-Line Model. Tools (Profiling). Relation with Simulators (Spike, Gem5). Energy Consumption Metrics. Dynamic Power and Static Power. Measuring Energy Consumption. Energy Optimization (Clock Gating, Power Gating, DVFS). Tools (Simulation, Power monitoring).

Full-or-part-time: 2h
Theory classes: 2h

Case Study: A Vector Unit as a Complex IP Core

Description:
Funcionamiento funcional de una Unidad Vectorial. El concepto de procesamiento de vectores. ¿Cómo conectar la VPU a un procesador? (interfaz, jerarquía de memoria). Arquitectura de una Unidad Vectorial. Registros vectoriales, unidades funcionales vectoriales (carriles), registro de longitud vectorial, etc. Optimizaciones del compilador para procesamiento de vectores (vectorización). Análisis de VPU: Área, Rendimiento, Compensaciones energéticas. Verificación. Un verdadero ejemplo.

Full-or-part-time: 2h
Theory classes: 2h

Extending Processors with Custom Hardware: Custom Instructions

Description:
Algorithmic Machines. Datapaths. The trade-off between computing and storage resources. Estimating Performance Benefits. Reviewing Profiling. Identifying Candidate instructions for Hardware Implementation. Custom Instructions and Processor Architecture Extensions. Functional Units for Custom Instructions. Additional integration requirements (decoding stage). Extending RISC-V processors. HDL development. Functional Validation (Extending Spike). Performance Estimation (Extending ISS, Gem5). Synthesis. Emulation. Examples (Hamming Distance, Color Space Conversion).

Full-or-part-time: 2h
Theory classes: 2h

Memory hierarchy and devices

Description:
Memory Hierarchy and Cache Basics. Levels of memory hierarchy: registers, cache, main memory, secondary storage. Types of caches (L1, L2, L3). Cache Architectures. Components (tag, index, block offset). Operation. read and write mechanisms, cache hit and miss. Mapping; direct-mapped, set-associative, fully associative. Cache Replacement Policies (LRU, FIFO, Random, ...). Cache Coherency. Problems of incoherency. Cache coherence in multi-cores. Overview of coherence protocols like MESI, MOESI, and directory-based approaches. Virtual Memory and Cache Interaction. Memory interaction with cache memory. TLB, Page Walking. Interfacing with bus attached devices. The memory slave role. Busmastering from devices. Generic DMA contollers. IRQs. Performance Impact of DMA and IRQ. Bandwidth impact of DMA. Latency Impact of IRQs. Verification of bus-attached devices. Isolating device from stimuli generation (ISS, Spike, etc.).

Full-or-part-time: 2h
Theory classes: 2h

Parallel Architectures and Parallel Programming

Description:
Flynn’s Taxonomy: SISD, SIMD, MIMD, SPMD. SIMT. Data parallelism, task parallelism, pipeline parallelism. Parallel Architectures in Practice (SMT, NUMA, GPU). Parallel Programming. Memory: shared memory, distributed memory, hybrid. Programming paradigms: threads, message passing. Synchronization challenges. Existing models: OpenMP, MPI, CUDA, etc. The JPEG dataflow and its possible pipelined/data parallel execution.

Full-or-part-time: 2h
Theory classes: 2h

HW/SW Codesign

Description:
HW/SW Codesign and High Level Synthesis. Communication, synchronization, hardware-software partitioning. The role of system-level modeling and simulation tools. Model Based Design. HLS Tools and Workflow. Contrast it with traditional RTL design approaches. HLS tools and languages (e.g., C++, SystemC). HLS in System Verification and HW/SW Co-Design. Functional Verification. Performance Estimation. Designing bus attached coprocessors with HLS. Introducing to the DCT.

Full-or-part-time: 2h
Theory classes: 2h

IP Core Business

Description:
IP Core Parametrization. Language and tools. Real-world examples. Quality Assessment of IP Cores. Simulation models. Certification. Standards. Associated costs. Markets of IP Cores. Current landscape of the IP core market. Levels of IP (from RTL to Chiplets). Trends and forecasts. Business Models and Licensing. Protection of IP Cores. Importance of IP Core Protection. Attacks and Reverse engineering. Methods of protection (obfuscation, encryption, hardware security). Patents.

Full-or-part-time: 2h
Theory classes: 2h

GRADING SYSTEM

Labs (50%)
Final Exam (50%)

BIBLIOGRAPHY

Basic:
- Hennessy, John L; Patterson, David A. Computer architecture : a quantitative approach. Sixth edition. Cambridge, MA: Elsevier/Morgan Kaufmann, [2019]. ISBN 9780128119051.
- Spear, Chris; Tumbush, Greg. SystemVerilog for verification: a guide to learning the testbench language features [on line]. 2nd ed. New York: Springer, 2012 [Consultation: 27/03/2024]. Available on: https://link-springer-com.recursos.biblioteca.upc.edu/book/10.1007/978-1-4614-0715-7. ISBN 9781461407157.
- Maaref. M. Architecting and building high-speed xoCs : design, develop, and debug complex FPGA-based systems-on-chip [on line]. 1st ed. Birmingham: Packt Publishing, Limited, 2022 [Consultation: 19/04/2024]. Available on: https://ebookcentral-proquest-com.recursos.biblioteca.upc.edu/lib/upcatalunya-ebooks/detail.action?pq-origsite=primo&docID=30239983. ISBN 1-80181-985-8.
- Taraate, Vaibbhav. ASIC Design and Synthesis. RTL Design Using Verilog [on line]. Singapore: Springer Nature, 2021 [Consultation: 08/04/2024]. Available on: https://link-springer-com.recursos.biblioteca.upc.edu/book/10.1007/978-981-33-4642-0. ISBN 9789813346420.
- Thomas, D. Logic design and verification using SystemVerilog. Revised. Lexington, Kentucky: CreateSpace, 2016. ISBN 9781523364022.
- Wile, Bruce ; Goss, John c.; Roesner, Wolfgang. Comprehensive Functional Verification. The complete industry cycle [on line]. Amsterdam ; Boston: Morgan Kaufmann Publishers, 2005 [Consultation: 08/04/2024]. Available on: https://ebookcentral-proquest-com.recursos.biblioteca.upc.edu/lib/upcatalunya-ebooks/detail.action?pq-origsite=primo&docID=234976. ISBN 9781281008398.
- Salemi, R. The UVM Primer: an introduction to the Universal Verification Methodology. Boston: Boston Light Press, 2013. ISBN 9780974164939.