



## Guía docente

# 2301216 - AIPCD - Diseño Avanzado de Núcleos de Propiedad Intelectual

Última modificación: 04/04/2024

**Unidad responsable:** Escuela Técnica Superior de Ingeniería de Telecomunicación de Barcelona

**Unidad que imparte:** 1022 - UAB - (CAS) pendent.

**Titulación:** MÁSTER UNIVERSITARIO EN INGENIERÍA DE SEMICONDUCTORES Y DISEÑO MICROELECTRÓNICO (Plan 2024). (Asignatura optativa).

**Curso:** 2024

**Créditos ECTS:** 4.0

**Idiomas:** Inglés

## PROFESORADO

---

**Profesorado responsable:** Castells Rufas, David

**Otros:** Moretó Planas, Miquel  
Genovese, Ignacio

## CAPACIDADES PREVIAS

---

Diseño digital basado en un lenguaje de descripción hardware de nivel RTL (VHDL, Verilog,...).

Diseño y simulación de sistemas digitales básicos: funciones lógicas combinacionales y secuenciales, funciones aritméticas y máquinas de estados finitos.

Implementación y depuración de sistemas digitales básicos sobre dispositivos reconfigurables (FPGAs).

Desarrollo de aplicaciones software basadas en microprocesador/microcontrolador.

C/C++, lenguaje de programación Python.

## METODOLOGÍAS DOCENTES

---

Clases de teoría.

Trabajos prácticos de laboratorio.

Trabajo individual (a distancia).

Examen final.

## OBJETIVOS DE APRENDIZAJE DE LA ASIGNATURA

---

Identificar diferentes repertorios de instrucciones y seleccionar el más adecuado para implementar un sistema.

Ser capaz de estimar el potencial de paralelismo de los algoritmos para guiar la selección o el diseño de la plataforma.

Identificar cuellos de botella de rendimiento y energía.

Ser capaz de crear hardware específico (IP Core) con métodos de aseguramiento de calidad que permitan su fácil integración en los sistemas.

Ser capaz de desarrollar código de software optimizado para el mejor uso de los núcleos IP para minimizar el tiempo de ejecución y el consumo de energía.

Aprenda a diseñar un sistema informático que combine IP complejas disponibles (como procesadores, memorias, coprocesadores, etc.) para realizar una función específica.

Adquiera conocimientos y práctica práctica sobre la creación de prototipos FPGA y métodos de verificación basados en software.



## HORAS TOTALES DE DEDICACIÓN DEL ESTUDIANTADO

| Tipo                       | Horas | Porcentaje |
|----------------------------|-------|------------|
| Horas grupo pequeño        | 12,0  | 12.00      |
| Horas grupo grande         | 18,0  | 18.00      |
| Horas aprendizaje autónomo | 70,0  | 70.00      |

**Dedicación total:** 100 h

## CONTENIDOS

### Tipos de núcleos IP

**Descripción:**

Núcleos IP del procesador (microprocesadores, microcontroladores, DSP). Memorias, RAM (SRAM, SDRAM,...), ROM, Caché, NVM. Controladores de memoria. Interfaces de comunicación. Interfaces punto a punto (serie/paralelo, unidireccional/bidireccional) Interfaces multipunto (multimaestro, multiesclavo). Interfaces Mapeadas en Memoria (Buses, AXI), Redes en Chip. Integración de núcleos de interfaz IP en sistemas electrónicos (Generadores, Platform Builder,...). Núcleos especializados. (criptográfico, procesamiento de imágenes, ...). Personalización y adaptación de núcleos especializados.

**Dedicación:** 2h

Grupo grande/Teoría: 2h

### Verificación del núcleo IP

**Descripción:**

Importancia de la Verificación. Garantizar la corrección, confiabilidad y funcionalidad de los núcleos IP. El proceso de verificación (flujo desde el diseño hasta las pruebas). Metodologías de prueba sistemáticas. Metodologías de verificación. Verificación basada en simulación. Simulación basada en eventos versus simulación basada en ciclos (QuestaSIM vs Verilator). Verificación formal. UVM (bancos de pruebas, agentes, secuencias,...). Cocotb, CI/CD (Flujos). Emulación FPGA. Estudios de casos del mundo real: lecciones aprendidas y desafíos enfrentados en proyectos del mundo real.

**Dedicación:** 2h

Grupo grande/Teoría: 2h

### Métricas

**Descripción:**

Importancia de las métricas. Tipos de Métricas (área, rendimiento, consumo de energía). Definición de área: área de chip, recuento de puertas, utilización de recursos. Métodos para medir el área. Herramientas (Síntesis...) Métricas de Rendimiento. Retraso (tiempo de configuración, tiempo de espera, retraso de propagación) y Fmax. Tiempo de ejecución. Banda ancha. Modelo de línea de techo. Herramientas (Perfilado). Relación con Simuladores (Spike, Gem5). Métricas de consumo de energía. Potencia dinámica y potencia estática. Medición del consumo de energía. Optimización de energía (Clock Gating, Power Gating, DVFS). Herramientas (Simulación, Monitoreo de energía).

**Dedicación:** 2h

Grupo grande/Teoría: 2h



### Estudio de caso: una unidad vectorial como núcleo IP complejo

**Descripción:**

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.

**Dedicación:** 2h

Grupo grande/Teoría: 2h

### Ampliación de procesadores con hardware personalizado: instrucciones personalizadas

**Descripción:**

Máquinas algorítmicas. Rutas de datos. El equilibrio entre recursos informáticos y de almacenamiento. Estimación de los beneficios de rendimiento. Revisión de perfiles. Identificación de instrucciones candidatas para la implementación de hardware. Instrucciones personalizadas y extensiones de arquitectura de procesador. Unidades funcionales para instrucciones personalizadas. Requisitos de integración adicionales (etapa de decodificación). Ampliación de procesadores RISC-V. Desarrollo de HDL. Validación Funcional (Pico Extendido). Estimación de rendimiento (Extending ISS, Gem5). Síntesis. Emulación. Ejemplos (distancia de Hamming, conversión del espacio de color).

**Dedicación:** 2h

Grupo grande/Teoría: 2h

### Jerarquía de memoria y dispositivos.

**Descripción:**

Jerarquía de memoria y conceptos básicos de caché. Niveles de jerarquía de la memoria: registros, caché, memoria principal, almacenamiento secundario. Tipos de cachés (L1, L2, L3). Arquitecturas de caché. Componentes (etiqueta, índice, desplazamiento de bloque). Operación. Mecanismos de lectura y escritura, caché al azar. Cartografía; mapeado directo, asociativo por conjuntos, totalmente asociativo. Políticas de sustitución de caché (LRU, FIFO, Aleatoria,...). Coherencia de caché. Problemas de incoherencia. Coherencia de caché en multinúcleos. Descripción general de protocolos de coherencia como MESI, MOESI y enfoques basados en directorios. Interacción entre memoria virtual y caché. Interacción de la memoria con la memoria caché. TLB, página caminando. Interfaz con dispositivos conectados al bus. El papel de esclavo de la memoria. Busmastering desde dispositivos. Controladores DMA genéricos. IRQ. Impacto en el rendimiento de DMA e IRQ. Impacto del ancho de banda de DMA. Impacto de la latencia de las IRQ. Verificación de dispositivos conectados al autobús. Dispositivo aislante de la generación de estímulos (ISS, Spike, etc.).

**Dedicación:** 2h

Grupo grande/Teoría: 2h

### Arquitecturas paralelas y programación paralela

**Descripción:**

Taxonomía de Flynn: SISD, SIMD, MIMD, SPMD. SIMT. Paralelismo de datos, paralelismo de tareas, paralelismo de canalización. Arquitecturas paralelas en la práctica (SMT, NUMA, GPU). Programación paralela. Memoria: memoria compartida, memoria distribuida, híbrida. Paradigmas de programación: hilos, paso de mensajes. Desafíos de sincronización. Modelos existentes: OpenMP, MPI, CUDA, etc. El flujo de datos JPEG y su posible ejecución paralela de datos/canalización.

**Dedicación:** 2h

Grupo grande/Teoría: 2h



### Codiseño HW/SW

**Descripción:**

Codiseño HW/SW y Síntesis de Alto Nivel. Comunicación, sincronización, particionamiento hardware-software. El papel de las herramientas de simulación y modelado a nivel de sistema. Diseño basado en modelos. Herramientas y flujo de trabajo HLS. Compárelo con los enfoques tradicionales de diseño RTL. Herramientas y lenguajes HLS (por ejemplo, C++, SystemC). HLS en Verificación de Sistemas y Co-Diseño HW/SW. Verificación Funcional. Estimación del desempeño. Diseño de coprocesadores conectados a bus con HLS. Introducción al DCT.

**Dedicación:** 2h

Grupo grande/Teoría: 2h

### Negocio en torno a los núcleos IP

**Descripción:**

Parametrización del núcleo IP. Motivación. Lenguaje y herramientas. Ejemplos del mundo real. Evaluación de Calidad de Núcleos IP. Modelos de simulación. Certificación. Estándares. Costos asociados. Mercados de Cores IP. Panorama actual del mercado principal de IP. Niveles de IP (desde RTL hasta Chiplets). Tendencias y previsiones. Modelos de Negocio y Licencias. Protección de Núcleos IP. Importancia de la protección del núcleo IP. Ataques e ingeniería inversa. Métodos de protección (ofuscación, cifrado, seguridad de hardware). Patentes.

**Dedicación:** 2h

Grupo grande/Teoría: 2h

## SISTEMA DE CALIFICACIÓN

Laboratorios (50%)

Examen final (50%)

## BIBLIOGRAFÍA

**Básica:**

- 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 [en línea]. 2nd ed. New York: Springer, 2012 [Consulta: 27/03/2024]. Disponible a: <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 x0Cs : design, develop, and debug complex FPGA-based systems-on-chip [en línea]. 1st ed. Birmingham: Packt Publishing, Limited, 2022 [Consulta: 19/04/2024]. Disponible a: <https://ebookcentral-proquest-com.recursos.biblioteca.upc.edu/lib/upcatalunya-ebooks/detail.action?pg-origsite=primo&docID=30239983>. ISBN 1-80181-985-8.
- Taraate, Vaibbhav. ASIC Design and Synthesis. RTL Design Using Verilog [en línea]. Singapore: Springer Nature, 2021 [Consulta: 08/04/2024]. Disponible a: <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 [en línea]. Amsterdam ; Boston: Morgan Kaufmann Publishers, 2005 [Consulta: 08/04/2024]. Disponible a: <https://ebookcentral-proquest-com.recursos.biblioteca.upc.edu/lib/upcatalunya-ebooks/detail.action?pg-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.