



UNIVERSIDAD  
POLITÉCNICA  
DE MADRID

PROCESO DE  
COORDINACIÓN DE LAS  
ENSEÑANZAS PR/CL/001



E.T.S. de Ingenieros  
Industriales

# ANX-PR/CL/001-01

## GUÍA DE APRENDIZAJE

### ASIGNATURA

**53001548 - Embedded Processing Architectures**

### PLAN DE ESTUDIOS

05BG - Master Universitario En Electronica Industrial

### CURSO ACADÉMICO Y SEMESTRE

2022/23 - Segundo semestre

## Índice

---

### Guía de Aprendizaje

1. Datos descriptivos.....	1
2. Profesorado.....	1
3. Conocimientos previos recomendados.....	2
4. Competencias y resultados de aprendizaje.....	2
5. Descripción de la asignatura y temario.....	3
6. Cronograma.....	5
7. Actividades y criterios de evaluación.....	7
8. Recursos didácticos.....	9

## 1. Datos descriptivos

---

### 1.1. Datos de la asignatura

<b>Nombre de la asignatura</b>	53001548 - Embedded Processing Architectures
<b>No de créditos</b>	3 ECTS
<b>Carácter</b>	Optativa
<b>Curso</b>	Primer curso
<b>Semestre</b>	Segundo semestre
<b>Período de impartición</b>	Febrero-Junio
<b>Idioma de impartición</b>	Inglés/Castellano
<b>Titulación</b>	05BG - Master Universitario en Electronica Industrial
<b>Centro responsable de la titulación</b>	05 - Escuela Técnica Superior De Ingenieros Industriales
<b>Curso académico</b>	2022-23

## 2. Profesorado

---

### 2.1. Profesorado implicado en la docencia

<b>Nombre</b>	<b>Despacho</b>	<b>Correo electrónico</b>	<b>Horario de tutorías</b> *
Alfonso Rodriguez Medina (Coordinador/a)	UD Electrónica	alfonso.rodriguez@upm.es	Sin horario. Please send an email to arrange a meeting

\* Las horas de tutoría son orientativas y pueden sufrir modificaciones. Se deberá confirmar los horarios de tutorías con el profesorado.

## 3. Conocimientos previos recomendados

---

### 3.1. Asignaturas previas que se recomienda haber cursado

El plan de estudios Master Universitario en Electronica Industrial no tiene definidas asignaturas previas recomendadas para esta asignatura.

### 3.2. Otros conocimientos previos recomendados para cursar la asignatura

- Generic knowledge of microprocessor/microcontroller programming
- Basic computer architecture background

## 4. Competencias y resultados de aprendizaje

---

### 4.1. Competencias

CB06 - Poseer y comprender conocimientos que aporten una base u oportunidad de ser originales en el desarrollo y/o aplicación de ideas, a menudo en un contexto de investigación

CB07 - Que los estudiantes sepan aplicar los conocimientos adquiridos y su capacidad de resolución de problemas en entornos nuevos o poco conocidos dentro de contextos más amplios (o multidisciplinares) relacionados con su área de estudio

CB08 - Que los estudiantes sean capaces de integrar conocimientos y enfrentarse a la complejidad de formular juicios a partir de una información que, siendo incompleta o limitada, incluya reflexiones sobre las responsabilidades sociales y éticas vinculadas a la aplicación de sus conocimientos y juicios

CE01 - Comprender, diseñar y analizar sistemas y componentes electrónicos en el ámbito de la electrónica industrial. Modelización y caracterización de sistemas electrónicos complejos.

CG01 - Haber adquirido conocimientos avanzados y demostrado, en un contexto de investigación científica y tecnológica o altamente especializado, una comprensión detallada y fundamentada de los aspectos teóricos y prácticos y de la metodología de trabajo en uno o más campos de estudio

CG02 - Saber aplicar e integrar sus conocimientos, la comprensión de estos, su fundamentación científica y sus capacidades de resolución de problemas en entornos nuevos y definidos de forma imprecisa, incluyendo contextos de carácter multidisciplinar tanto investigadores como profesionales altamente especializados.

CG06 - Haber desarrollado la autonomía suficiente para participar en proyectos de investigación y colaboraciones científicas o tecnológicas dentro de su ámbito temático, en contextos interdisciplinares y, en su caso, con una alta componente de transferencia del conocimiento.

CT01 - Uso de la lengua inglesa

## 4.2. Resultados del aprendizaje

RA25 - Saber clasificar las arquitecturas y modelos de computación para el caso de los sistemas multicore y los manycore.

RA26 - Aplicar criterios de valoración y de selección de las opciones óptimas para resolver problemas de cómputo complejos, empleando arquitecturas de procesamiento avanzadas.

RA24 - Aplicar e integrar los conocimientos sobre diseño de sistemas microprocesadores avanzados y de procesamiento paralelo

## 5. Descripción de la asignatura y temario

---

### 5.1. Descripción de la asignatura

This subject is focused as an architectural approach towards more performing processing architectures. Nowadays, there is a paradigmatic change from single processing architectures to multiple processor / multiple core approaches. There is no way to increase performance with just a single core, and so, other alternatives are required.

We are assuming students have basic knowledge of microprocessor programming and basic architectures. Taking this as the starting point, we first review ways of improving performance by means of enhanced architectures, improved from the point of view of internal bus arrangements, load/store operation, RISC instruction set coding, and, later, pipelining and memory hierarchy are reviewed in detailed. All these approaches are done quantitatively and trading cost vs performance (following the world widespread theories from Patterson and Hennesy),

Once it is shown that there are no further ways to improve performance with single core architectures, we move to multi-core and many-core approaches. Here, the problem is not only architectural, but also it has an impact on how

to program and synchronize this type of systems. Therefore, this part contains an interleaved scheme where architectures are reviewed and then, ways of programming them are explained and practiced.

In this regard, multi-core (shared memory or distributed memory) programming is covered with pthreads and OpenMP, many-core (e.g., GPU) system programming is covered with OpenCL. The course also addresses the appearance of domain-specific architectures and the methods to program them.

## 5.2. Temario de la asignatura

1. Computer Architecture Basics
2. Memory Hierarchy in Computing Systems
3. Parallelism in Computing Systems
  - 3.1. Instruction-Level Parallelism
  - 3.2. Thread-Level Parallelism & Multi-Core Architectures
  - 3.3. Data-Level Parallelism & Many-Core Architectures
4. Domain-Specific Computing

## 6. Cronograma

### 6.1. Cronograma de la asignatura \*

Sem	Actividad en aula	Actividad en laboratorio	Tele-enseñanza	Actividades de evaluación
1	<b>Course Introduction</b> Duración: 02:00 LM: Actividad del tipo Lección Magistral			
2	<b>Computer Architecture Basics (I)</b> Duración: 02:00 LM: Actividad del tipo Lección Magistral			
3	<b>Computer Architecture Basics (II)</b> Duración: 02:00 LM: Actividad del tipo Lección Magistral			
4	<b>Memory Hierarchy</b> Duración: 02:00 LM: Actividad del tipo Lección Magistral			
5		<b>LAB on Memory Hierarchy (gem5)</b> Duración: 02:00 PL: Actividad del tipo Prácticas de Laboratorio		
6	<b>Instruction-Level Parallelism</b> Duración: 02:00 LM: Actividad del tipo Lección Magistral			
7	<b>Thread-Level Parallelism and Multi-Core Architectures</b> Duración: 02:00 LM: Actividad del tipo Lección Magistral			
8		<b>LAB on Multi-Core Programming (pthreads)</b> Duración: 02:00 PL: Actividad del tipo Prácticas de Laboratorio		
9		<b>LAB on Multi-Core Programming (OpenMP)</b> Duración: 02:00 PL: Actividad del tipo Prácticas de Laboratorio		
10		<b>LAB on Multi-Core Programming (MPI)</b> Duración: 02:00 PL: Actividad del tipo Prácticas de Laboratorio		
11	<b>Data-Level Parallelism and Many-Core Architectures</b> Duración: 02:00 LM: Actividad del tipo Lección Magistral			

12		<b>LAB on Many-Core Architectures (OpenCL)</b> Duración: 02:00 PL: Actividad del tipo Prácticas de Laboratorio		
13	<b>Domain-Specific Architectures</b> Duración: 02:00 LM: Actividad del tipo Lección Magistral			
14		<b>LAB on Domain-Specific Architectures</b> Duración: 02:00 PL: Actividad del tipo Prácticas de Laboratorio		
15				
16				
17				<b>Final written exam</b> EX: Técnica del tipo Examen Escrito Evaluación continua y sólo prueba final Presencial Duración: 02:00  <b>Project #1: gem5 system simulation</b> TI: Técnica del tipo Trabajo Individual Evaluación continua y sólo prueba final No presencial Duración: 10:00  <b>Project #2: pthread programming</b> TI: Técnica del tipo Trabajo Individual Evaluación continua y sólo prueba final No presencial Duración: 10:00  <b>Project #3: OpenMP programming</b> TI: Técnica del tipo Trabajo Individual Evaluación continua y sólo prueba final No presencial Duración: 10:00  <b>Project #4: MPI programming</b> TI: Técnica del tipo Trabajo Individual Evaluación continua y sólo prueba final No presencial Duración: 10:00  <b>Project #5: OpenCL programming</b> TI: Técnica del tipo Trabajo Individual Evaluación continua y sólo prueba final No presencial Duración: 10:00

Para el cálculo de los valores totales, se estima que por cada crédito ECTS el alumno dedicará dependiendo del plan de estudios, entre 26 y 27 horas de trabajo presencial y no presencial.

\* El cronograma sigue una planificación teórica de la asignatura y puede sufrir modificaciones durante el curso derivadas de la situación creada por la COVID-19.



## 7. Actividades y criterios de evaluación

### 7.1. Actividades de evaluación de la asignatura

#### 7.1.1. Evaluación (progresiva)

Sem.	Descripción	Modalidad	Tipo	Duración	Peso en la nota	Nota mínima	Competencias evaluadas
17	Final written exam	EX: Técnica del tipo Examen Escrito	Presencial	02:00	50%	5 / 10	CT01 CE01 CG01
17	Project #1: gem5 system simulation	TI: Técnica del tipo Trabajo Individual	No Presencial	10:00	10%	/ 10	CG06 CB06 CB07 CT01 CE01 CG01 CG02 CB08
17	Project #2: pthread programming	TI: Técnica del tipo Trabajo Individual	No Presencial	10:00	10%	/ 10	CB06 CB07 CT01 CE01 CG01 CG02 CB08 CG06
17	Project #3: OpenMP programming	TI: Técnica del tipo Trabajo Individual	No Presencial	10:00	10%	/ 10	CB07 CT01 CE01 CG01 CG02 CB08 CG06 CB06
17	Project #4: MPI programming	TI: Técnica del tipo Trabajo Individual	No Presencial	10:00	10%	/ 10	CG06 CB06 CB07 CT01 CE01 CG01 CG02 CB08

17	Project #5: OpenCL programming	TI: Técnica del tipo Trabajo Individual	No Presencial	10:00	10%	/ 10	CB07 CT01 CG06 CB06 CE01 CG01 CG02 CB08
----	--------------------------------	---	---------------	-------	-----	------	--

### 7.1.2. Prueba evaluación global

Sem	Descripción	Modalidad	Tipo	Duración	Peso en la nota	Nota mínima	Competencias evaluadas
17	Final written exam	EX: Técnica del tipo Examen Escrito	Presencial	02:00	50%	5 / 10	CT01 CE01 CG01
17	Project #1: gem5 system simulation	TI: Técnica del tipo Trabajo Individual	No Presencial	10:00	10%	/ 10	CG06 CB06 CB07 CT01 CE01 CG01 CG02 CB08
17	Project #2: pthread programming	TI: Técnica del tipo Trabajo Individual	No Presencial	10:00	10%	/ 10	CB06 CB07 CT01 CE01 CG01 CG02 CB08 CG06
17	Project #3: OpenMP programming	TI: Técnica del tipo Trabajo Individual	No Presencial	10:00	10%	/ 10	CB07 CT01 CE01 CG01 CG02 CB08 CG06 CB06
17	Project #4: MPI programming	TI: Técnica del tipo Trabajo Individual	No Presencial	10:00	10%	/ 10	CG06 CB06 CB07 CT01 CE01 CG01 CG02 CB08

17	Project #5: OpenCL programming	TI: Técnica del tipo Trabajo Individual	No Presencial	10:00	10%	/ 10	CB07 CT01 CG06 CB06 CE01 CG01 CG02 CB08
----	--------------------------------	--	---------------	-------	-----	------	--

### 7.1.3. Evaluación convocatoria extraordinaria

No se ha definido la evaluación extraordinaria.

## 7.2. Criterios de evaluación

The score is based on:

Final exam - 50%

Practical projects - 50%

## 8. Recursos didácticos

### 8.1. Recursos didácticos de la asignatura

Nombre	Tipo	Observaciones
Computer Architecture: A Quantitative Approach (6th edition)	Bibliografía	John L. Hennessy & David A. Patterson, Morgan Kaufmann, 2017
Computer Organization and Design: The Hardware Software Interface (2nd RISC-V edition)	Bibliografía	David A. Patterson & John L. Hennessy, Morgan Kaufmann, 2020