

EMBEDDED SYSTEMS

LEARNING GUIDE

MSC IN SYSTEMS AND SERVICES ENGINEERING
FOR THE INFORMATION SOCIETY



UNIVERSIDAD POLITÉCNICA DE MADRID

2010 - 2011

This page intentionally left blank

CONTENTS

1.	BASIC INFORMATION	5
2.	LECTURERS	5
3.	REQUIRED BACKGROUND	5
3.1.	Approved Courses	5
3.2.	Other Learning Results	6
4.	SYLLABUS	6
4.1.	Lesson 1: Programmable Logic Devices (0.25 ECTS)	6
4.1.1.	LEARNING OUTCOMES	6
4.1.2.	ACHIEVEMENT INDICATORS	6
4.1.3.	CONTACT HOURS (3 HOURS)	6
4.1.3.1.	<i>Lectures (3 hours)</i>	6
4.1.4.	ADDITIONAL STUDENT WORK (3 HOURS)	7
4.2.	Lesson 2: VHDL Language (0.65 ECTS)	7
4.2.1.	LEARNING OUTCOMES	7
4.2.2.	ACHIEVEMENT INDICATORS	7
4.2.3.	CONTACT HOURS (5 HOURS)	7
4.2.3.1.	<i>Lectures (4 hours)</i>	7
4.2.3.2.	<i>Laboratory work (1 hour)</i>	8
4.2.4.	ADDITIONAL STUDENT WORK (12 HOURS)	8
4.3.	Lesson 3: Functional Verification and Test-Bench Design (0.65 ECTS)	8
4.3.1.	LEARNING OUTCOMES	8
4.3.2.	ACHIEVEMENT INDICATORS	8
4.3.3.	CONTACT HOURS (5 HOURS)	9
4.3.3.1.	<i>Lectures (2 hours)</i>	9
4.3.3.2.	<i>Laboratory work (3 hours)</i>	9
4.3.4.	ADDITIONAL STUDENT WORK (12 HOURS)	9
4.4.	Lesson 4: Parallelism & Pipelining (0.2 ECTS)	9
4.4.1.	LEARNING OUTCOMES	9
4.4.2.	ACHIEVEMENT INDICATORS	9
4.4.3.	CONTACT HOURS (2 HOURS)	10
4.4.3.1.	<i>Lectures (2 hours)</i>	10
4.4.4.	ADDITIONAL STUDENT WORK (3 HOURS)	10
4.5.	Lesson 5: Intermediate Project: Simple Peripheral Verification (0.7 ECTS)	10
4.5.1.	LEARNING OUTCOMES	10

4.5.2.	ACHIEVEMENT INDICATORS	10
4.5.3.	CONTACT HOURS (6 HOURS)	11
4.5.3.1.	<i>Laboratory work (6 hours)</i>	11
4.5.4.	ADDITIONAL STUDENT WORK (12 HOURS)	11
4.6.	Lesson 6: Configurable Embedded Systems (0.75 ECTS)	11
4.6.1.	LEARNING OUTCOMES	11
4.6.2.	ACHIEVEMENT INDICATORS	11
4.6.3.	CONTACT HOURS (9 HOURS)	11
4.6.3.1.	<i>Lectures (1 hour)</i>	11
4.6.3.2.	<i>Case study: Altera Nios II (5 hours)</i>	12
4.6.3.3.	<i>Laboratory work (3 hours)</i>	12
4.6.4.	ADDITIONAL STUDENT WORK (11 HOURS)	12
4.7.	Lesson 7: Software Design Flow for Embedded Systems (1.0 ECTS)	12
4.7.1.	LEARNING OUTCOMES	12
4.7.2.	ACHIEVEMENT INDICATORS	12
4.7.3.	CONTACT HOURS (12 HOURS)	13
4.7.3.1.	<i>Lectures (2 hours)</i>	13
4.7.3.2.	<i>Case study (4 hours)</i>	13
4.7.3.3.	<i>Laboratory work (6 hours)</i>	14
4.7.4.	ADDITIONAL STUDENT WORK (15 HOURS)	14
4.8.	Lesson 8: Course Project (0.8 ECTS)	14
4.8.1.	ACHIEVEMENT INDICATORS	14
4.8.2.	STUDENT WORK (23 HOURS)	14
5.	PLANNING	15
6.	WORKLOAD	15
7.	REFERENCES	16
7.1.	Main References	16
7.2.	Auxiliary References	16
8.	CAD TOOLS	17
9.	REQUIRED LABORATORY EQUIPMENT	18
10.	ASSESSMENT PROCEDURES	18
11.	MASTER AND COURSE LEARNING OUTCOMES	20

1. BASIC INFORMATION

- Course name: Embedded Systems
- Subject: Electronics
- Title: MSc in Systems and Services Engineering for the Information Society
- Year: 2010/2011
- First year, second semester
- Five ECTS
- Fourteen weekly contact sessions, 3 hours each
- Maximum of 16 students
- Compulsory in the Master professional itinerary and optional in the research one
- Language: English

2. LECTURERS

Name	Room	Dept.	email
M. César RODRÍGUEZ LACRUZ*	4114	SEC	mcesar@sec.upm.es
César SANZ ÁLVARO	6104	SEC	cesar@sec.upm.es
Eduardo JUÁREZ MARTÍNEZ	4101	EF	ejuarez@euitt.upm.es

* Coordinator

3. REQUIRED BACKGROUND

3.1. Approved Courses

- Advanced Digital Architectures (MISSSI-10)

3.2. Other Learning Results

- Analysis, application and design of wired digital circuits
- Application of the Von Neumann's computer architecture
- Application of processor peripherals
- Application of processor interrupts
- Programming and debugging using the C language (with emphasis on structures, pointers and memory management)

4. SYLLABUS

4.1. Lesson 1: Programmable Logic Devices (0.25 ECTS)

4.1.1. LEARNING OUTCOMES

- L0.01. To know Programmable Logic Devices as a target technology for implementing digital electronic designs
- L0.02. To know the evolution of these devices by means of analyzing the product roadmap from some relevant manufacturers

4.1.2. ACHIEVEMENT INDICATORS

- AI.01. For a given set of applications select the best suited PLD type and family for each one

4.1.3. CONTACT HOURS (3 HOURS)

4.1.3.1. Lectures (3 hours)

- Introduction
 - PLD concept
 - Implementation technologies
 - Device configuration
 - Classification and applications
- Programmable Logic Devices review
 - Simple PLD
 - Complex PLD
 - FPGA

- FPGA with embedded processors

4.1.4. ADDITIONAL STUDENT WORK (3 HOURS)

- Readings

4.2. Lesson 2: VHDL Language (0.65 ECTS)

4.2.1. LEARNING OUTCOMES

- LO.03. To comprehend VHDL language and to apply to combinational circuits modeling
- LO.04. To comprehend VHDL language and to apply to sequential circuits modeling
- LO.05. To comprehend VHDL language and to apply to hierarchical models
- LO.06. To know the concept of test-bench in VHDL
- LO.07. To learn basic concepts of the VHDL language
- LO.08. To learn the flow for VHDL design using CAD tools

4.2.2. ACHIEVEMENT INDICATORS

- AI.02. Implementation of digital circuits using VHDL language
- AI.03. To be able to use a VHDL simulation tool

4.2.3. CONTACT HOURS (5 HOURS)

4.2.3.1. Lectures (4 hours)

- Designing with Hardware Description Languages (HDL)
 - Introduction
 - Modeling fundamentals
 - Combinational circuits description for logic synthesis
 - Sequential circuits description for logic synthesis
 - Complex circuits description
 - The test-bench
- VHDL language
 - Design units and libraries
 - Types and operators
 - VHDL objects

- Processes
- Hierarchy

4.2.3.2. *Laboratory work (1 hour)*

- Design flow for VHDL design using CAD tools
 - Modeling
 - Simulation
 - Synthesis
 - Physical design

4.2.4. ADDITIONAL STUDENT WORK (12 HOURS)

- Hand-on tutorial to learn using ModelSim
- To implement and verify some example circuits

4.3. Lesson 3: Functional Verification and Test-Bench Design (0.65 ECTS)

4.3.1. LEARNING OUTCOMES

- L0.09. To know the difference between the RTL and functional approaches to VHDL system specification
- L0.10. To comprehend, apply and implement subprograms in VHDL with functions and procedures
- L0.11. To comprehend, apply and implement VHDL Test-bench specifications by means of assert statements
- L0.12. To comprehend, apply and implement Test-bench stimulus with VHDL files
- L0.13. To implement VHDL Test-bench stimulus specifications by means of subprograms, vectors or signals
- L0.14. To comprehend and apply cover analysis tools to verify system specifications

4.3.2. ACHIEVEMENT INDICATORS

- AI.04. Implementation of simple VHDL test-bench specifications for functional verification

4.3.3. CONTACT HOURS (5 HOURS)

4.3.3.1. Lectures (2 hours)

- VHDL language syntax for functional verification
 - Functions and procedures
 - Assert statements
 - File management
- Test-bench design
 - Signal generation
 - Procedures
 - Complex stimuli

4.3.3.2. Laboratory work (3 hours)

- Project: Design the test-bench of a simple given entity using signal generation, procedures and complex stimuli

4.3.4. ADDITIONAL STUDENT WORK (12 HOURS)

- Homework
- Readings

4.4. Lesson 4: Parallelism & Pipelining (0.2 ECTS)

4.4.1. LEARNING OUTCOMES

- LO.15. To understand the concept of pipelining
- LO.16. To know the relationship between latency and throughput
- LO.17. To understand the concept of parallelism
- LO.18. To know the tradeoff between area and speed in digital circuits

4.4.2. ACHIEVEMENT INDICATORS

- AI.05. Partial implementation of a complex algorithm balancing the number of operators *vs* the circuit performance

4.4.3. CONTACT HOURS (2 HOURS)

4.4.3.1. Lectures (2 hours)

- Segmentation
 - Concept of pipeline
 - Initial latency
 - Latency and throughput
 - Example
 - Static and dynamic pipelines
- Parallelism
 - Concept
 - Parallelism vs pipeline
 - Example

4.4.4. ADDITIONAL STUDENT WORK (3 HOURS)

- Exercices

4.5. Lesson 5: Intermediate Project: Simple Peripheral Verification (0.7 ECTS)

4.5.1. LEARNING OUTCOMES

- L0.19. To partially implement the support package to verify the VHDL specification of an image interpolation and decimation filter
- L0.20. To partially implement the VHDL specification of an image interpolation and decimation filter
- L0.21. To partially implement a VHDL test-bench specification of an image interpolation and decimation filter
- L0.22. To partially implement a VHDL test-bench specification of an image scaler built with the interpolation and decimation filter

4.5.2. ACHIEVEMENT INDICATORS

- AI.06. Implementation of a complex VHDL test-bench specification for the functional verifications of an image scaler

4.5.3. CONTACT HOURS (6 HOURS)

4.5.3.1. *Laboratory work (6 hours)*

- Project: Implement the functional verification of an image processing entity specify in VHDL to scale an image

4.5.4. ADDITIONAL STUDENT WORK (12 HOURS)

- Homework
- Readings

4.6. Lesson 6: Configurable Embedded Systems (0.75 ECTS)

4.6.1. LEARNING OUTCOMES

- I0.23. Comprehend the architecture of current configurable embedded processors
- I0.24. Comprehend the main features of the most usual peripherals for configurable embedded processors
- I0.25. Apply the processor/peripheral interconnection scheme for a current configurable embedded processor
- I0.26. Apply the hardware design flow for a current configurable embedded processor in order to synthesize the architecture of and embedded system

4.6.2. ACHIEVEMENT INDICATORS

- AI.07. Synthesize upon a given FPGA based board a given architecture comprising a configurable embedded processor and some peripherals, including a custom made one

4.6.3. CONTACT HOURS (9 HOURS)

4.6.3.1. *Lectures (1 hour)*

- Concept of configurable embedded system
- Current configurable embedded systems

4.6.3.2. *Case study: Altera Nios II (5 hours)*

- Processor architecture
- Avalon bus
- Available peripherals
- SoPC Builder
- Design flow

4.6.3.3. *Laboratory work (3 hours)*

- Project: Synthesize a given configurable embedded system architecture upon a FPGA based board. The system will contain a processor and various peripherals, including a custom made one

4.6.4. ADDITIONAL STUDENT WORK (11 HOURS)

- Readings

4.7. Lesson 7: Software Design Flow for Embedded Systems (1.0 ECTS)

4.7.1. LEARNING OUTCOMES

- L0.27. Comprehend the main features of RTOS and generic OS for a current configurable embedded processor
- L0.28. Comprehend the software structure of a OS driver aimed to manage a custom made peripheral
- L0.29. Apply the OS support for a user program to access a device
- L0.30. Apply the hardware and software tools used to deploy and debug and OS and its applications upon a configurable embedded processor

4.7.2. ACHIEVEMENT INDICATORS

- AI.08. Reproduce a given tutorial describing the steps needed to use a custom made peripheral of a configurable embedded system supporting a generic OS

4.7.3. CONTACT HOURS (12 HOURS)

4.7.3.1. Lectures (2 hours)

- Fundamentals of Linux driver development
 - Contexts (process/kernel/atomic)
 - Character & block devices
 - Other devices (net, video, filesystems)
 - Device major & minor numbers (*mknod*)
 - Modules
- The structure of a Linux driver module
 - Driver functions
 - Data structures
 - Module registering

4.7.3.2. Case study (4 hours)

Linux kernel support

- For inter-context data exchange: *copy_to_user()*, *copy_from_user()*
- For memory allocation
 - *kmalloc()*, *kfree()*
 - Lookaside caches
 - Memory pools
 - *__get_free_page()*
 - *vmalloc()*, *vfree()*
 - Per CPU variables
 - Boot time allocation
- For synchronization
 - Spinlocks
 - Semaphores
 - Mutexes
- For timing
 - Jiffies
 - Delays
 - Tasklets
 - Workqueues
- To communicate with hardware
 - I/O and memory addressing
 - Barriers
- For interrupt management
 - Registering IRQ handlers
 - Top and bottom halves

- IRQ sharing

4.7.3.3. Laboratory work (6 hours)

- Tutorial: Deploying a uClinux OS upon a configurable embedded system
 - Hardware design
 - Toolchain compilation
 - Kernel configuration (vendor_hwselect, Kconfig scripts, Makefile scripts, make menuconfig)
 - Kernel compilation
 - Booting (JTAG)
 - Cross building user applications
 - Remote debugging user applications
 - Remote kernel debugging
- Tutorial: Building a uClinux character device to access a memory chip

4.7.4. ADDITIONAL STUDENT WORK (15 HOURS)

- Tutorial: Basic *NIX commands and tools
 - Basic commands: *cd*, command options, *ls*, *pwd*, *cp*, *mv*, *rm*, *mkdir*, *rmdir*, *cat*, *more*, *man*
 - B shell: *history*, *!*, *=*, *unset*, *>*, *<*, *|*
 - Process control: *free*, *ps*, *&*, *^Z*, *bg*, *fg*, *kill*
 - Other utilities: *find*, *grep*, *tar*, *dd*
- Readings

4.8. Lesson 8: Course Project (0.8 ECTS)

4.8.1. ACHIEVEMENT INDICATORS

- AI.09. Synthesize an embedded system comprising a configurable embedded processor with a custom made peripheral and its accompanying software to fit a given application

4.8.2. STUDENT WORK (23 HOURS)

- Design: Access some inboard resources from a user application in a board comprising an FPGA with a Nios II processor running uClinux

5. PLANNING

Session	1 st hour	2 nd hour	3 rd hour
1	L1 lecture	L1 lecture	L1 lecture
2	L2 lecture	L2 lecture	L2 lab
3	L2 lecture	L2 lecture	L3 lecture
4	L3 lecture	L3 lab	L3 lab
5	L3 lab	L4 lecture	L4 lecture
6	L5 lab	L5 lab	L5 lab
7	L5 lab	L5 lab	L5 lab
8	L6 lecture	L6 lecture	L6 lecture
9	L6 lecture	L6 lecture	L6 lecture
10	L6 lab	L6 lab	L6 lab
11	L7 lecture	L7 lecture	L7 lecture
12	L7 lecture	L7 lecture	L7 lecture
13	L7 lab	L7 lab	L7 lab
14	L7 lab	L7 lab	L7 lab

6. WORKLOAD

Lesson	Lecture hours	Laboratory hours	Contact hours	Additional hours	Total hours	ECTS
1	3	0	3	3	6	0.25
2	4	1	5	12	17	0.65
3	2	3	5	12	17	0.65
4	2	0	2	3	5	0.20
Intermediate Project	0	6	6	12*	18	0.70
6	6	3	9	11	20	0.75
7	6	6	12	15	27	1.00
Course Project	0	0	0	23*	23	0.80
Totals:	23	19	42	91	133	5.00

* Including assessment hours

7. REFERENCES

7.1. Main References

- The Design Warrior's Guide to FPGAs
 - Clive Maxfield
 - Elsevier; 2004
 - 542 pages
 - ISBN: 978-0750676045
- VHDL for Logic Synthesis
 - Andrew Rushton
 - Wiley; 2 edition (July 7, 1998)
 - 390 pages
 - ISBN: 978-0471983255
- Altera Nios II Literature
 - Freely downloadable from Altera's web page at <http://www.altera.com>
 - Nios II Processor Reference Handbook
 - Volume 4: SoPC Builder
 - Embedded Design Handbook
- Linux Device Drivers, 3rd Edition
 - Jonathan Corbet, Alessandro Rubini & Greg Kroah-Hartman
 - O'Reilly Media; 3 edition (February 7, 2005)
 - 640 pages
 - ISBN: 978-0596005900
 - Freely downloadable from <http://lwn.net/Kernel/LDD3/>
- Nios Wiki
 - <http://www.nioswiki.com>, although recently moved to <http://www.alterawiki.com>
 - Contains valuable information about uClinux OS deployment upon Nios II based systems

7.2. Auxiliary References

- The Designer's Guide to VHDL
 - Peter J. Ashenden
 - Morgan Kaufmann; 3 edition (May 29, 2008)
 - 936 pages

- ISBN: 978-0120887859
- Essential Linux Device Drivers
 - Sreekrishnan Venkateswaran
 - Prentice Hall; 1 edition (April 6, 2008)
 - 744 pages
 - ISBN: 978-0132396554
- Understanding the Linux Kernel
 - Daniel P. Bovet, Marco Cesati
 - O'Reilly Media; 3 edition (November 2005)
 - 944 pages
 - ISBN: 978-0596005658
- Building Embedded Linux Systems
 - Karim Yaghmour, Jon Masters, Phillippe Gerum, Gilad Ben-Yossef
 - Pragma; 2 edition (2 Sep 2008)
 - 464 pages
 - ISBN: 978-0596529680
- Embedded Linux Primer: A Practical Real-World Approach
 - Christopher Hallinan
 - Prentice Hall; 2 edition (3 Nov 2010)
 - 656 pages
 - ISBN: 978-0137017836

8. CAD TOOLS

- ModelSim SE/DE/PE
 - Multilanguage simulator
 - There is a free student version of the ModelSim PE tool (with some limitations) downloadable from <http://model.com>
- Quartus II
 - Altera's FPGA design tool
 - There is a free version of Quartus II (named the "Web Edition") freely downloadable from <http://www.altera.com>
- Nios II Embedded Design Suite
 - Altera's Nios II tools for embedded system design
 - Downloadable from <http://www.altera.com>
- The uClinux Nios II distribution
 - Downloadable from <http://www.niosftp.com/pub/linux/>

9. REQUIRED LABORATORY EQUIPMENT

For each student couple the following equipment will be required:

- A PC running the CentOS 5 operating system
 - This OS is freely downloadable from <http://www.centos.org>
 - Other Linux distributions can also apply
- A Terasic-Altera DE2-70 board
 - <http://www.terasic.com>

10. ASSESSMENT PROCEDURES

- AP.1. Initial assessment measuring the achievement of prior learning (weight 0%, must be between 0 % and 10 %)
- AP.2. Auto-assessment exercises (weight 0%, must be between 0 % and 10 %)
- AP.3. Exercises to be solved in the classroom (weight 25%, must be between 10 % and 50 %)
- AP.4. Individual and group homework (weight 15%, must be between 0 % and 20 %)
- AP.5. Laboratory works (weight 60%, must be between 20 % and 60 %)

		Assessment Procedures								
		Indicator								
	Outcome	AI.01	AI.02	AI.03	AI.04	AI.05	AI.06	AI.07	AI.08	AI.09
L1	L0.01	4								
	L0.02	4								
L2	L0.03		4							
	L0.04		4							
	L0.05		4							
	L0.06		4							
	L0.07		4							
	L0.08				3					
L3	L0.09				3					
	L0.10				3					
	L0.11				3					
	L0.12				3					
	L0.13				3					
	L0.14				3					
L4	L0.15					4				
	L0.16					4				
	L0.17					4				
	L0.18					4				
L5	L0.19						5			
	L0.20						5			
	L0.21						5			
	L0.22						5			
L6	L0.23							3		5
	L0.24							3		5
	L0.25							3		5
	L0.26							3		5
L7	L0.27								3	5
	L0.28								3	5
	L0.29								3	5
	L0.30								3	5
Weight		5 %	5 %	5 %	10 %	5 %	20 %	5 %	5 %	40 %
Date		TBD	TBD	TBD	TBD	TBD	TBD	TBD	TBD	TBD

11. MASTER AND COURSE LEARNING OUTCOMES

		Master								
	Course	R01	R07	R08	R09	R10	R11	R12	R13	R14
L1	L0.01							X		
	L0.02							X		
L2	L0.03				X					
	L0.04				X					
	L0.05					X				
	L0.06						X			
	L0.07				X	X	X			
	L0.08									X
L3	L0.09				X	X	X			
	L0.10						X			X
	L0.11						X			X
	L0.12						X			X
	L0.13						X			X
	L0.14						X			X
L4	L0.15								X	
	L0.16								X	
	L0.17								X	
	L0.18								X	
L5	L0.19						X			X
	L0.20				X	X				
	L0.21						X			X
	L0.22						X			X
L6	L0.23	X						X		
	L0.24	X						X		
	L0.25	X						X		
	L0.26									X
L7	L0.27		X							
	L0.28			X						
	L0.29			X						
	L0.30			X						X