IEEE Transactions on Industrial Electronics 57.10 (2010): 3308 – 3316

DOI: http://dx.doi.org/10.1109/TIE.2010.2040557

Course on Digital Electronics Oriented to Describing Systems in VHDL

Francisco J. Azcondo, *Senior Member, IEEE,* Angel de Castro, *Member, IEEE*, Christian Brañas, *Member, IEEE*

***Abstract.*— A new orientation for a multidisciplinary Industrial Engineering program course on Digital Electronics is proposed. Students are trained to describe complete systems in VHDL. These systems have been previously modeled with Matlab/Simulink ®, including both continuous and discrete components. Digital controllers based on concurrent hardware are designed for these systems in the same framework. After developing and simulating the behavioral model of the system or plant, the controller is re-described under the synthesis constraints and implemented in an FPGA evaluation board for verification along with the system under control. In this way, the course is not only intended for digital electronics specialists but also for students with diverse engineering backgrounds.**

***Index Terms*— Digital systems, Education, Hardware design languages, Modeling.**

I. Introduction

UNIVERSITY programs combine specialization and multidisciplinary education. The integration of different disciplines is often addressed by the students on their own, after receiving the content of the different courses. The incorporation of mature research results into the course content tends to increase the specialization of the educational syllabus, making necessary new efforts of integration among disciplines. On the other hand, modeling and design tools can be adapted to describe diverse systems that deal with information or energy processing. The understanding of the general information / energy processing system is a good motivation to learn specific technology, in this case digital electronic design.

Nowadays there is a common need to adapt course content to students with different backgrounds who participate in exchange programs. The objective is to make their integration possible without reducing the academic objectives.

The elective course Digital Electronic Systems, within the fifth (last) year of the multidisciplinary Industrial Engineering Degree (Bachelor and Master) at the University of Cantabria (UC) was initially conceived for specialization purposes. The initial objective was to provide students with deeper knowledge of digital electronics after a first course on fundamentals. The group of students that this course was addressed to was homogeneous in the past, belonging to the specialization in Industrial Electronics and Systems Engineering. During recent years the number of non specialist students that select the course has grown and it is also very common to receive students that participate in exchange agreements; the most common are: Erasmus (European University Exchange Program), SICUE (Spanish Student Exchange Program) and bilateral agreements with North and South American Universities. Recently, the course has been integrated into a new program taught in lingua franca (English) named "Diploma in Advanced Applied Technologies for Industry". The Diploma, described in Table I, includes optional courses in Electronics, Systems Engineering and Mechanics oriented to students with engineering background coming from different Universities. Further details can be found in [1].

TABLE I.
PROGRAM OF THE DIPLOMA "ADVANCED APPLIED TECHNOLOGIES FOR INDUSTRY"

Diploma Advanced Applied Technologies for Industry
Program Description
30 ECTS credits (Elective and Optional)
5657 Spanish History and Culture for Engineering (5ECTS)
5652 Electronic Circuits and Devices (5ECTS)
5655 Digital Electronic Systems (5ECTS)
5651 Advanced Machine Design (5ECTS)
5654 Perception Systems (5ECTS)
5653 Modeling and Simulation of Dynamic Systems (5ECTS)
5656 Spanish Language for Engineers (5ECTS)

The paper describes the new contents and teaching methodology intended to enable students from diverse origins and engineering backgrounds to specify, model and simulate systems using VHDL [2]. While it is a common practice to teach a HDL to obtain synthesizable RTL (register transfer level) descriptions of digital circuits, here new capabilities of VHDL as an educational tool are explored. Students attain greater abilities in designing test-benches since they are based on the plant modeling. From a digital designer's perspective the test-bench file is complemented with the behavioral model

Manuscript received September 20, 2008. Accepted for publication December 23, 2009 Manuscript sent September 20, 2008.

F. J. Azcondo is with the University of Cantabria, ETSII y T, 39005 Santander Spain. (Phone: 34-942201546; fax: 34-942201873; e-mail: azcondo@ieee.org.)

A. de Castro is with the Universidad Autonoma de Madrid, 28049 Madrid Spain. (e-mail: angel.decastro@uam.es)

Ch. Brañas is with the University of Cantabria, ETSII y T, 39005 Santander Spain. (e-mail: branasc@unican.es)

A previous version of this paper was presented in the TAEE 2008 Conference, Zaragoza, Spain. This work has been supported by the TEC2008-01753 projects TIN2009-07333-E/TSI of the Spanish Ministry of Science and Innovation

of the plant being controlled. Once the whole system is described in VHDL no other software or design framework is required to complete the design and verification. In a second stage the students confront the design of digital controllers, first simulating the whole plant under the same framework and later through experimental verification, implementing the controller in a programmable logic device. In this way, students incorporate specification and design capabilities in integrated circuit controllers (concurrent) [3]-[5] as a complement to microprocessor-based controllers (sequential).

When required, the link with students' background is made using mathematical modeling software. In this case, the plant is first modeled and simulated using Matlab ® code or Simulink ® blocks. Later, the correspondence between Simulink ® blocks and basic VHDL descriptions is identified [6]. Initial examples use descriptions of switched power electronic converters, although the method can be generalized to other multivariable systems [7]. The VHDL model formalizes the specification under a non proprietary standard, independent of the simulation and synthesis software. Both system and initial controller models are behavioral. After obtaining the appropriate specification skills, the controller description is modified to make possible its synthesis.

Free access simulation software such as ModelSim-XE Starter ®, from Mentor Graphics ®, includes a graphic interface that enables both analog and digital representation of the circuit signals. Therefore, simulation results are straightforwardly analyzed as in any other continuous time oriented simulator.

The techniques that are adapted for training purposes here have been successfully applied in research projects. These projects are focused on modeling and controlling power converters connected to the utility [8]-[10] and applied to the control of electric machine drives [11]-[16].

Low-cost FPGA [17] evaluation boards allow students to be introduced to small to medium complexity digital designs in a one-semester course. These boards achieve enough speed performance to enable the students to be trained with cases similar to the ones found in real industrial applications.

II.- STUDENTS AND METHODOLOGY

The course was presented in this form for the first time during the academic year 2007-08, being duplicated in the first and second semesters. In the first semester the course was taught in Spanish and in the second in English within the Diploma in Advanced Applied Technologies. Three students participated in each term from different exchange programs, as shown in Fig. 1, which also includes the recently enrolled students in the second semester of the academic year 2008-09. The academic backgrounds of the students are diverse. None of them had previous knowledge of VHDL although one of them had previous experience with Verilog.

The training methodology is consistent with the guidelines of the European Area of Higher Education. Objectives cover the following aspects: Technical, methodological and social skills along with personal skills.

A. Technical skills

The course objectives are to acquire competence in: 1) VHDL systems modeling, starting from a mathematical description, 2) design and 3) implementation of medium complexity digital electronic systems using programmable devices with emphasis on the following aspects:

- Knowledge of the framework composed of different software tools, using both text and graphic interfaces. This is completed with an evaluation board, used to implement and verify the resulting designs.

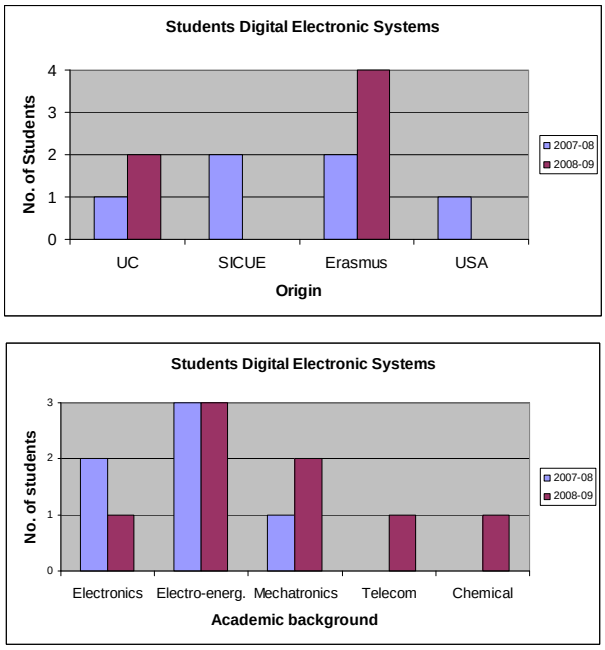

Fig. 1. Origin and academic background of the students.

- Determine the steps that the designer (engineer) should carry out during the system specification and modeling in order to obtain reliable control designs.

- Knowledge of the VHDL language elements.

- Translation of the mathematical models of the components of a system into VHDL descriptions.

- Analysis, synthesis and simulation of VHDL-described combinational and sequential circuits. Gain practical experience of basic concepts of digital electronics using VHDL.

The following points refer to advanced aspects of the course that allow the students to tackle medium complexity designs:

- Analysis, synthesis and simulation of synchronous digital circuits.

- Selection of suitable programmable hardware depending on the application's technical needs.

- Criteria to select the most suitable strategy for the description of state-machines.

- Design adaptation for implementation in the target programmable device.
- Clock signal distribution strategy and use of the "digital clock manager" (DCM) block.
- Definition of functional test for synchronous digital circuits.
- Use of the `loop` sentences to describe repetitive functions or functions applied to buses instead of single signals.
- Generalization of the circuit description by using `generic` and `generate` sentences.

B. Methodological skills

Technical details of the training methodology approach will be given in section III. Students develop a practical approach to the design of digital circuits using VHDL and their implementation in complex programmable logic devices, CPLDs, or field programmable gate arrays (FPGAs), motivated by the VHDL modeling of a system that will be controlled by the digital circuit. After a few theoretical classes on the VHDL language, students develop low-complexity digital circuits that will be components of the final controller. Students describe, simulate and synthesize digital circuits focused on possible real industry applications in the laboratory.

C. Social skills

The participation of the students is encouraged during the theoretical classes. They should share their proposals with the instructors and other students to discuss strong and weak points and consistency with specifications. Discussions on system and digital circuit specifications are an important issue in the participative classes. Students learn to reuse previously designed code and to take into account initial conditions and interaction with other circuits.

Depending on the number of students, one or more working groups are formed to solve the given specifications. The maximum number of students allowed in the course is sixteen. The practical case (project) is divided by the students into blocks whose modeling and design is done by two or three students. Class presentations motivate the work load coordination.

D. Personal skills

The Digital Electronic Systems course familiarizes engineering students with the definition of behavioral models using VHDL from previous mathematical models. The behavioral models in VHDL are defined under the same framework that will be used to design digital circuits focused on the control of the modeled systems (synthesizable models). Students develop practical digital electronics design skills, use of technical datasheets and search for useful information on the web. They also learn to organize laboratory resources and to verify circuits using a commercial development board.

E. Workload distribution

The course size is 5 European-Credit Transfer System Credits, ECTS, and the student workload is distributed as follows:

TABLE II
STUDENTS WORKLOAD DISTRIBUTION

Contact with instructors	45 hours
Class preparation	30 hours
Written documentation	20 hours
Work on oral presentations	20 hours
Work on preparing the final exam	10 hours

F. Assessment

After one month receiving instruction on VHDL and design tools, the students receive specifications of the plant to be modeled and controlled by the digital circuit. Based on the student's oral and written reports, the resulting performance, and the optimization of the VHDL description and digital circuit implementation, marks are given at the end of the course according to the following general criteria:
No satisfactory solution: fail.
Simulation of the basic performance: C.
Implementation in FPGA and functional verification: B.
Verified new contributions that improve the circuit performance: A.
Students are also required to write a self-assessment report. Some of the points highlighted by the students are:

- Basic background in engineering is essential to follow the classes.
- The course teaches the methods of engineering: making a plan of what to do, looking at the requirements that are given, and then solving the issues to carry out appropriate simulation and testing.
- The course improves the students' team skills.
- Troubleshooting the control circuitry is found to be faster, because a simulation is easier to set up than a complete circuit set-up.
- Good technical experience to bring back to their Universities.

III.- FRAMEWORK

Although the Digital Electronic Systems course is focused on digital electronics, particularly on VHDL, the goal presented to the students is to develop a digital controller of a given system (plant), which is not digital. Two objectives are followed with this technique. The first one is to motivate students from diverse backgrounds, presenting a goal that can be well understood with their previous knowledge. The second is to give the students some skills that are not usually included in a digital electronics course, such as analog systems modeling using VHDL and closed loop simulation of digital controllers. These skills can be very useful for students with diverse backgrounds (see section II). Students integrate

the description of the plant, the concurrent controller and the interfaces between them in a single simulation and design framework. As an illustrative example, a summary of a design presented to the students would be the following, showing the hierarchy of the system.

- Test bench. Integration of behavioral models and the digital controller for closed-loop simulation.
  - Plant under control (*non-synthesizable*). The system to be controlled is modeled through finite difference equations translated to VHDL.
  - Digital controller (*synthesizable*). This is the digital circuit that will be designed and implemented by the students.
  - Interface between the controller and the plant. Usually, it has both digital and analog parts:
    - Digital part of the interface (*synthesizable*). For example, the digital block that controls the ADCs.
    - Analog part of the interface (*non-synthesizable*). For example, the external ADCs.

It is important to highlight from the beginning which parts will be synthesizable and, therefore, implemented by the students in a programmable device, and which parts are only behavioral models. Description constraints for synthesis will be introduced in a second stage, motivating the learning of concepts about digital circuit design.

Since the learning process starts by understanding the plant model and its control requirements, the design approach is "top–down". The description process starts with a specification and an algorithmic description of the plant and the controller. Later, the controller and other synthesizable parts will be redesigned for synthesis. At this initial stage students learn about the different abstraction levels and the advantages that can be obtained from a concurrent description language.

The design strategy oriented to synthesis is learnt developing low-complexity codes for controller and data acquisition components. In these first steps, academic examples that correspond to small digital systems such as registers or counters are re-used. After the initial training, students propose a control algorithm initially oriented to evaluation purposes, which will be later adapted for synthesis. The introduction of the restrictions to make the synthesis possible is very suitable for training non specialist students in digital design, including the use of different types of objects in VHDL. The general idea is that the circuit is synthesizable when it is described using a clear RTL model. All the VHDL code of their synthesizable circuit must correspond to the structures used for basic digital components, which are re-used. Therefore, the students have a clear idea of the result of the synthesis process in their controllers.

According to the training sequence represented in Fig. 2, the instructor proposes the study of a plant with non trivial complexity, in order to show the need for standard specification with a software tool that enables the adequate verification of the model. The initial study of the plant is carried out with Matlab–Simulink ®. The system definition requires the identification of the state variables, the differential equations (state equations) and switch elements or conditional bifurcations. For extending the scope of the applications, a mathematical tool is preferred, but some students are used to working with SPICE whose *netlist* definition is a good starting point to identify the structural description style in VHDL.

The modeling process core consists in identifying the system components and obtaining their mathematical behavior, and then translating these models into VHDL. The description approach is top-down, beginning with the identification of inputs and outputs. Then the differential equations that define each component behavior are translated into finite difference equations using a sufficiently small time increment, Δt. This time increment imposes the clock frequency of the VHDL description that models the plant, this clock being a virtual one and not necessarily coincident with the clock of the part of the circuit described for synthesis. When the model requires the description of discontinuities, bifurcation sentences are used, as in the case of switch modeling.

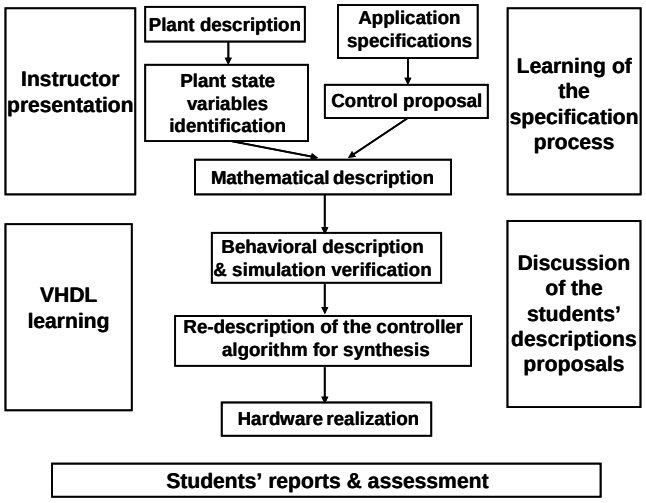

Fig. 2. Learning process flowchart.

Examples of modeling switched mode power supplies

To illustrate the proposed methodology, examples based on available laboratory prototypes are described. The plants are switched power converters that the students model. Controllers [18], whose specifications are proposed by the instructor, are designed by the students and practical results are verified experimentally.

1) Case of a resonant converter: As is shown in Fig. 3, the plant is composed of two switches and a resonant tank. The target application is to ignite a discharge lamp.

The circuit is solved using the mathematical software by connecting the models of each circuit component depicted in Fig. 4. The circuit switches are modeled by a switch element whose state depends on a control signal. A threshold level is defined for the control signal in order to determine whether the output corresponds to input one or input two. The control circuit generates the signal that defines the switches' states.

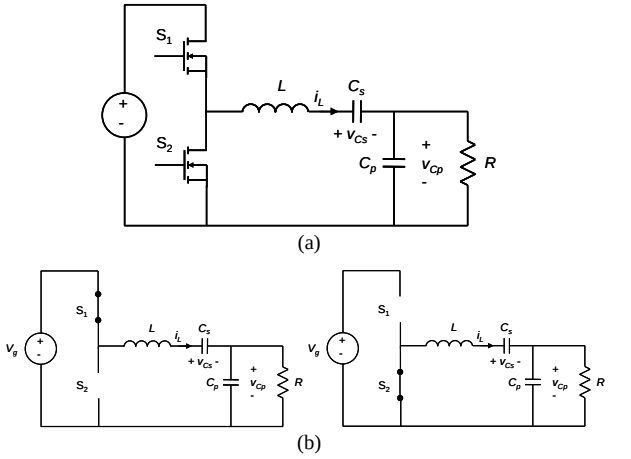

Fig. 3. (a) Class D LCC resonant converter. (b) States within a switching period. Left state 1, right state 2.

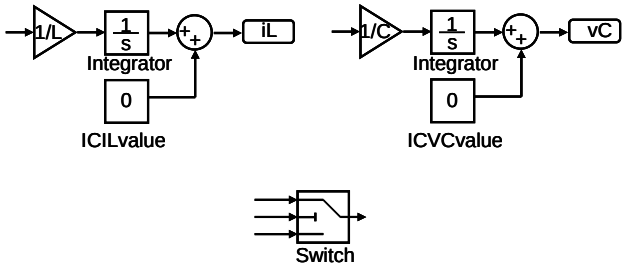

Fig. 4. Models of the resonant converter components.

Finite difference equations [19] for each component are obtained from the original differential equations and the circuit connections during states 1 and 2. In this case, equations (1) and (2) are valid for both states, equation (3) is for state 1 and equation (4) for state 2. The digital clock period should be equal to the time increment used in the finite difference equations, Δt, or at least it should be a multiple of Δt. This is the easiest way of making sure that the change from the equations of one state to another is coincident with the time in which the control signal (on-off) changes, and no drift takes place. It should be taken into account that the control signal is generated by the digital circuit and can change every clock period.

$$v_{Cs}[n] = v_{Cs}[n-1] + \frac{i_L[n-1]}{C_s}\Delta t \tag{1}$$

$$v_{Cp}[n] = v_{Cp}[n-1] + \frac{i_L[n-1] - \frac{v_{Cp}[n-1]}{R}}{C_p}\Delta t \tag{2}$$

$$i_L[n] = i_L[n-1] + \frac{V_g - v_{Cp}[n-1] - v_{Cs}[n-1]}{L}\Delta t \tag{3}$$

$$i_L[n] = i_L[n-1] + \frac{-v_{Cp}[n-1] - v_{Cs}[n-1]}{L}\Delta t \tag{4}$$

The switches that select the on and off state are translated into VHDL with bifurcation sentences e.g. `if … then … elsif … else … end if`. In this way, the translation from difference equations to VHDL code is quite straightforward, as shown in Fig. 5.

```
--Libraries and packages declaration
entity LCsCP_phaseModel is
   port( Vg    : in real;
         R     : in real;
         OnOff : in std_logic; -- On '1', off '0'
         Iin   : out real;
         Vo    : out real;
         Vcso  : out real);
end LCsCP_phaseModel;

architecture Behavioral of LCsCP_phaseModel is
  constant Il_init : real := 0.0;
  constant Vcs_init : real := 200.0;
  constant Vo_init : real := 0.0;
  signal Il : real := Il_init;
  signal Vcs : real := Vcs_init;
  signal Vcp : real := Vo_init;
  constant L : real := 225.0e-6;
  constant Rl : real := 0.5;
  constant Cs : real := 33.0e-9;
  constant Cp : real := 3.3e-9;
  constant dt2 : real := 1.0e-9;
  constant Cycle2 : time := 1 ns; -- Equal to dt2
begin
  Iin <= Il;
  Vo <= Vcp;
  Vcso <= Vcs;
  Calculation: process
  begin
    Vcs <= Vcs + Il*dt2/Cs;
    Vcp <= Vcp + (Il – Vcp/R)*dt2/Cp;
    if OnOff = '1' then
      Il <= Il + (Vg-Vcs-Vcp)*dt2/L;
    else
      Il <= Il + (-Vcs-Vcp)*dt2)/L;
    end if;
    wait for Cycle2; -- Time increment
   end process Calculation;
end Behavioral;
```

Fig. 5. VHDL model of a Class D LCC resonant converter.

The purpose of the controller is to generate the adequate state 1 to state 2 and vice-versa sequences (switching frequency) to ignite the lamp. The main controller inputs are the sign detection of the converter current and the lamp state (on – off). The lamp is modeled by *R*, as in fig. 3, or an open circuit when it is in the on and off states respectively.

A simulation of both, the converter and its controller models, is run in ModelSim ®. Both digital and analog signals can be displayed at the same time, as shown in Fig. 6, where from top to bottom the following signals can be observed: 1) input voltage, V_g, 2) equivalent resistance of the lamp, R, 3) digital signal that detects zero crossing of the resonant current, *ZeroCross*, 4) on-off signal for the high side switch, *OnOff*, 5) delayed version of on-off after the driver, *OnOffDel*, 6) resonant current in analog format, i_L, 7) output voltage in analog format, v_o, and 9) series capacitor voltage in analog format, v_{Cs}. The representation of both analog and digital signals together helps the students to understand the behavior of the whole system. Furthermore, any internal signal can be added to the simulation, which is very helpful for debugging the design, especially for inexperienced designers such as students.

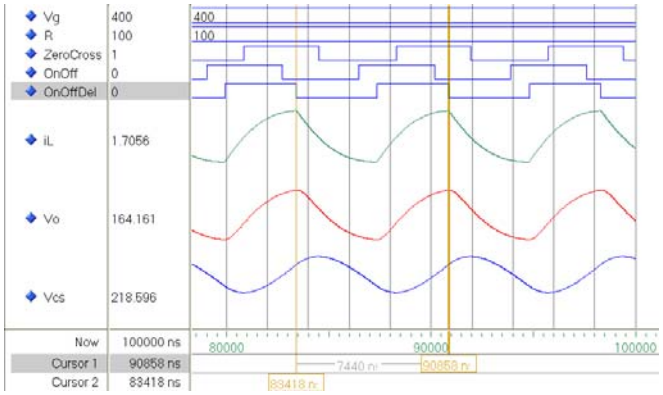

Fig. 6. Simulation results of the resonant converter using the digital design environment.

2) Case of a coupled inductor SEPIC converter: In this case students describe a more complex switched power converter, as is a SEPIC with coupled inductors, considering both continuous and discontinuous conduction modes (CCM or DCM). The model is more complex than in the resonant converter, so this converter is more appropriate for students with some experience in modeling systems. The converter, shown in Fig. 7, will act as a front-end power factor correction stage. For the sake of clarity, equations describing the converter behavior do not consider losses in the circuit components. Once the method is understood it would not be difficult to introduce parasitic effects.

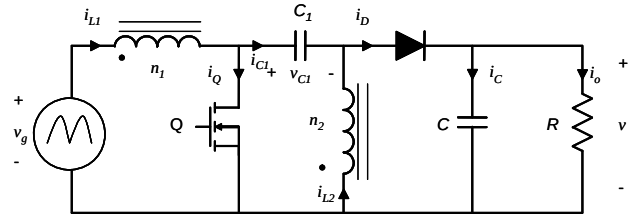

Fig. 7. Single-Ended Primary Inductor Converter (SEPIC).

Students receive the state equations for the different possible switch states: 1) Q on and D off, 2) Q off and D on, and 3) Q off and D off, which are defined by Eqs. (5) to (7) respectively. The coupled inductors have been defined as a function of the primary magnetic inductance, L_{m1}, the coupling coefficient, k, and the turn ratio $n=n_2/n_1$ equal to one. DCM appears during the Q off time if $i_{L1}+i_{L2}=0$.

$$\begin{bmatrix} \dot{i}_{L1} \\ \dot{i}_{L2} \\ \dot{v}_{C1} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 0 & -\dfrac{k^2}{L_{m1}(1-k^2)n} & 0 \\ 0 & 0 & \dfrac{k}{L_{m1}(1-k^2)n^2} & 0 \\ 0 & -\dfrac{1}{C_1} & 0 & 0 \\ 0 & 0 & 0 & -\dfrac{1}{RC} \end{bmatrix} \begin{bmatrix} i_{L1} \\ i_{L2} \\ v_{C1} \\ v \end{bmatrix} + \begin{bmatrix} \dfrac{k}{L_{m1}(1-k^2)} \\ -\dfrac{k^2}{L_{m1}(1-k^2)n} \\ 0 \\ 0 \end{bmatrix} [v_g] \quad (5)$$

$$\begin{bmatrix} \dot{i}_{L1} \\ \dot{i}_{L2} \\ \dot{v}_{C1} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 0 & -\dfrac{k}{L_{m1}(1-k^2)} & \dfrac{k^2-nk}{L_{m1}(1-k^2)n} \\ 0 & 0 & \dfrac{k^2}{L_{m1}(1-k^2)n} & \dfrac{nk^2-k}{L_{m1}(1-k^2)n^2} \\ \dfrac{1}{C1} & 0 & 0 & 0 \\ \dfrac{1}{C} & \dfrac{1}{C} & 0 & -\dfrac{1}{RC} \end{bmatrix} \begin{bmatrix} i_{L1} \\ i_{L2} \\ v_{C1} \\ v \end{bmatrix} + \begin{bmatrix} \dfrac{k}{L_{m1}(1-k^2)} \\ -\dfrac{k^2}{L_{m1}(1-k^2)n} \\ 0 \\ 0 \end{bmatrix} [v_g] \quad (6)$$

$$\begin{bmatrix} \dot{i}_{L1} \\ \dot{i}_{L2} \\ \dot{v}_{C1} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 0 & -\dfrac{k^2+nk}{2L_{m1}(1-k^2)n} & 0 \\ 0 & 0 & \dfrac{nk^2+k}{2L_{m1}(1-k^2)n^2} & 0 \\ \dfrac{1}{C1} & 0 & 0 & 0 \\ 0 & 0 & 0 & -\dfrac{1}{RC} \end{bmatrix} \begin{bmatrix} i_{L1} \\ i_{L2} \\ v_{C1} \\ v \end{bmatrix} + \begin{bmatrix} \dfrac{k^2+nk}{2L_{m1}(1-k^2)n} \\ -\dfrac{nk^2+k}{2L_{m1}(1-k^2)n^2} \\ 0 \\ 0 \end{bmatrix} [v_g] \quad (7)$$

The translation of the mathematical description of the different states into Simulink ® diagrams is intuitive, although it is a step given by the instructor for the students. For this case, a three-level hierarchy was preferred. The lowest level has the coupled inductor model, whose inputs are the voltages applied to the components' terminals and the outputs are the resulting currents, as shown in Fig. 8.

This model is a component of the next level, which corresponds to the SEPIC converter, including capacitors and switches modeled as described in Fig. 4 along with the coupled inductor block.

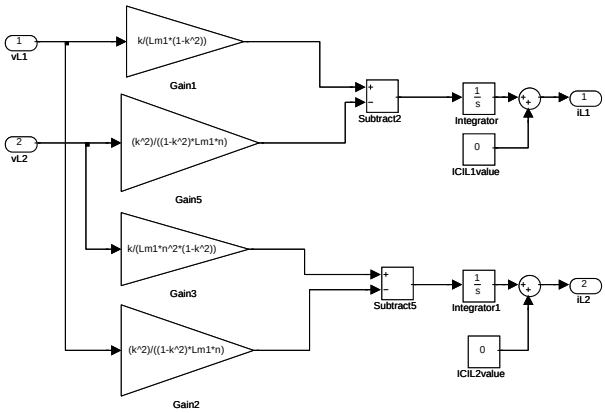

Fig. 8. Coupled inductor model.

The highest hierarchy level includes the SEPIC converter block connected to the rectified utility voltage, the controller drive signal and a resistive load. The voltage supply and the load descriptions will be translated to the test bench of the VHDL project. The design oriented to synthesis will be a digital control circuit.

VHDL codes corresponding to the described hierarchy levels are presented in Figs. 9 to 11. A difference between the models of the resonant and SEPIC converters is that the latter is presented in a hierarchical way due to its increased complexity. This is a good opportunity to introduce structural

VHDL descriptions to students, using components and hierarchies.

```
--Libraries and packages declaration
entity Coupled_inductor is
   port(
   inputs and outputs declaration);
end Coupled_inductor;

architecture Behavioral of Coupled_inductor is
   --constants and signals declaration
begin
   --declarations
   CALC_IL1: process -- State vbles are updated each integ period
   begin
   wait for Cycle2; -- Cycle2 defined in package, equal to dt2
   IL1_aux <= IL1_aux + ((VL1*(k/(Lm1*(1.0-k*k)))-
VL2*((k*k)/(n*Lm1*(1.0-k*k))))*dt2);
   IL2_aux <= IL2_aux + ((VL2*(k/(Lm1*(n*n)*(1.0-k*k)))-
VL1*((k*k)/(n*Lm1*(1.0-k*k))))*dt2);
   end process CALC_IL1;
end Behavioral;
```

Fig. 9. VHDL coupled inductor model.

```
-- Libraries and packages declaration
Entity CoupledSEPICModel is
  Port(inputs and outputs declaration);
End CoupledSEPICModel;

architecture Behavioral of CoupledSEPICModel is
   -- Component declaration for coupled inductor
   component Coupled_inductor
   port(inputs and outputs declaration);
   end component;
   constants and signals declaration
begin
   -- Instantiate the coupled inductor
   u1: Coupled_inductor port map(ports and signals mapping);
   CALC_VL1: process              -- voltage across L1
   begin
      wait for Cycle2;            -- Cycle2 defined in package
      if OnOff = '1' then         -- switch on
         Vl1 <= Vg;
      elsif (Id_off > 0.0) then         -- diode on
         Vl1 <= Vg - (V1 + VoAux);
      else
         Vl1 <= (Vg - V1)/2.0;
      end if;
   end process CALC_VL1;
   CALC_VL2: process              -- voltage across L2
   begin
      --calculation of the voltage across L2;
   end process CALC_VL2;
   CALC_V1: process               -- voltage across C1
   begin
      wait for Cycle2;            -- Cycle2 defined in package
      if OnOff = '1' then         -- switch on
         V1 <= V1 - ( (Il2*dt2) / C1 ); -- dt2 defined in package
      else
         V1 <= V1 + ((Il1*dt2) / C1);
      end if;
   end process CALC_V1;
   CALC_VoAux: process            -- voltage across C
   begin
      wait for Cycle2;            -- Cycle2 defined in package
      if OnOff = '1' then         -- switch on
         if Resist then         -- Resistive load
            VoAux <= VoAux - ( (VoAux/R)*dt2 / C );
         else
            VoAux <= VoAux - ( (Ir*dt2) / C );
         end if;
      else            -- switch off
         if Id_off > 0.0 then      -- diode on
            if Resist then      -- Resistive load
               VoAux <= VoAux + ( (Id_off - (VoAux/R))*dt2 / C );
            else
               VoAux <= VoAux + ( (Id_off - Ir) * dt2 / C );
            end if;
         else          -- DCM
            if Resist then      -- Resistive load
               VoAux <= VoAux - ( (VoAux/R)*dt2 / C ) ;
            else
               VoAux <= VoAux - ( (Ir*dt2) / C ) ;
            end if;
         end if;
      end if;
   end process CALC_VoAux;

   --other processes
end Behavioral;
```

Fig. 10. VHDL SEPIC model.

```
--Libraries and packages declaration
entity SEPICModelTb_vhd is
   generic (nbits: integer:=8); -- data resolution
end SEPICModelTb_vhd;

architecture Test of SEPICModelTb_vhd is

  -- Component Declaration for the Unit Under Test (UUT)
  component CoupledSEPICModel
  port(inputs and outputs declaration);
  end component;
  component PFCCtrl
  generic (inputs and outputs declaration);
  end component;

   --Constants and signals declaration;

Begin -- Instantiate the Unit Under Test (UUT)
  uut: CoupledSEPICModel port map(ports and signals mapping);

  TheCtrl: PFCCtrl port map(ports and signals mapping);

  Clk_process : PROCESS
   BEGIN
      --Clock process
   END PROCESS Clk_process;

   Init: PROCESS
   BEGIN
      --Reset process
   END PROCESS Init;

  VGGENERATOR: process
  -- Utility voltage, Vg, is generated using the function “sin” in
  -- DWMath package
    variable VgAux : real;
  begin
    t <= 0.0; -- It starts at the beginning of the cycle
    loop
      wait for Cycle2;
      t <= t + dt2;
      VgAux := VgMax*sin(2.0*pi*50.0*t);
      if VgAux > 0.0 then
        Vg <= VgAux;
      else
        Vg <= -VgAux;
      end if;
    end loop;
  end process VGGENERATOR;

  LOAD : process
  begin
    R <= (300.0**2)/150.0; --150.0 W
    Ir <= 0.0; -- P = Us*Ir = 0 W
    Resist <= true; -- Resistive load, R takes effect, Ir not
    wait for 100 ns;
    -- R receives new values
    wait; -- permanent wait
  end process LOAD;
  -- other processes

end Test;
```

Fig. 11. Test-bench file.

IV.- HARDWARE PLATFORM

At the beginning of the course, the digital controllers are described as a set of algorithms. However, once students have seen how to describe simple digital blocks in synthesizable VHDL, the controller is described in this way. At this point, the key is to make the students think about the RTL scheme of their controller. They have to think about how to implement their control algorithms using digital blocks, such as multiplexers, counters etc... Once the RTL scheme is clear, synthesizable code is obtained easily following the code examples of the basic digital blocks that have been presented during the course. Furthermore, this methodology is also an introduction to code re-use.

Depending on the plant (resonant or SEPIC converter), the interface between the controller and the plant is different. In the case of the resonant converter, a closed-loop phase controller can be created using only binary signals such as the input current sign detection (*ZeroCross* in Fig. 6). In this case, the interface is very simple, consisting of only voltage level

adaptation for the sensors. In the case of the SEPIC converter, closed-loop operation requires the use of ADCs. However, open-loop operation avoids the ADCs. It should be pointed out that this is not a control theory course, but a course of digital electronics. Any controller that includes a wide variety of digital blocks is valid, whether it is closed-loop or not.

Once the VHDL controller description is validated for synthesis, students implement the circuit to verify its operation completing the academic design cycle. The hardware chosen for the laboratory practical is a Xilinx ® Spartan-3E FPGA from Xilinx. Among the available hardware resources the main device is the XC3S500E Spartan-3E FPGA, that includes 232 pins configurable as an input or output and more than 10,000 logic cells. The project development board also has A/D and D/A converters, different clock sources, among them a 50 MHz on-board oscillator, switches, LEDs, a 16-character LCD display and PS/2 and VGA communication ports among other resources that enable the practical work to be done while minimizing the use of external components. Not all the components on the board are used in every case, such as the A/D and D/A converters. However, including a wide set of resources allows the board to be used in different designs.

The design implementation using the chosen hardware requires that the students learn to use the framework ISE Foundation (Xilinx), where the available software tools required to complete the design sequence are: initial debugging, behavioral simulation, synthesis, post synthesis simulation and final device configuration. Both behavioral and post-routing simulations are introduced. The first one is recommended for initial debugging, while the latter avoids delay or synchronization problems, very difficult to identify in the prototype. Using both simulations is the best way to ensure correct operation of the hardware circuit from the first use.

An example of the resulting practical work is shown in Fig. 12, where the ignition lamp demonstration can be observed. Two oscilloscopes are used; one for the digital control signals and another for the outputs of the power section.

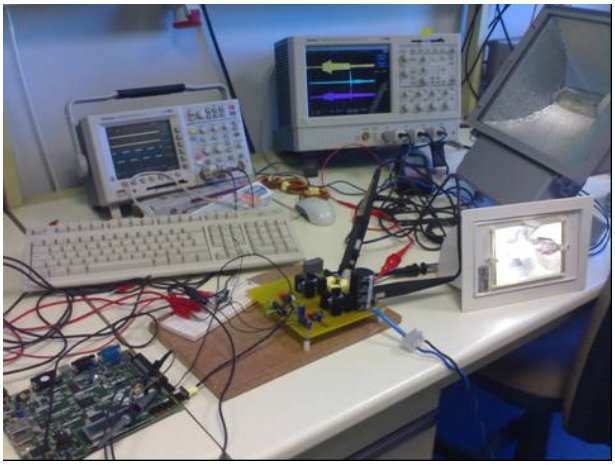

Fig. 12. Example of laboratory test-bench.

V.- CONCLUSIONS

The new contents and training methodology for a Digital Electronic Systems course have been presented, illustrated with practical cases of power converters, where students are trained to model diverse systems using VHDL starting from models previously described with a mathematical tool such as Matlab/Simulink ®. System modeling provides motivation for learning digital circuit design oriented to the control of the previously modeled systems. The design is developed under a single digital design framework. This framework enables the simulation of analog and digital systems by translating the systems' differential equations into finite difference equations. Then the direct correspondence between the mathematical blocks and VHDL elements is determined.

Teaching experience indicates that this approach increases the students' skills in designing test-benches, since they are not initially conceived as a group of test vectors but as the behavioral model of the plant under control.

On the other hand, the adaptation of digital-design oriented tools to modeling and simulating mixed signal systems is an extra effort that includes the definition of the time of the integration step, acceptance of rather long simulation times and familiarization with non-trivial details of the simulation tool in order to configure the representation of the analog signals. However, alternatives such as the originally conceived mixed-signal design framework are not so accessible for the students' characteristics of this course.

The new approach allows the course to be offered in the context of a new Diploma in Advanced Applied Technologies for Industry addressed to multidisciplinary engineering students who are not necessarily specialized in electronics, coming from diverse universities and backgrounds. Student exchange programs are therefore promoted with this activity. The resulting digital circuit designs are as similar as possible to real industrial application cases. Students are provided with a practical and efficient alternative to microcontroller-based control techniques.

REFERENCES

[1] Diploma in Advanced Applied Technologies for Industry, Univ. of Cantabria program. Available: http://www.unican.es/Centros/etsiit/daati/.

[2] T. Riesgo, Y. Torroja, and E. de la Torre, "Design methodologies based on hardware description languages," IEEE Trans. Ind. Electron., vol. 46, no. 1, pp. 3–12, Feb. 1999.

[3] R. Dubey, P. Agarwal, and M. K. Vasantha, "Programmable logic devices for motion control—a review," IEEE Trans. Ind. Electron., vol. 54, no. 1, pp. 559–566, Feb. 2007.

[4] E. Monmasson, and M. N. Cirstea, "FPGA design methodology for industrial control systems—a review," IEEE Trans. Ind. Electron., vol. 54, no. 4, pp. 1824–1842, Aug. 2007.

[5] L. Corradini, P. Mattavelli, E. Tedeschi, and D. Trevisan, "High-bandwidth multisampled digitally controlled dc–dc

converters using ripple compensation," IEEE Trans. Ind. Electron., vol. 55, no. 4, pp. 1501–1508, Apr. 2008.
[6] S. P. Valsan, K. S. Swarup, "High-Speed Fault Classification in Power Lines: Theory and FPGA-Based Implementation," IEEE Trans. on Industrial Electronics, vol. 56, no. 5, pp. 1793-1800, May 2009.
[7] A Dinu, M Cirstea, S Cirstea, "Direct Neural Networks Hardware Implementation Algorithm," IEEE Trans. on Industrial Electronics, to be published.
[8] A. de Castro, P. Zumel, O. Garcia, T. Riesgo, and J. Uceda, "Concurrent and simple digital controller of an AC/DC converter with power factor correction based on an FPGA," IEEE Trans. Power Electron., vol. 18, no. 1, pp. 334–343, Jan. 2003.
[9] A. de Castro, "Custom hardware digital control applied to switching power converters," Ph.D. dissertation, Dept. Elect. Eng., Univ. Politecnica de Madrid, Spain, 2004.
[10] Zeliang Shu, Yuhua Guo, Jisan Lian, "Steady-State and Dynamic Study of Active Power Filter With Efficient FPGA-Based Control Algorithm," IEEE Trans. on Industrial Electronics, vol. 55, no. 4, pp. 1527-1536, April 2008.
[11] A. Aounis, S. E. Cirstea, and M. N. Cirstea, "Reusable VHDL architectures for induction motor PWM vector control, targeting FPGAs," in Proc. IEEE IECON, 2006, pp. 4923–4928.
[12] M. N. Cirstea, "VHDL for industrial electronic systems integrated development," in Proc. IEEE ISIE, 2006, pp. 1516–1520.
[13] M. W. Naouar, A. A. Naassani, E. Monmasson, and I. Slama-Belkhodja, "FPGA-Based Predictive Current Controller for Synchronous Machine Speed Drive," IEEE Trans. Power Electron., vol. 23, no. 4, pp. 2115–2126, Jul. 2008.
[14] F.-J. Lin, L.-T. Teng, C.-Y. Chen, and C.-K. Chang, "Robust RBFN control for linear induction motor drive using FPGA," IEEE Trans. Power Electron., vol. 23, no. 4, pp. 2170–2180, Jul. 2008.
[15] Da Zhang, Hui Li, "A Stochastic-Based FPGA Controller for an Induction Motor Drive With Integrated Neural Network Algorithms," IEEE Trans. on Industrial Electronics, vol. 55, no. 2, pp. 551-561, Feb 2008.
[16] L. Idkhajine, E. Monmasson, M. W. Naouar, A. Prata, K. Bouallaga, "Fully Integrated FPGA-Based Controller for Synchronous Motor Drive," IEEE Trans. on Industrial Electronics, vol. 56, no. 10, pp. 4006-4017, Oct 2009.
[17] J. J. Rodriguez-Andina, M. J. Moure, and M. D. Valdes, "Features, design tools, and application domains of FPGAs," IEEE Trans. Ind. Electron., vol. 54, no. 4, pp. 1810–1823, Aug. 2007.
[18] M. Zamora, H. Wu, M. P. Henry, "An FPGA Implementation of Frequency Output," IEEE Trans. on Industrial Electronics, vol. 56, no. 3, pp. 648-653, March 2009.
[19] H. Levy, F. Lessman, Finite Difference Equations. New York: Dover Publications, 1992.

Francisco J. Azcondo (S'90–M'92–SM'00) was born in Santander, Spain, in 1965. He received the degree in electrical engineering from the Universidad Politecnica de Madrid, Madrid, Spain, in 1989, and the Ph.D. degree from the University of Cantabria, Santander, in 1993. From 1990 to 1995, he worked in the design of highly stable quartz crystal oscillators. Since 1995, he has been an Associate Professor in the Department of Electronics Technology, Systems and Automation Engineering, University of Cantabria. From February to August 2004, he was a Special Member of the Department of Electrical and Computer Engineering, University of Colorado, Boulder, and in the summer of 2006, he was a Visiting Researcher in the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada. His research interests include switch-mode power converters and their control for discharge lamps, electrical discharge machining, arc welding, and power factor correction applications.

Angel de Castro (M'08) was born in Madrid, Spain, in 1975. He received the M.Sc. and the Ph.D. degrees in electrical engineering from the Universidad Politecnica de Madrid, Madrid, Spain, in 1999 and 2004, respectively.
He has been an Assistant Professor in the Universidad Autonoma de Madrid since 2006. Previously, he was an Assistant Professor in the Universidad Politecnica de Madrid since 2003. His research interests include digital control of switching mode power supplies, field programmable gate arrays and mobile nodes in wireless sensor networks.

Christian Brañas (M'03) received the M.S. degree in electronics engineering from the Instituto Superior Politécnico Jose A. Echeverria (ISPJAE), Havana, Cuba, in 1992, and the Ph.D. degree in electronics engineering from the University of Cantabria, Santander, Spain, in 2001. From 1992 to 1995, he was an Instructor Professor with the ISPJAE. Since 1995, he has been with the Department of Electronics Technology, Systems and Automation Engineering, University of Cantabria, where he is currently an Associate Professor. His research interests include the design, modeling, and control of resonant inverters and switch-mode power supplies.