

ORIENTACIÓN DE LA ASIGNATURA SISTEMAS ELECTRÓNICOS DIGITALES AL MODELADO DE SISTEMAS EN VHDL PARTIENDO DE ESQUEMAS MATLAB-SIMULINK

FRANCISCO J. AZCONDO¹, ÁNGEL DE CASTRO² CHRISTIAN BRAÑAS¹

¹*Grupo de Ingeniería Microelectrónica Departamento de Tecnología Electrónica, Ingeniería de Sistemas y Automática. Escuela Técnica Superior de Ingenieros Industriales y de Telecomunicación. Universidad de Cantabria. España.*

²*Departamento de Ingeniería Informática. Escuela Politécnica Superior. Universidad Autónoma de Madrid. España.*

javier.azcondo@unican.es, angel.decastro@uam.es branasc@unican.es

Se propone una nueva orientación para la asignatura Sistemas Electrónicos Digitales donde se enseñe a los alumnos a modelar sistemas en VHDL partiendo de esquemas desarrollados en Matlab/Simulink para posteriormente abordar propuestas de controladores digitales sintetizables integrados en la descripción del sistema. De esta forma, la asignatura no sólo se dirige a alumnos de la intensificación sino que se puede incorporar a currículos multidisciplinares dentro de los estudios de ingeniería.

1. Introducción

La asignatura optativa Sistemas Electrónicos Digitales de quinto curso del Plan de Estudios de Ingeniero Industrial de la Universidad de Cantabria se concibe inicialmente como una asignatura de especialización, en el contexto de la Intensificación Automática y Electrónica Industrial, que dé continuidad a la asignatura obligatoria Ampliación de Sistemas Electrónicos, donde se imparten los fundamentos de Electrónica Digital. De esta forma, los contenidos se han venido orientando al diseño de sistemas digitales: combinatoriales, secuenciales, a nivel de transferencia de registros y a nivel de procesador, con ejemplos bien descritos en VHDL o desarrollados para microcontroladores PIC. Desde hace ya varios años, el perfil de los alumnos que eligen la asignatura no corresponde al de especialistas en Tecnología Electrónica, sino al de Ingeniero Industrial sin especialidad, y es habitual que se incorporen alumnos Erasmus, Séneca y de convenios bilaterales con Latinoamérica. En este curso 2007-08, esta asignatura se incorpora a un nuevo programa, según se describe en la Fig. 1 y en [1], denominado “Diploma in Advanced Applied Technologies for Industry” que se impartirá en inglés, y se hace necesario orientar los contenidos a cualquier alumno del ámbito de la ingeniería. En definitiva, se propone hacer más coherente la asignatura con el perfil del alumnado y prepararla para recibir alumnos con estudios previos de diferente procedencia. Para ello, la asignatura se centrará en dotar a los alumnos con capacidades para especificar y modelar sistemas con el lenguaje VHDL, para luego abordar el diseño de controladores digitales. Esta técnica es útil en diversos aspectos de la ingeniería y presenta las técnicas de control digital orientadas a su implementación en circuitos concurrentes (FPGA o ASIC), como complemento al control secuencial realizado por microprocesador (computador), más extendido en el control de elementos y procesos industriales.

El enlace con los conocimientos asumibles por los alumnos se realiza partiendo de la descripción en código Matlab o en bloques Simulink y los ejemplos iniciales tienen como contenido la descripción de topologías de convertidores electrónicos de potencia conmutados, si bien el método es generalizable para cualquier tipo de sistema multivariable. La descripción VHDL permite formalizar la especificación bajo un estándar independiente de la herramienta de simulación y en su caso de síntesis, creando códigos de tamaño muy reducido. Simuladores como ModelSim, con versiones de libre acceso, presentan la opción de representar datos tanto analógicos como digitales para evaluar fácilmente los resultados.

Precedentes de estas técnicas han sido utilizados con éxito en trabajos de investigación orientados al modelado y control de convertidores electrónicos de potencia conectados a la red eléctrica [2], y aplicados al control de máquinas eléctricas [3-4]. Las actuales tarjetas de evaluación de FPGAs permiten abordar diseños de cierta complejidad y buenas prestaciones de velocidad con un coste reducido y son muy adecuadas para la realización de prácticas en la asignatura, acercándose a soluciones muy cercanas a posibles casos planteados por la industria.

<p>Diploma in Advanced Applied Technologies for Industry</p> <p>Program Description 30 ECTS credits of Elective and Optional Subjects:</p> <p>5657 Spanish History and Culture for Engineers (5 ECTS) 5652 Electronics Circuits and Devices (5 ECTS) 5655 Digital Electronics Systems (5 ECTS) 5651 Advanced Machine Design (5 ECTS) 5654 Perception Systems (5 ECTS) 5653 Modelling and Simulation of Dynamical Systems (5 ECTS) 5656 Spanish Language for Engineers (5 ECTS)</p>
--

Figura 1. Programa del Diploma “Advanced Applied Technologies for Industry”.

2. Entorno de trabajo

La idea general que se plantea para la asignatura Sistemas Electrónicos Digitales es conseguir que los alumnos adquieran experiencia de modelado, simulación y reutilización de código con un único lenguaje de diseño como es VHDL que responde a un estándar IEEE y mediante el cual se pueden integrar tanto el modelo de “la planta” como el diseño del controlador, junto con las interfaces de conexión entre ellos. El esquema del entorno de trabajo se completa con la descripción que modela la alimentación, señales de entrada, (estímulos) y lee las salidas del sistema y las de interés para su verificación. A modo de ejemplo, un entorno de trabajo puede responder al presentado en el siguiente esquema.

- Banco de pruebas
 - Planta a controlar
 - Componente 1 de la planta
 - Componente 2 de la planta
 - ...
 - Etc.
 - Controlador de la planta (sintetizable)
 - Interfaz con la planta
 - Componentes de la interfaz
 - Adquisición de datos
 - Componentes de la adquisición de datos
 - Algoritmos de control
 - Componentes de los algoritmos

Figura 2. Esquema de organización de los ficheros de un proyecto descrito en VHDL.

Si el modelo descrito está orientado a obtener la síntesis del controlador de la planta, es conveniente presentar una modificación de este esquema distinguiendo la parte a realizar (implementar) en un circuito digital, que tendrá un tratamiento de descripción distinto para adaptarse al hardware de destino.

- Banco de pruebas
 - Planta a controlar
 - Componentes de la planta
 - Controlador digital (Parte sintetizable)
 - Componentes del circuito digital
 - Componentes digitales de la interfaz
 - Parte no sintetizable de la interfaz

Figura 3. Esquema de organización distinguiendo la parte sintetizable y no sintetizable.

El objetivo es que el alumno realice un planteamiento “top – down” del sistema haciendo inicialmente énfasis sobre la correcta especificación y solución del control, quedando en segundo término la posible síntesis del algoritmo de control. En este apartado se presentan los diferentes niveles de abstracción y estilos o estrategias de descripción y se realizan consideraciones sobre las ventajas para el modelado de sistemas que supone disponer de un lenguaje de descripción concurrente. Una vez que se conocen las sentencias VHDL, se revisa la característica concurrente de la descripción con ejemplos de extrapolación de funciones utilizando la sentencia “loop”.

La estrategia de diseño orientado a la síntesis se aprende desarrollando componentes del sistema realizados con códigos simples, que provienen de la reutilización de ejemplos académicos, que responden a pequeños sistemas digitales; como registros, contadores, etc., que son componentes de la adquisición de datos. A partir de este entrenamiento, el alumno puede plantear un algoritmo de control, inicialmente orientado a su evaluación y posteriormente introducir las restricciones para conseguir una descripción sintetizable. La traducción de descripción mediante algoritmo no sintetizable a código sintetizable resulta muy formativa para introducir a alumnos no especialistas en el diseño digital a la vez de para ejercitar la utilización de los tipos de objetos en VHDL.

Se parte de una planta, de complejidad no trivial, para introducir la necesidad de estandarizar la especificación mediante algún tipo de herramienta que permita evaluarla. Para realizar el estudio de la planta se parte de un modelo realizado en Matlab – Simulink, donde se identifiquen las variables de estado y las ecuaciones diferenciales que las relacionan, así como los elementos de conmutación o bifurcaciones condicionales del sistema. Se prefiere una herramienta matemática por su generalidad, pero lógicamente para alumnos familiarizados con SPICE, el “netlist” que se obtiene con esta herramienta es un buen punto de partida para identificarlo con el estilo de descripción estructural y realizar una división del proyecto en sus componentes que se modelarían por separado.

El núcleo del proceso de modelado de la planta consiste, precisamente, en identificar los componentes del sistema, extraer su comportamiento matemático y resolver su descripción en VHDL. Para ello, se realiza una aproximación “top-down” a cada componente determinando, inicialmente, sus entradas y salidas. Como idea general, los comportamientos descritos mediante ecuaciones diferenciales son traducidos a ecuaciones en diferencias con un incremento de tiempo suficientemente pequeño. La especificación de incremento de tiempo impone la frecuencia de reloj a utilizar por la descripción VHDL que modela la planta. Este reloj es virtual y no tiene por qué coincidir con la señal de reloj de la descripción objeto de una síntesis posterior. En un modelado algorítmico, las situaciones de discontinuidad se resuelven por sentencias de bifurcación, que permiten modelar conmutadores.

2.1. Ejemplo de modelado de fuentes conmutadas

Para ilustrar el método propuesto se toma un ejemplo disponible en el laboratorio con el que poder ilustrar los pasos de modelado y contrastar resultados. En este caso, se aborda la descripción de una fuente conmutada de cierta complejidad, como es el convertidor SEPIC con inductancias acopladas y

considerando los posibles casos de modo de conducción continua y discontinua. El convertidor actúa como corrector de factor de potencia. El esquema del convertidor se muestra en la Fig.4. Por claridad se presentan las ecuaciones que definen el comportamiento en el caso ideal; donde se consideran inductancias y condensadores ideales (sin pérdidas) y tampoco se consideran pérdidas de conducción ni conmutación en los semiconductores. Una vez presentado el método, no es difícil considerar estos elementos de pérdidas.

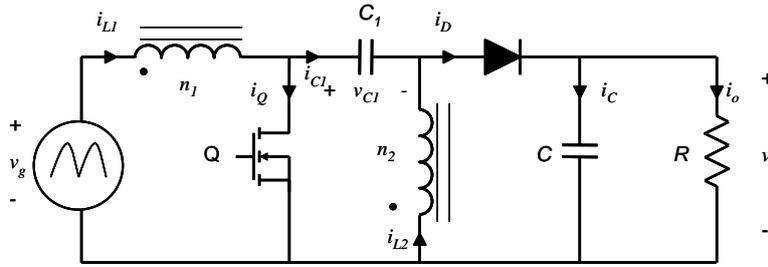


Figura 4. Convertidor SEPIC.

Para modelar esta planta se definen las ecuaciones de estado en las diferentes situaciones de conmutación y las condiciones que imponen que el transistor Q conduzca, que el diodo conduzca o bien que ninguno de los dos semiconductores conduzca. A modo de ejemplo se presentan las ecuaciones de estado durante el tiempo de conducción del transistor (tiempo de ON) y del diodo (tiempo de OFF) en las Ec. (1) y (2) respectivamente.

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

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

donde las inductancias acopladas se han definido en función de la inductancia de magnetización vista desde el primario, L_{m1} , el coeficiente de acoplamiento k y la relación de transformación $n=n_2/n_1$, que en este caso es unidad. La situación de conducción discontinua se produce si durante el tiempo de OFF se da la condición $i_{L1}+i_{L2}=0$ y el convertidor queda definido por (3).

Una vez descritos los diferentes estados de un sistema mediante expresiones matemáticas, la traducción a un esquema Simulink es directa e intuitiva. En este caso se ha preferido emplear una pequeña estructura con jerarquía de tres niveles. En el nivel inferior aparece el modelo de las inductancias

acopladas utilizando como entradas las tensiones aplicadas a los devanados y obteniendo como salida las intensidades de los devanados, según se indica en la Fig. 5.

$$\begin{bmatrix} \dot{i}_{L1} \\ \dot{i}_{L2} \\ \dot{v}_{C1} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 0 & -\frac{k^2 + nk}{2L_{m1}(1-k^2)n} & 0 \\ 0 & 0 & \frac{nk^2 + k}{2L_{m1}(1-k^2)n^2} & 0 \\ \frac{1}{C1} & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{RC} \end{bmatrix} \begin{bmatrix} i_{L1} \\ i_{L2} \\ v_{C1} \\ v \end{bmatrix} + \begin{bmatrix} \frac{k^2 + nk}{2L_{m1}(1-k^2)n} \\ -\frac{nk^2 + k}{2L_{m1}(1-k^2)n^2} \\ 0 \\ 0 \end{bmatrix} [v_g] \quad (3)$$

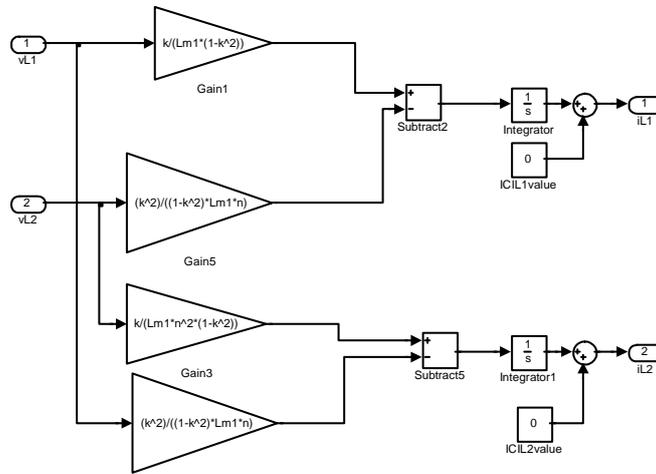


Figura 5. Modelo de inductancias acopladas

Este modelo se empleará como componente del siguiente nivel, representado en el Fig. 6, que corresponde al convertidor SEPIC.

En este nivel (SEPIC) aparecen, junto con el bloque de inductancias acopladas, los condensadores y los interruptores. Los condensadores se modelan mediante una ganancia inversa al valor de capacidad y conectada al valor de intensidad del condensador (entrada) y un elemento integrador cuya salida (incremento de tensión) se suma al valor de tensión inicial del condensador. El valor resultante es la tensión del condensador (salida). Los interruptores se modelan con conmutadores cuyo estado depende de una señal de control sobre la que se define un umbral. Dependiendo del valor de la señal de control en relación con el umbral, la situación del conmutador corresponde a uno de los dos estados posibles. El valor de la señal de control se obtiene del circuito de control que se encuentra en el nivel superior o bien de la condición que detecta modo de conducción continua o discontinua.

En el nivel superior, presentado en la Fig. 7, el convertidor SEPIC es el componente al que se conectan la alimentación, la carga y el circuito de control. En este nivel se definen, por tanto, los estímulos y se obtienen las salidas y corresponderá, por tanto, al banco de pruebas (test-bench) de la descripción VHDL. Las Figs. 5 y 6 muestran el modelo Matlab/Simulink del convertidor SEPIC utilizando las Ecs. (1) a (3) para definir el comportamiento durante el tiempo de ON y OFF, incluyendo el caso de conducción discontinua. En la Fig. 7 se presenta el bloque que corresponde al convertidor SEPIC bajo el gobierno de un controlador, alimentado con la tensión de red rectificadas y conectando a la salida una carga resistiva.

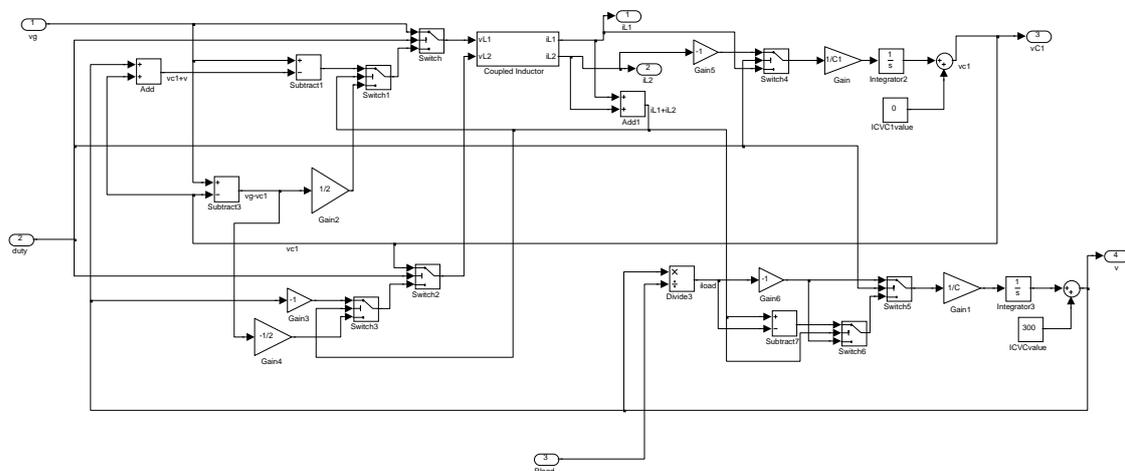


Figura 6. Modelo del convertidor SEPIC

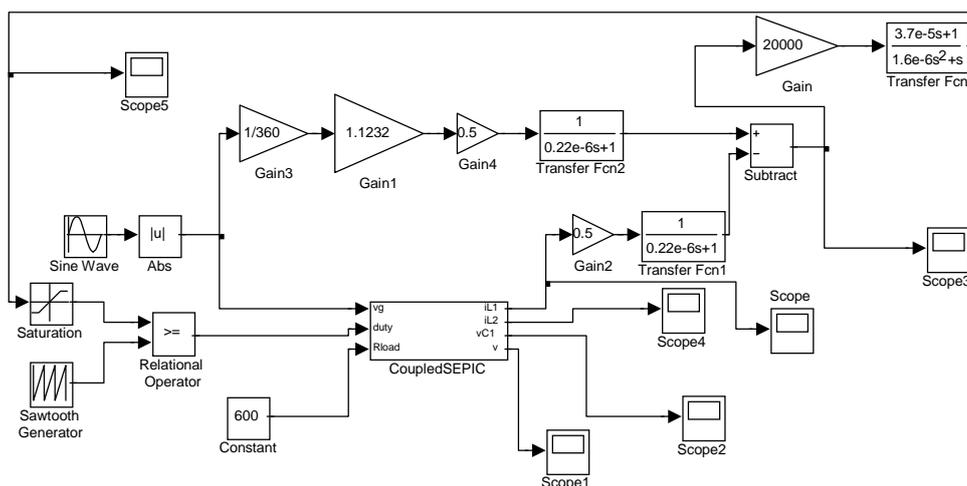


Figura 7. Propuesta de adquisición de datos, controlador analógico y estímulos para el modelo SEPIC

La traslación a un modelo de comportamiento en VHDL requiere traducir las ecuaciones diferenciales de cada estado a ecuaciones en diferencias finitas [5]. Un periodo de reloj suficientemente pequeño se utiliza como el incremento de tiempo que permite pasar del dominio continuo al dominio discreto. En general y por simplicidad, se puede hacer coincidir este periodo de reloj de discretización con el periodo de reloj del circuito digital que implementa el control de la planta, aunque no es obligatorio.

De esta forma se simulan tanto la planta y el controlador con una misma señal de sincronismo. Los conmutadores que seleccionan el estado vigente en cada instante se traducen a sentencias de bifurcación del tipo *if...then...elsif...else...end if*. Los sub-bloques reincorporan al modelo VHDL *instanciando* componentes descritos en diferentes ficheros.

A continuación se muestran códigos VHDL que representan los esquemas de las Figs. 5 – 7.

```

Declaración de bibliotecas y paquetes
entity Coupled_inductors is
  port(
    declaración de entradas y salidas);
end Coupled_inductors;

architecture Behavioral of Coupled_inductors is
  declaración de constantes y señales
begin
  declaraciones
  CALC_IL1: process -- Cada periodo de integración se actualizan las variables de estado
  begin
    wait for Cycle2; -- ver Cycle2 en package UAM_UC_Pack
    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); -- ver dt2 en package
  end process CALC_IL1;
end Behavioral;

```

Figura 6. Modelo de inductancias acopladas

```

--Declaración de bibliotecas y paquetes
Entity CoupledSEPICModel is
  Port( declaración de entradas y salidas);
End CoupledSEPICModel;

architecture Behavioral of CoupledSEPICModel is
  -- Component Declaration for coupled inductors
  component Coupled_inductors
  port( declaración de entradas y salidas);
  end component;
  declaración de constantes y señales
begin
  -- Instantiate the coupled inductors
  ul: Coupled_inductors port map(conexión de señales y puertos);
  CALC_VL1: process -- Tensión en L1
  begin
    wait for Cycle2; -- ver Cycle en package UAM_UC_Pack
    if OnOff = '1' then -- interruptor cerrado
      V11 <= Vg;
    elsif (Id_off > 0.0) then -- diodo conduce
      V11 <= Vg - (V1 + VoAux);
    else
      V11 <= (Vg - V1)/2.0;
    end if;
  end process CALC_VL1;
  CALC_VL2: process -- Tensión en L2
  begin
    determinación de la tensión en L2;
  end process CALC_VL2;
  CALC_V1: process -- Tensión en C1
  begin
    wait for Cycle2; -- ver Cycle en package UAM_UC_Pack
    if OnOff = '1' then -- interruptor cerrado
      V1 <= V1 - ( (IL2*dt2) / C1 ); -- ver dt en package UAM_UC_Pack
    else
      V1 <= V1 + ((I11*dt2) / C1);
    end if;
  end process CALC_V1;
  CALC_VoAux: process -- Tensión en C
  begin
    wait for Cycle2; -- ver Cycle2 en package UAM_UC_Pack
    if OnOff = '1' then -- interruptor cerrado
      if Resist then -- Carga resistiva
        VoAux <= VoAux - ( (VoAux/R)*dt2 / C );
      else
        VoAux <= VoAux - ( (Ir*dt2) / C );
      end if;
    else -- interruptor abierto
      if Id_off > 0.0 then -- diodo conduce
        if Resist then -- Carga resistiva
          VoAux <= VoAux + ( (Id_off - (VoAux/R))*dt2 / C );
        else
          VoAux <= VoAux + ( (Id_off - Ir) * dt2 / C );
        end if;
      else -- Conducción discontinua
        if Resist then -- Carga resistiva
          VoAux <= VoAux - ( (VoAux/R)*dt2 / C );
        else
          VoAux <= VoAux - ( (Ir*dt2) / C );
        end if;
      end if;
    end if;
  end process CALC_VoAux;

  otras declaraciones
end process;
end Behavioral;

```

Figura 7. Modelo del convertidor SEPIC

```

Declaración de bibliotecas y paquetes
entity SEPICModelTb_vhd is
    generic (nbits: integer:=8); -- programa la resolución del dato
end SEPICModelTb_vhd;

architecture Test of SEPICModelTb_vhd is

    -- Component Declaration for the Unit Under Test (UUT)
    component CoupledSEPICModel
    port(declaración de entradas y salidas);
    end component;

    component PFCCtrl
    generic (declaración de entradas y salidas);
    end component;

    Declaración de señales y constantes;

Begin -- Instantiate the Unit Under Test (UUT)
    uut: CoupledSEPICModel port map(conexión de señales y puertos);

    TheCtrl: PFCCtrl port map(conexión de señales y puertos);
    Relojprocess : PROCESS
    BEGIN
        Proceso del reloj;
    END PROCESS Relojprocess;

    Inicio: PROCESS
    BEGIN
        Proceso del reset;
    END PROCESS Inicio;

    VGGENERATOR: process
    -- Se genera Vg utilizando la funcion "sin" de DWMath
    variable VgAux : real;
    begin
        t <= 0.0; -- Empieza al principio de ciclo
        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;

    CARGA : process
    begin
        R <= (300.0**2)/150.0; --600.0; -- Vo=300V, P = 150 W
        Ir <= 0.0; -- P = Us*Ir = 0 W
        Resist <= true; -- Carga resistiva
        wait for 100 ns;
        wait; -- realiza un wait permanente
    end process CARGA;

    decaraciones;

end Test;

```

Figura 8. Fichero de banco de pruebas

4.- Plataforma Hardware y Entorno de Trabajo

Una vez obtenidas las descripciones VHDL sintetizables, los alumnos deben realizar la implementación de los circuitos y verificar su funcionamiento, completando el ciclo del proceso de diseño. La plataforma hardware elegida para las prácticas es la placa de desarrollo basada en la familia de FPGA's de Xilinx SPARTAN 3E [6]. Entre los recursos que incorpora la placa de desarrollo se puede destacar el dispositivo XC3S500E Spartan-3E FPGA, con hasta 232 pines de entrada-salida y más de 10,000 celdas lógicas. Contamos además con convertidores A/D y D/A, varias fuentes de reloj, entre ellas un oscilador en la placa que proporciona una frecuencia de 50MHz, interruptores, pulsadores, LED's, un display LCD de dos líneas y 16 caracteres, un puerto PS/2 y otro VGA, entre otros elementos [6]. Los recursos que incorpora la placa de desarrollo propuesta permiten cumplir todo el programa de prácticas minimizando el uso de elementos externos.

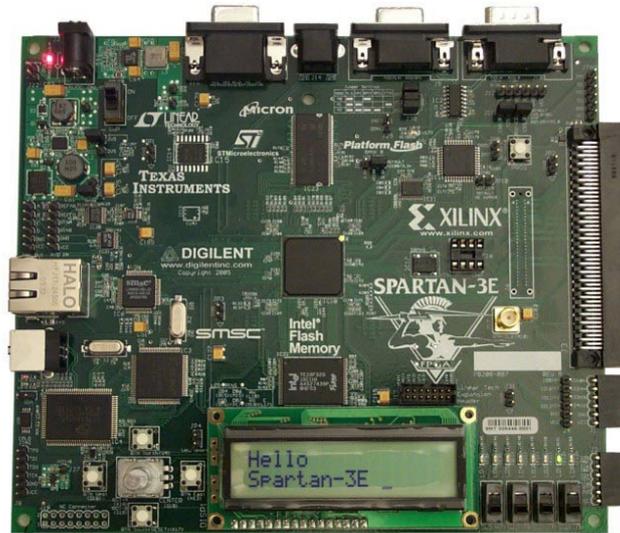


Figura 9. Placa de desarrollo basada en la familia de FPGA's de Xilinx SPARTAN 3E

La implementación de los diseños en la placa de desarrollo requiere del conocimiento por parte de los alumnos del entorno de trabajo ISE Foundation [7]. Este entorno de trabajo es desarrollado por Xilinx y en el mismo se encuentran disponibles todas las herramientas software necesarias para completar el flujo de diseño desde la depuración inicial de la descripción VHDL, simulación funcional, síntesis, simulación post-síntesis hasta la configuración final del dispositivo.

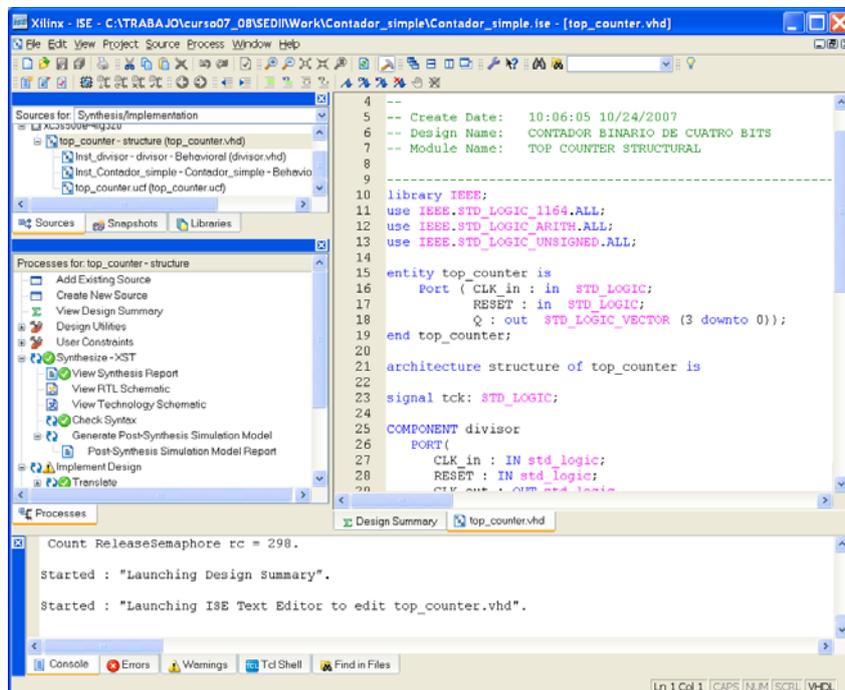


Figura 10. Ventana principal del Project Navigator del entorno de trabajo ISIE Foundation.

5.- Alumnado y metodología

Tal y como estaba previsto, la procedencia del alumnado es diversa. En la primera experiencia del curso 2007-08 participan tres alumnos en la versión en español de la asignatura, impartida en el primer cuatrimestre y cuatro en la versión en inglés, impartida en el segundo cuatrimestre. Se matriculan principalmente alumnos de intercambio como se muestra en la Fig. 11. Los alumnos son de segundo ciclo y su orientación académica es electrónica, electro-energética y mecatrónica. Ninguno de los alumnos tiene conocimiento previo en VHDL, si bien el alumno que procede de EE.UU. conoce el lenguaje Verilog. Esta circunstancia se aprovecha para que lidere la coordinación del trabajo práctico que deben realizar los alumnos.

Los objetivos académicos se plantean para adaptar la metodología docente al Espacio Europeo de Educación Superior (EEES) del que se tiene experiencia académica en el Centro (ETS de Ingenieros Industriales y de Telecomunicación de la Universidad de Cantabria) a través de planes piloto en planes de estudio diferentes al de Ingeniero Industrial. Estos objetivos se dividen en: Competencia técnica, competencia metodológica, competencia social y habilidades personales.

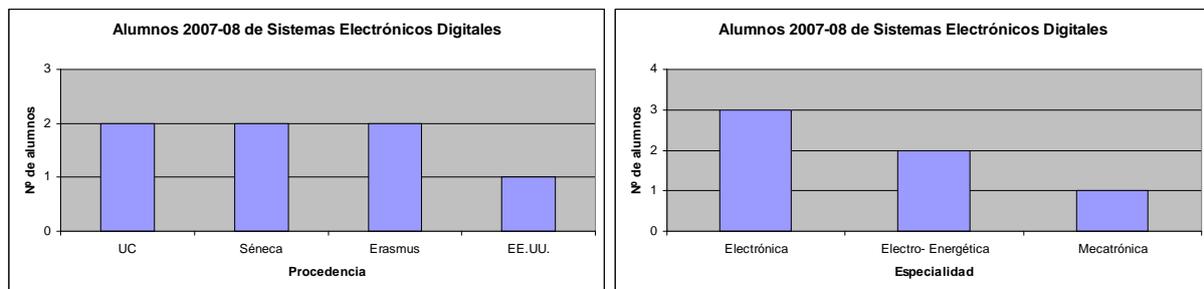


Figura 11. Procedencia y especialidad de los alumnos

5.1. Competencia técnica

El curso presenta una enseñanza práctica para que los estudiantes adquieran competencias en el modelado de sistemas en VHDL partiendo de una descripción matemática y en el diseño y realización de circuitos electrónicos digitales de complejidad media utilizando dispositivos programables, con énfasis en los siguientes aspectos:

- Conocimiento de un entorno de trabajo integrado por diferentes herramientas software, tanto con interfaz de texto como gráfico, complementado con una tarjeta electrónica sobre la que implementar y verificar los diseños.
- Determinación de las tareas fundamentales que el diseñador debe llevar a cabo para modelar sistemas, establecer especificaciones y conseguir diseños fiables.
- Conocimiento de la sintaxis de los principales elementos del lenguaje de descripción de hardware VHDL.
- Traducción de elementos de un sistema a descripción VHDL y análisis, síntesis y simulación de diseños combinatoriales y secuenciales simples utilizando VHDL.
- Práctica de los conceptos básicos de la electrónica digital utilizando VHDL.
- Selección de la lógica programable adecuada en función de las necesidades técnicas de la aplicación.
- Reconocimiento de la estrategia más adecuada para la descripción de máquinas de estado con VHDL.
- Adaptación de los diseños para su implementación en dispositivos lógicos programables.
- Análisis, síntesis y simulación de circuitos digitales síncronos.

- Definición de la estrategia de distribución de la señal de reloj cuando se desarrolla un circuito digital.
- Definición, planificación y realización de un test funcional de un circuito digital síncrono.

5.2. Competencia metodológica

En el curso se realiza una aproximación práctica al diseño de circuitos digitales descritos en VHDL y realizados en CPLDs o FPGAs. Tras algunas clases teóricas sobre el lenguaje VHDL y la secuencia de diseño, los estudiantes resolverán diferentes circuitos de baja complejidad que podrán constituir elementos constructivos de un circuito final más complejo. A través de las diferentes clases prácticas, los estudiantes describen, simulan, sintetizan y realizan circuitos digitales de interés industrial.

5.3. Competencia social

Se espera que los estudiantes participen en clase. Deben compartir las soluciones que proponen con el profesor y con el resto de estudiantes. Aprenderán la capacidad de reutilización de código y a enfrentarse a la solución de las especificaciones bajo condiciones iniciales y de contorno reales y considerando la interacción con otros circuitos. Muchas clases serán participativas basadas en el análisis de las especificaciones de los circuitos. Los estudiantes formarán uno o más pequeños grupos para tratar de resolver las especificaciones propuestas.

5.4. Habilidades personales

En el curso se presentan clases sobre el material básico junto con la utilización de herramientas de diseño para el modelado de sistemas, previamente descritos con una herramienta matemática y la descripción y síntesis de circuitos electrónicos digitales. Los estudiantes desarrollarán habilidades de diseño, de utilización de documentación práctica y de búsqueda de información útil en Internet. También organizarán los recursos de diseño en el laboratorio y aprenderán a verificar los circuitos que se diseñen mediante una herramienta hardware de desarrollo de prototipos.

5.5. Distribución del trabajo

En el curso tiene asignados 5 créditos ECTS y el esfuerzo de trabajo previsto para los estudiantes se distribuye, en horas, de la siguiente manera:

Horas de contacto con los profesores:	45
Preparación y seguimiento de las clases:	30
Generación de documentación:	20
Presentaciones:	20
Preparación del examen final:	10

Figura 12. Distribución de las horas de trabajo de los alumnos

6. Conclusiones

El artículo presenta un nuevo contenido y metodología de enseñanza de la asignatura Sistemas Electrónicos Digitales en el que se entrena a los alumnos en el modelado de sistemas en VHDL, traducción de modelos desarrollados en Matlab/Simulink y el diseño de circuitos digitales orientados al

control de estos sistemas, todo en un mismo entorno de trabajo de simulación y síntesis de circuitos digitales. Esto permite ofrecer la asignatura tanto como asignatura optativa del Plan de Estudios de Ingeniero Industrial, siendo parte de la Intensificación Automática y Electrónica Industrial, como para no especialistas y en el contexto de un nuevo programa “Diploma in Advanced Applied Technologies for Industry” destinado a alumnos, principalmente extranjeros, de procedencia diversa tanto geográfica como de especialidad. Los diseños de circuitos digitales sintetizables surgen así dentro del contexto de una aplicación industrial como alternativa o complemento al control digital por computador.

Referencias

- [1] <http://www.unican.es/Centros/etsiit/daati/>
- [2] A. de Castro, *Aplicación del control digital basado en hardware específico para convertidores de potencia conmutados*. Tesis Doctoral. Universidad Politécnica de Madrid. Madrid (2003).
- [3] A. Aounis, S. E. Cirstea, M. N. Cirstea. *Reusable VHDL architectures for induction motor PWM vector control, targeting*. 32nd Annual Conference on IEEE Industrial Electronics, IECON 2006. pp.4923 – 4928. París (2006)
- [4] M. N. Cirstea. *VHDL for industrial electronic systems integrated development*. IEEE International Symposium on Industrial Electronics, ISIE 2006. pp. 1516 – 1520. Montreal (2006)
- [5] H. Levy, F. Lessman. *Finite Difference Equations*. Dover Publications, Nueva York, Estados Unidos de América (1992)
- [6] Spartan TM-3E Starter Kit Board Referente Page: <http://www.xilinx.com/s3estarter>
- [7] Manual del ISIE 9.1 Foundation: http://www.xilinx.com/support/sw_manuals/xilinx9/index.htm