

# Hardware *copyleft* como Herramienta para la Enseñanza de Sistemas Embebidos

Carlos Iván Camargo – Universidad Nacional de Colombia  
cicamargoba@unal.edu.co

**Abstract**—El gran avance de las técnicas de fabricación de Circuitos Integrados ha permitido que los sistemas embebidos sean parte fundamental de nuestra vida, aún sin darnos cuenta diariamente interactuamos con decenas de ellos. Esto unido a la disponibilidad de herramientas software de desarrollo gratuitas (compiladores, simuladores, librerías, Sistemas Operativos, herramientas CAD) abre grandes posibilidades comerciales para países en vía de desarrollo ya que no son necesarias grandes inversiones de capital para la concepción, diseño, y fabricación de estos sistemas. Sin embargo, en la actualidad muy pocas universidades ofrecen cursos que permitan crear las habilidades necesarias para la realización de un producto comercializable, lo que se traduce en un abandono de la producción local y el aumento de la dependencia con la industria manufacturera asiática. Por otro lado, las herramientas utilizadas en la actualidad (tanto SW como HW) proporcionan un nivel de abstracción relativamente alto impidiendo que el estudiante entienda el funcionamiento global de un sistema digital, lo que le impide generar habilidades (específicamente las relacionadas con la concepción e implementación) necesarias para realizar el proceso completo. En este artículo se presenta una metodología para la enseñanza de diseño de sistemas embebidos utilizando herramientas hardware y software abiertas que ayuda a resolver los problemas mencionados anteriormente.

**Index Terms**—Educación en Ingeniería, Sistemas Embebidos, Hardware *copyleft*, Co-diseño HW/SW.

## I. INTRODUCCIÓN

El mercado de los sistemas embebidos continúa en aumento y su campo de acción se ha extendido en casi todas las actividades humanas. Según BBC, inc.<sup>1</sup> el mercado para el software embebido puede crecer de \$1.6 billones a \$3.5 billones en 2009, con una tasa de crecimiento anual (AAGR) del 16%, mientras la tasa de crecimiento para las tarjetas embebidas es del 10%; según *Venture Development Corporation (VDC)* más de un billón de dispositivos embebidos fueron vendidos en el 2004. De acuerdo con VDC el porcentaje de dispositivos basados en sistemas operativos comerciales tiende a disminuir cayendo del 43.1% en 2001 a 37.1% en 2004, esta tendencia se debe a la utilización de herramientas de libre distribución GNU/Linux [Stallman, 1999].

Cada vez con más frecuencia, se notan signos de la invasión digital, por ejemplo, en el aumento de chips embebidos en los dispositivos que utilizamos a diario, se ha demostrado [M. Weiser, 1993] que una persona que vive en un país industrializado interactúa con un promedio de 40 chips al día, de los cuales 5 son capaces de comunicarse en redes. Se estima que dentro de 10 años estaremos en contacto con cientos de

estos chips, la mayoría de los cuales acceden a densas redes de información [D. Servant, 2002], muchos de estos artefactos toman la apariencia de objetos que utilizamos en nuestra vida diaria (herramientas, vestuario, electrodomésticos, etc) pero son mejorados con sensores, actuadores, procesadores y software embebido. Esto unido a: el gran nivel de integración obtenido por la industria de los semiconductores; la disponibilidad de herramientas software de desarrollo gratuitas (compiladores, simuladores, librerías, sistemas operativos); la gran oferta de bienes y servicios relacionados con la manufactura; abre grandes posibilidades comerciales, ya que no es necesario contar con grandes sumas de dinero para la formación de una empresa de concepción, diseño, implementación y operación de sistemas digitales comercializables. Esta situación es ideal para la formación de empresas de base tecnológica y para la generación de empleo para técnicos y profesionales en electrónica, ciencias de la computación, diseñadores industriales, publicistas, administración, personal de mercadeo y ventas.

Sin embargo, en la actualidad estamos presenciando una tendencia global a delegar las tareas de manufactura de sistemas digitales a países asiáticos, donde la mano de obra calificada es abundante y barata; se presentan casos donde los creadores de una determinada tecnología no la desarrollan y dejan que estos países se beneficien de sus descubrimientos [A. Grove, 2010]. Esta situación se agrava a medida que las grandes empresas manufactureras asiáticas como Foxconn<sup>2</sup> capturan la producción de los grandes diseñadores como Apple, Nokia, DELL, HP y Microsoft, lo que genera el cierre de empresas manufactureras a lo largo del mundo, con la consecuencia de pérdida de empleo masivo. En la actualidad según *Bureau of Labor Statistics* y *Thomson Financial Extel Company Report* Foxconn emplea a más personas que Apple, Dell, Microsoft, HP, Intel y Sony combinados; esta situación es más grave en países en vía de desarrollo (como el nuestro), donde no existe la plataforma tecnológica para diseñar dispositivos digitales, y su papel se reduce a importadores de tecnología sin la capacidad de generar productos que satisfagan necesidades locales. Lo anterior lleva a preguntarse por la función y la situación de un profesional en áreas afines a la ingeniería electrónica en países donde no existe la capacidad de concepción y diseño.

Por otro lado, la tendencia moderna en los programas académicos a la utilización de herramientas de alto nivel para la enseñanza en áreas afines al desarrollo de dispositivos digitales [Jon Hall, 2009] ocasiona que los profesionales no

<sup>1</sup><http://www.bccresearch.com/>

<sup>2</sup><http://www.foxconn.com/>

adquieran las habilidades necesarias para completar la cadena concepción - diseño - implementación y operación, en la mayoría de los casos se generan habilidades para la concepción y el diseño a alto nivel y dejan los otros pasos en manos de herramientas especializadas y/o a empresas asiáticas. Esta situación resulta la más atractiva desde el punto de vista económico, ya que no es necesario adquirir maquinaria costosa ni contratar personal calificado para operarlas; sin embargo, limita la generación de empleo local a personas con un nivel de formación alto [A. Grove, 2010] generando desempleo en las personas con menor formación académica. Según John Hall presidente y CEO de Linux International “ algunas facultades preparan a la gente en el uso de productos en vez de tecnologías de nivel básico” [Jon Hall, 2009]. Esta situación unida al abandono tanto de la industria como en la academia de la implementación hace que la dependencia con las empresas manufactureras asiáticas aumente cada vez más.

Según el ex-director ejecutivo de Intel Andy Grove [A. Grove, 2010] la solución está en hacer de la creación de empleo la política económica gubernamental más importante y hacer que las demás giren en torno a ella. Además, es necesario volver a la producción interna con el fin de generar nuevos empleos, y volver a adoptar medidas que protejan la producción interna de los productos asiáticos. Sin embargo, para lograrlo es necesario crear en los profesionales las habilidades para implementar productos comercializables.

En este artículo presentamos un programa académico basado en la utilización de software y hardware libre para el área de electrónica digital que desarrolla las habilidades necesarias para Concebir, Diseñar Implementar y operar sistemas digitales. Adicionalmente, se relata la experiencia al aplicarlo en el departamento de ingeniería eléctrica y electrónica de la Universidad Nacional de Colombia.

### A. Flujo de diseño de sistemas embebidos

Los Sistemas Embebidos son sistemas heterogéneos que contienen componentes Software (microcontroladores, microprocesadores y DSPs) y Hardware (funciones implementadas en Dispositivos lógicos programables PLDs); por este motivo, es necesario adquirir habilidades en la utilización de lenguajes de programación como C o C++ para implementar las funciones software y Verilog o VHDL para la implementación de las tareas hardware; adicionalmente, deben conocer las diferentes formas de comunicación entre estos dos tipos de funciones. Aunque en el mercado existen herramientas que permiten la entrada de diseño utilizando lenguajes de alto nivel como *SystemC* o *SpecC* y proporcionan el código para implementar las tareas software, hardware y su interfaz de comunicación; no es recomendable utilizarlas en el ciclo de formación básico ya que impide que se conozca el flujo de diseño completo, suministrando un nivel de abstracción en el cual no es necesario conocer las herramientas utilizadas en la implementación<sup>3</sup>.

En la figura 1 se muestran los conceptos que deben dominar los diseñadores de sistemas embebidos, y las tareas que deben

realizarse para la concepción, diseño e implementación de un sistema embebido. En una gran parte de los programas académicos se estudian únicamente los temas relacionados con la concepción y diseño centrándose en las especificaciones funcionales del sistema, utilizando herramientas comerciales o COTS (Commercial off-the-shelf) para su implementación. Esta combinación genera dependencia e impide la generación de habilidades necesarias para implementar un sistema digital teniendo en cuenta restricciones económicas, físicas, eléctricas, ergonómicas, comerciales, etc. Nuestra propuesta se basa en la utilización de herramientas abiertas tanto hardware como software que permitan recorrer todo el proceso de concepción, diseño e implementación y de esta forma obtener un entendimiento integral del proceso sin generar dependencia a productos comerciales.

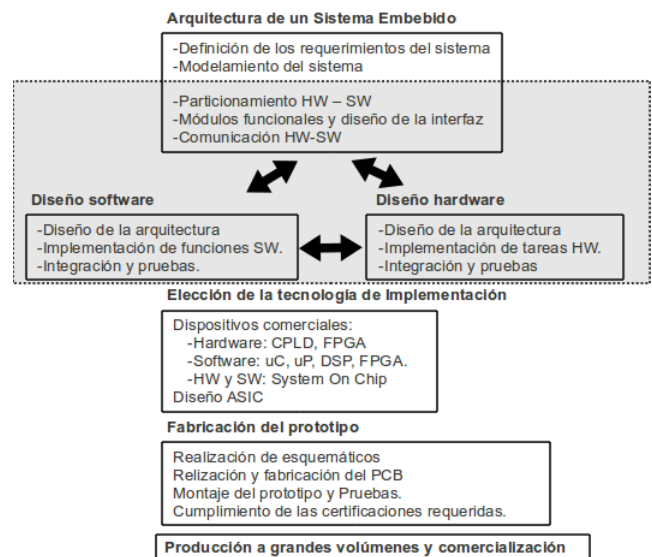


Fig. 1. Educación de sistemas embebidos. Tomada de:[H. Mitsui et al., 2009] y modificada

1) *Dominios de Diseño y Niveles de abstracción:* Existen tres dominios en los que se puede describir un sistema digital (ver Figura 2) *Funcional:* Describe el comportamiento funcional y temporal del sistema *Estructural:* Describe su composición a partir de bloques básicos y *Físico* relacionado con la estructura física del sistema a nivel de circuito integrado o placa de circuito impreso [Gajski D.D. et al., 2009]. Cada uno de estos dominios puede ser descrito utilizando diferentes niveles de abstracción; un nivel de abstracción alto permite el uso de lenguajes de alto nivel facilitando la entrada de diseño al extraer la funcionalidad de la parte física; por otro lado, los niveles de abstracción bajo utilizan bloques constructores elementales.

En el mercado existen herramientas que permiten entradas de diseño a nivel funcional utilizando especificaciones y algoritmos y de forma automática y optimizada generan representaciones en diferentes niveles de abstracción en los dominios estructural y físico. Es decir, a partir de las especificaciones y algoritmos que indican el funcionamiento del sistema pueden generar archivos para la fabricación de un circuito integrado o archivos de configuración o programación que pueden ser

<sup>3</sup>Vale la pena anotar que la utilización de *SystemC* o *SpecC* es muy importante en la etapa de modelamiento y validación de la funcionalidad

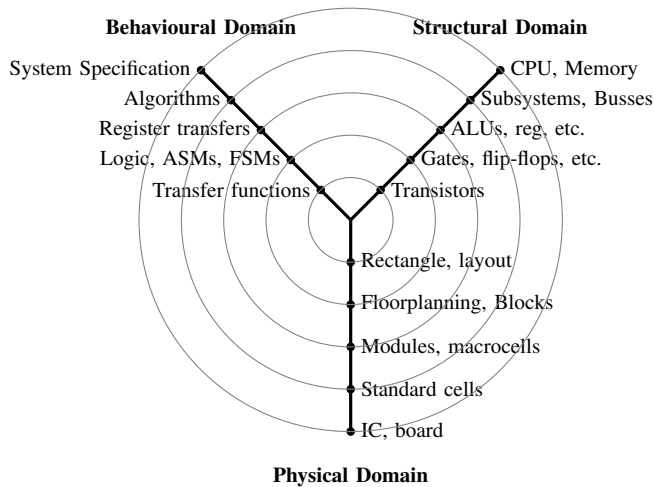


Fig. 2. Niveles de Abstracción Y de Gajski-Kuhn Fuente:[A. Gerstlauer et al., 2002]

utilizados en dispositivos comerciales. Desde el punto de vista comercial esto es muy útil ya que permite reducir el tiempo de diseño y los costos asociados. Sin embargo, presentan los inconvenientes mencionados anteriormente y por lo general son muy costosas.

## II. LA INICIATIVA CDIO

La iniciativa CDIO <sup>4</sup> ha sido desarrollada con ayuda de académicos, industriales, ingenieros y estudiantes[Worldwide CDIO Initiative], es un proyecto mundial que busca desarrollar una nueva visión de la educación en ingeniería y ha sido adoptada por un creciente número de instituciones académicas a lo largo del mundo. Esta iniciativa se basa en la suposición que los egresados de los centros de formación en ingeniería deben ser capaces de: **Concebir, Diseñar, Implementar y Operar** sistemas complejos en un entorno basado en equipos de trabajo que crean sistemas y productos. La educación de la ingeniería y las demandas del mundo real están tomando caminos separados, hacer parte de este esfuerzo mundial nos ayuda a mantener nuestros planes académicos actualizados con los cambios que se realizan en países más industrializados. Los principales objetivos de la iniciativa CDIO son: [Worldwide CDIO Initiative]:

- Educar a los estudiantes para dominar un conocimiento más profundo de los fundamentos técnicos.
- Educar a los ingenieros para liderar la creación y operación de nuevos productos y sistemas.
- Educar futuros investigadores para entender la importancia estratégica y el valor de su trabajo.

Las premisas que capturan la visión, objetivos y fundamentos pedagógicos de la iniciativa, concuerdan con la necesidad expuesta anteriormente de crear empleo y volver a la producción local de dispositivos digitales. Estas premisas son:

- Es posible cumplir las necesidades propias de la profesión mientras al mismo tiempo se realiza el proceso de Concebir, diseñar, implementar y operar sistemas en el contexto de los sistemas de Ingeniería.

- Los resultados de la formación deben ser fijados por los sectores interesados (Academia, Industria, Gobierno) y deben formar una secuencia de experiencias de aprendizaje, algunas de las cuales son experimentales, es decir, deben enfrentar a los estudiantes a situaciones que encontrarán en el ejercicio de su profesión.
- La adecuada construcción de esta cadena de actividades tendrán un doble impacto en la formación de los estudiantes, por un lado facilitará el aprendizaje de habilidades críticas e inter-personales y fortalecerá las habilidades de construcción de sistemas, productos y procesos, mientras se mejora el aprendizaje de los conceptos fundamentales.

### A. Estructura del Plan de Estudios CDIO

La figura 3 muestra los bloques constructores del plan de estudios CDIO, en el primer nivel podemos observar que todo individuo interesado en obtener habilidades técnicas posee *Habilidades Personales y Profesionales*, las cuales son fundamentales para la práctica. Con el fin de desarrollar sistemas de ingeniería complejos, los estudiantes deben dominar los fundamentos del *razonamiento y conocimiento técnico*. Para trabajar en un entorno moderno basado en grupos los estudiantes deben desarrollar *habilidades interpersonales* de comunicación y trabajo en equipo. Finalmente con el fin de ser capaz de crear y operar productos y sistemas un estudiante debe entender el concepto de *concebir, diseñar, implementar y operar Sistemas en el Contexto Social y Empresarial*[Edward F. Crawley, 2001]

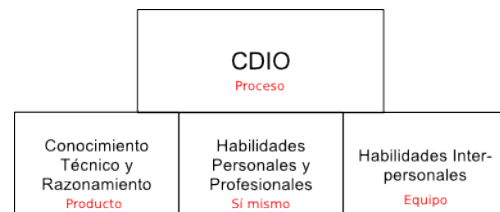


Fig. 3. Bloques constructores de conocimiento, habilidades y actitudes necesarias para concebir, diseñar, implementar y operar sistemas en el contexto social y empresarial fuente:[Edward F. Crawley, 2001]

1) *Nivel 1: Razonamiento y Conocimiento Técnico*: Los componentes del primer nivel *razonamiento y conocimiento técnico* son comunes a los planes de estudio de las ingenierías modernas y son:

- Fundamentos Avanzados de Ingeniería.
- Fundamentos del núcleo de Ingeniería.
- Conocimiento científico.

La razón de colocar este bloque constructor en el primer nivel es solo para recordar que el objetivo primordial de cualquier programa de pregrado es el desarrollo de un profundo conocimiento de fundamentos técnicos. En este trabajo no se cambiará este componente ya que para hacerlo es necesario un consenso con las demás carreras de la facultad de Ingeniería, labor que puede tomar varios años.

2) *Habilidades Personales, Profesionales e Interpersonales*: Los niveles 2 y 3 se centran en las habilidades personales

<sup>4</sup><http://www.cdio.org>

que debe poseer un individuo para que pueda cumplir con el objetivo de la Iniciativa CDIO. El nivel 2 esta compuesto por:

- Las habilidades profesionales que representan las tres formas de pensar más practicadas por los ingenieros: Resolución de problemas, descubrimiento de conocimiento y pensamiento sistémico.
- Actitudes que incluyen integridad y comportamiento profesional así como las necesarias para planear la profesión.

Las habilidades que no hacen parte del contexto profesional ni del inter-personal son llamadas *habilidades y actitudes personales*, incluyen el carácter, iniciativa, perseverancia, formas de pensar más genéricas como pensamiento crítico, creativo y habilidades propias como curiosidad, aprendizaje continuo y manejo del tiempo.

Las habilidades inter-personales, son un subconjunto de las habilidades personales y se dividen en dos grupos que se traslapan llamados: Equipo de trabajo y comunicaciones. El grupo de trabajo se refiere a las habilidades necesarias para formar, operar, fortalecer y liderar un equipo con habilidades específicas de un equipo de trabajo técnico. La comunicación se compone de habilidades para idear estrategias de comunicación y aquellas para utilizar los medios orales, escritos, electrónicos y gráficos y en el caso colombiano el uso del idioma Inglés.

3) *Habilidades CDIO*: Habilidades necesarias para Concebir, Diseñar, Implementar y Operar Systems en el Contexto Social y empresarial. Estos cuatro componentes son necesarios para que los egresados de las carreras de ingeniería eléctrica y electrónica sean capaces de absorber los conocimientos que las nuevas tecnologías proporcionan, adaptarlos a la situación tecnológica y al contexto social del país para generar productos que resuelvan necesidades locales. Para satisfacer una necesidad de la sociedad es necesario conocer la dinámica empresarial, los principios que la rigen y como se debe actuar en una empresa de cualquier tipo y tamaño.

### B. Implementación del Plan de Estudios CDIO

La Figura 4 muestra los componentes que deben ser especificados para implementar el plan de estudios CDIO al currículo de las asignaturas del área de electrónica digital. En primer lugar se encuentran los resultados esperados del proceso de aprendizaje, esto es, Qué deben saber y que deben ser capaces de hacer los estudiantes al final del curso? Para contestar a esta pregunta es necesario definir las **habilidades** que serán reforzadas o desarrolladas y los *objetivos* de cada asignatura.

Para alcanzar los objetivos definidos en el primer paso, es necesario generar una serie de **actividades** que le permitan al estudiante: retener nuevos conocimientos y habilidades y desarrollar las competencias deseadas, el número de actividades debe ser tal que cubran todas las habilidades que se quieran desarrollar o reforzar.

Finalmente, se deben desarrollar métodos de evaluación que permitan conocer el nivel de competencia de los estudiantes, y de esta forma ajustar las actividades para obtener los resultados esperados.

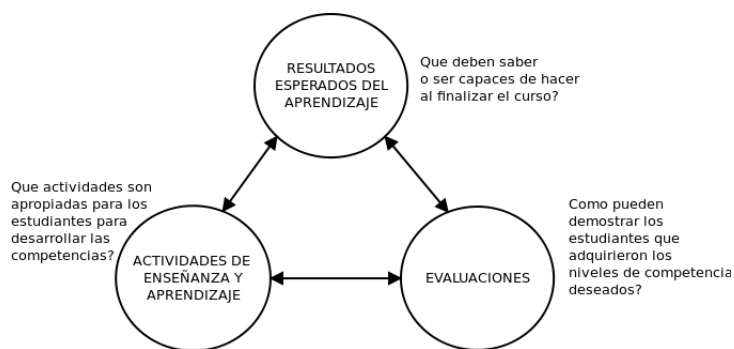


Fig. 4. Objetivos, Actividades, y Evaluación:

Competencias de las Habilidades CDIO Nivel 2 y 3			
APTITUDES PERSONALES Y PROFESIONALES	Nivel 1		
	E. Digital I	E. Digital I	Sist. Emb.
<i>Planteamiento y Resolución de problemas de Ingeniería</i>		EU	
1 Identificación y Formulación del problema		EU	
2 Modelamiento		EU	
3 Solución y recomendación		EU	
<i>Experimentación y Descubrimiento de Conocimiento</i>		U	
4 Formulación de hipótesis		U	
5 Investigación experimental		U	
<i>Pensamiento Sistémico</i>		EU	
6 Pensamiento Global		U	
7 Surgimiento e interacciones		U	
<i>Habilidades y actitudes personales</i>		U	
8 Pensamiento creativo		IEU	
9 Pensamiento crítico		IEU	
10 Toma de conciencia de conocimientos propios		IEU	
11 Curiosidad y aprendizaje permanente <i>Habilidades y actitudes profesionales</i>		U	
12 Ética profesional, integridad, responsabilidad		U	
13 Comportamiento profesional		U	
39 Confianza y Lealtad		IEU	
HABILIDADES INTERPERSONALES	Nivel 1		
	E. Digital I	E. Digital I	Sist. Emb.
<i>Equipo de Trabajo</i>		EU	
14 Formar grupos efectivos	EU	U	U
15 Equipo de Liderazgo	EU	U	U
40 Equipo Técnico y Multi-disciplinario	EU	U	U
<i>Comunicaciones estructuradas</i>		EU	
16 Estrategia de comunicación	EU	U	U
17 Estructura de la comunicación	EU	U	U
18 Comunicación Escrita	EU	U	U
19 Comunicación Electrónica	EU	U	U
20 Presentación Oral	EU	U	U
<i>Comunicación en Idioma Extranjero</i>		U	
21 Inglés		U	
<i>Comunicaciones Informales: Relacionarse con los demás</i>		U	
41 Preguntar, Escuchar y Dialogar	EU	U	U
42 Negociación, compromiso y resolución de conflictos	EU	U	U
43 Establecimiento de conexiones	IEU	U	U

TABLE I  
COMPETENCIAS PARA LOS NIVELES 2 Y 3 CDIO

1) *Introducir, Enseñar y Usar*: Para trasladar esta lista de habilidades a objetivos de aprendizaje es necesario determinar el grado de competencia que se espera que el profesional adquiera en cada una de las asignaturas. Por supuesto, algunas de estas habilidades no pueden obtenerse en una asignatura y deben ser el resultado de un esfuerzo multi-disciplinar donde todo el plan académico contribuye a generarla, lo que requiere un consenso del personal académico. En el Departamento de Ingeniería Eléctrica y Electrónica de la Universidad Nacional de Colombia se está realizando esta tarea y los resultados presentados en este estudio hacen parte de esta iniciativa. Los niveles de competencia seleccionados para indicar el nivel en que debe ser apropiada una determinada habilidad son:

- 1) Introducir: Introduce pero no evalúa.
- 2) Enseñar: Enseña y evalúa.
- 3) Utilizar: Utiliza, puede ser evaluado o no.

2) *Competencias de las Habilidades CDIO 2 y 3*: La tabla I muestra las competencias IEU para las *Aptitudes Personales* y *Profesionales* de las tres asignaturas del área de electrónica digital.

Las anteriores habilidades son necesarias para que el profesional sea capaz de desempeñarse en el entorno laboral e interactuar con otros profesionales de la misma o de diferente área. El proyecto de hardware copleft juega un papel muy importante en las habilidades relacionadas con el trabajo en equipo y las comunicaciones estructuradas ya que al ser parte de una comunidad que utiliza el medio escrito como forma de comunicación (debido a que sus miembros se encuentran en diferentes países), personas en formación pueden aprender de profesionales expertos en varias áreas comunicándose de forma directa utilizando el gran número de listas de discusión disponibles en los proyectos abiertos.

### C. Competencias de las Habilidades C.D.I.O. Sistemas en el contexto Empresarial, Social y Ambiental - Innovación

La tabla II muestra las competencias IEU para las C.D.I.O. *Sistemas en el contexto Empresarial, Social y Ambiental - Innovación* de las tres asignaturas del área de electrónica digital. Este componente es de vital importancia para generar en el estudiante las habilidades necesarias para la creación y comercialización de nuevos productos, y es el objetivo principal del presente método.

HABILIDADES CDIO	Nivel I		
	E. Digital I	E. Digital II	Sist. Emb.
<b>Contexto Externo, Social, Económico y Ambiental</b>	IEU		
22 Rol y responsabilidad de los Ingenieros	IEU		
23 Impacto sobre la sociedad y el medio ambiente	IEU		
24 Cuestiones y valores actuales	IEU		
44 Sostenibilidad y necesidad de un desarrollo sostenible	IE	IE	IE
<b>Empresa y contexto empresarial</b>	EU		
25 Interesados en la empresa, metas y objetivos	I		
26 Espíritu Empresarial Técnico	I		
27 Trabajo exitoso en organizaciones	I		
45 Finanzas y Economía de los Proyectos de Ingeniería	IE	IE	IE
<b>Concepción y Administración de Sistemas en Ingeniería.</b>	IEU		
28 Entender las necesidades y establecer las metas	IEU	EU	U
29 Definir la función, concepto y arquitectura	IEU	EU	U
<b>Diseño</b>	IEU		
30 Proceso de Diseño	IEU	EU	U
31 Fases del proceso de Diseño y enfoques	IEU	EU	U
32 Utilización de conocimiento científico en el diseño	IEU	EU	U
33 Diseño específico	IEU	EU	U
34 Diseño multi-disciplinario	I	E	U
<b>Implementación</b>	EU		
35 Proceso de fabricación Hardware	IE	EU	U
36 Proceso de Implementación de Software	I	EU	U
37 Integración Software - Hardware	I	EU	U
38 Pruebas, verificación, validación y certificación	IE	EU	U

TABLE II

COMPETENCIAS DE LAS HABILIDADES C.D.I.O. SISTEMAS EN EL CONTEXTO EMPRESARIAL, SOCIAL Y AMBIENTAL - INNOVACIÓN

### D. Actividades

A continuación se enumeran las actividades propuestas para que el estudiante adquiera o mejore las competencias y habilidades de las tablas I y II (a cada competencia o habilidad se le asignó un número para poder referenciarla); en cada actividad aparece una lista de números que indica a que competencias está dirigida. Por ejemplo, la *lectura de material del curso* está relacionada con las habilidades 10 (toma de conciencia de conocimientos propios) y 11 (curiosidad y aprendizaje permanente)

1) *Lectura de material del curso 10, 11*: Con la lectura previa de los temas, el estudiante adquiere la capacidad de absorber conocimiento (11), identificar sus preferencias, deficiencias y buscar ayuda para suplirlas (10), lo cual ayuda al mejoramiento de las habilidades para el auto-aprendizaje, uno de los problemas detectados en los estudiantes es la

necesidad de una autoridad que le proporcione la información que necesita para resolver un problema o tomar una decisión.

2) *Lectura de material Técnico en Inglés 10, 11, 6, 30, 33, 21*: La mayor parte de la documentación de los componentes electrónicos esta escrita en inglés técnico, es necesario que el estudiante aprenda a entender este tipo de escritura y se familiarice con su estructura. Esto le permite identificar el funcionamiento de un componente del sistema (6,30), determinar que componente se adapta mejor a sus necesidades (33) y mejorar sus habilidades para comunicarse en inglés (21).

3) *Utilización de Metodologías de Diseño 1, 2, 3, 6, 7, 9, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38*: La metodología de diseño(30,31) de sistemas embebidos requiere identificar un problema(1, 28), plantear una solución(3,29,32) lógica de alto nivel (9), modelarla (2) a nivel de sistema(6), verificar el cumplimiento de los requerimientos(33,38); adicionalmente, proporciona métodos para determinar su arquitectura óptima y definir la función e interacción(37,7) de sus componentes software (36) y hardware (35).

4) *Implementación de Sistemas Digitales Sencillos 3, 14, 29, 30, 35, 36, 17, 18, 19*: La realización de prácticas de laboratorio en las que grupos de trabajo (14) implementan diseños de baja o media complejidad le permite al estudiante: formular recomendaciones (3) para que no se repitan errores en experiencias futuras, y utilizar sistemas de desarrollo (30) para la implementación de tareas HW y SW a bajo nivel (36). Con el fin de mejorar la capacidad de comunicación escrita (18, 19) se deben presentar informes que refuercen las habilidades generadas en la utilización de la metodología de diseño, para esto se recomienda que se incluya(17):

- Un diagrama de caja negra que indique las entradas y salidas del sistema.
- Una descripción de alto nivel de los algoritmos que implementan la solución (29).
- Un diagrama de bloques que indique el particionamiento y la interconexión entre sus componentes (30).
- Descripciones de alto nivel de cada uno de los componentes (31).
- La implementación y simulación de cada componente y del sistema completo (35), donde se muestre que el sistema cumple con las especificaciones funcionales(38)

5) *Proyecto Final 1,2,3, 14, 15, 30, 31, 32, 33, 34, 35, 22, 23, 24, 25, 26, 27*: Con el fin de fomentar el espíritu empresarial técnico (26) durante el semestre se trabajará para definir las especificaciones(1,2,3), diseñar(30,31,32,33,34) e implementar un dispositivo que resuelva una hipotética necesidad de la sociedad (22) (con la complejidad adecuada para cada curso), en la sesión teórica se tratarán aspectos relacionados con la concepción, diseño, Identificación y definición de las funciones de los componentes del sistema, mientras que en el componente práctico, los relacionados con la implementación de dichos componentes sobre PLDs o SoC, esto se hace para que el estudiante vea los Lenguajes de Descripción de Hardware (HDL) como herramientas que le ayudan en el proceso de diseño y no el objetivo del curso.

A los estudiantes se les hace una descripción funcional de alto nivel del sistema, ellos deben organizarse en grupos de trabajo (14,15), definir la función de cada integrante

(27,14,31), establecer estrategias de comunicación (16,31), realizar y/o cumplir un cronograma de actividades (25,31) que permitan resolver la necesidad en el tiempo especificado (22). Una de las estrategias de comunicación es la realización de presentaciones orales (20), en las que cada equipo de trabajo expondrá el estado de su sub-proyecto, indicando las razones que se tuvieron en cuenta en cada decisión y como se resolvieron los problemas encontrados (24). Adicionalmente todo este proceso debe documentarse en el sitio web del curso (wiki) con el objetivo de crear una base de proyectos que permitan a futuros estudiantes utilizar la experiencia obtenida (23) y en un determinado caso dar continuidad al proyecto.

El estudiante debe diseñar y construir placas de circuito impreso con los circuitos necesarios para su aplicación (35) siguiendo las normas de diseño establecidas por el fabricante (resolución, número de capas, costo) y las restricciones del circuito (Capacidad de corriente, niveles de ruido, compatibilidad electromagnética, etc).

Vale la pena aclarar que durante el primer curso los estudiantes no poseen la experiencia necesaria para realizar (sin asistencia) labores como la división de tareas, generación de un cronograma de actividades y fijar la estrategia de comunicación, razón por la cual el docente debe acompañar este proceso.

6) *Participación en listas de discusión 21*: Con el objeto de aumentar las capacidades en la comunicación en idioma extranjero, se alentará a los estudiantes a que hagan parte de listas de discusión en diferentes temas técnicos, algunos problemas que encontrarán en la realización de las diferentes prácticas deben ser consultados en estas listas para encontrar una forma de solución

7) *Análisis económico 45*: En la última etapa del proyecto final, el estudiante debe hacer el cálculo de los costos asociados al proceso de fabricación del prototipo teniendo en cuenta el costo del recurso humano y de los componentes utilizados. Adicionalmente, deben buscar empresas nacionales o extranjeras que suministren el servicio de fabricación de placas de circuito impreso, montaje de componentes con el fin de realizar cotizaciones para la fabricación de 100 unidades del dispositivo. Con esto se busca que el estudiante entienda las diferencias entre prototipado y producción y analice que factores son determinantes en el costo asociado en el proyecto y busque alternativas para reducirlos.

### III. MÉTODO PARA LA ENSEÑANZA DE SISTEMAS EMBEBIDOS

Basándose en la metodología de diseño para sistemas embebidos [Luis Alejandro Cortés, 2005] y en los dominios de diseño y niveles de abstracción de Gajski-Kuhn, se realizó una división de temas que busca crear habilidades de forma gradual e incremental. En la Figura 7 podemos observar esta división y las herramientas que se utilizarán en cada curso, como herramienta de desarrollo hardware se utilizará una plataforma que proporcione los archivos y documentos necesarios para replicarla, modificarla y pueda ser utilizada como base de desarrollos comerciales.

#### A. SIE: Plataforma abierta para el desarrollo de sistemas embebidos

En el mercado existe una gran variedad de plataformas que pueden ser utilizadas para el estudio de sistemas embebidos, sin embargo, no todas son adecuadas para la implementación del método que proponemos ya que se requiere: acceso a los esquemáticos y a los archivos de fabricación del PCB con posibilidad de modificación; acceso a la documentación completa del proceso de fabricación; acceso a la cadena de producción; utilización de herramientas abiertas para su programación; un PLD para la implementación de tareas HW; un procesador para la implementación de tareas SW; un canal de comunicación entre el procesador y el PLD; y una comunidad que desarrolle aplicaciones para dicha plataforma y que proporcione medios para el intercambio de información a través de listas de correo y wikis.

Después de una búsqueda minuciosa no se encontraron plataformas que cumplieran con estas condiciones, en especial con las relacionadas con el proceso de diseño y de producción, esto es normal, ya que la mayoría de las empresas no quieren que se fabriquen sus plataformas y los proyectos individuales no poseen la infraestructura necesaria para la producción masiva. Por este motivo, se decidió crear una plataforma que cumpliera con los requerimientos (plataforma SIE), para ello se buscaron proyectos similares que permitieran su creación y que el producto creado sea una extensión de dicho proyecto. El proyecto Qi-Hardware [Qi Hardware] busca definir el concepto *copyleft hardware* basándose en el movimiento de software libre y código abierto (FOSS [R. Stallman, 2007]) y proporciona un enlace con la industria manufacturera asiática, adicionalmente; su producto *BEN NANONOTE* es una plataforma que permite la reproducción y modificación (pero fué diseñada para desarrolladores software) junto a SIE son referencia para definir los requisitos que debe cumplir un dispositivo *hardware copyleft*

1) *Hardware copyleft*: El proyecto SIE<sup>5</sup> [W. Spraul et al.] fué creado para satisfacer las necesidades de los desarrolladores de hardware permitiendo la creación de aplicaciones comerciales bajo la licencia Creative Commons BY - SA [Creative Commons, 2004] la que permite la distribución y modificación del diseño (incluso para aplicaciones comerciales), con el único requisito de que los productos derivados deben tener la misma licencia y deben dar crédito al autor del trabajo original. Lo que constituye la base de los productos *hardware copyleft*.

Al ser inspirado en el movimiento FOSS, los dispositivos *hardware copyleft* comparten la misma filosofía [R. Stallman, 2007], y son el complemento perfecto del software libre. Para que un dispositivo HW sea reproducible y modificable es necesario: suministrar los archivos necesarios para la fabricación, es decir, los esquemáticos y los archivos de la placa de circuito impreso; la cadena de herramientas de compilación y depuración para desarrollo de aplicaciones; el código fuente de: el programa que inicializa la plataforma (*bootloader*), la herramienta que carga dicho programa en la memoria no

<sup>5</sup>SIE es la diosa del agua para la cultura muisca, antiguos habitantes del centro de Colombia



volátil (*usbboot*), el sistema de archivos y aplicaciones (*open-wrt*); documentación completa que indique como fué diseñada, como fué construida, como utilizarla, como desarrollar aplicaciones y tutoriales que expliquen el funcionamiento de los diferentes componentes.<sup>6</sup> Adicionalmente, se debe contar con la posibilidad de fabricación y montaje, lo que constituye la principal diferencia entre el software y el hardware libre. Esto contrasta fuertemente con el movimiento de software libre, en donde no se requiere inversión de capital para modificar un proyecto existente. Por esta razón, pueden existir varios niveles de libertad, un proyecto que utilice componentes costosos y de difícil consecución limitará su alcance a un sector determinado.

2) *Arquitectura*: La Figura 5 muestra el diagrama de bloques de la plataforma SIE, en ella podemos encontrar un procesador que posee periféricos para controlar la comunicación serial (UART), memorias micro-SD, un puerto I2C, un LCD a color de 3 pulgadas, 2 entradas y salidas de audio stereo, 2 entradas análogas; una FPGA que proporciona 25 señales de entrada/salida digitales de propósito general (GPIOs) y controla un conversor análogo digital de 8 canales. Existen dos canales de comunicación entre la FPGA y el procesador: uno para controlar el puerto JTAG, lo que permite la configuración de la FPGA desde el procesador (lo que elimina la necesidad de cables de programación); y otro que proporciona el bus de datos, dirección y control para comunicarse con las tareas HW o periféricos implementadas en la FPGA. El procesador utilizado es un Ingenic JZ4725 (MIPS) corriendo a 400MHz, se dispone de una memoria NAND de 2GB para almacenamiento de datos y programas, así como de una memoria SDRAM de 32 MB, lo que permite la ejecución de una gran variedad de aplicaciones Linux.

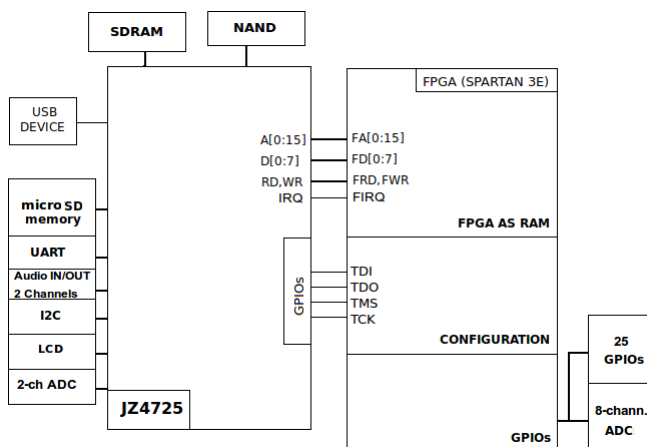


Fig. 5. Estructura de la plataforma de desarrollo SIE

3) *Comunicaciones*: SIE proporciona un canal de comunicación y alimentación a través del puerto USB-device, y es configurado para ser utilizado como una interfaz de red (*usb0*), permitiendo la transferencia de archivos y ejecución de una consola remota utilizando el protocolo ssh; este canal de comunicación también se utiliza para programar la memoria

NAND no volátil, por lo que para realizar la programación completa de los componentes de la plataforma solo es necesario un cable USB.

4) *Especificaciones físicas*: Las dimensiones de SIE son 8cm de largo, 8 cm de ancho, 1cm de altura; su placa de circuito impreso es de dos capas, utiliza líneas de 0.2m, vías de 0.3mm de diámetro, los componentes se encuentran en una sola capa y son TQFP o SMD, no posee componentes BGA o QFN lo que facilita el montaje manual. Todo esto hace que sea posible la reproducción y modificación de esta plataforma a un precio muy bajo. El costo de fabricación de esta tarjeta se estima en 70 usd (incluyendo el LCD) para 50 unidades. La figura 6 muestra la apariencia de la plataforma de desarrollo SIE.



Fig. 6. Plataforma de desarrollo SIE

### B. Curso básico

Para el curso básico se trabajará la mayor parte de los niveles de abstracción del dominio funcional; partiendo de unas especificaciones funcionales se generará un modelo del sistema utilizando algoritmos que describan el comportamiento de las diferentes tareas que implementan el sistema (bloques funcionales). Estos bloques serán implementados, en dispositivos lógicos programables como FPGAs o CPLDs. A partir de estos algoritmos se identifican las operaciones básicas aritméticas, lógicas que modifican los datos asociados a cada función, para generar el camino de datos; el camino de datos proporciona señales que controlan el instante en el que se ejecuta la operación; dichas señales deben ser generadas por un módulo (como máquina de control) diseñado para implementar un determinado algoritmo; estos dos módulos se implementarán con bloques lógicos básicos y máquinas de estados finitos utilizando lenguaje de descripción de hardware (VHDL, verilog). Estas descripciones son la entrada a herramientas que realizarán la transición al dominio estructural generando las compuertas lógicas, flip-Flops y las interconexiones que implementen la funcionalidad requerida. Durante el proceso es necesario realizar simulaciones que permitan comprobar el cumplimiento de las especificaciones iniciales, si no se cumple alguna de ellas se debe volver a repetir el proceso.

<sup>6</sup>Todo esto puede ser descargado del sitio del proyecto: <http://projects.qi-hardware.com/index.php/p/nn-usb-fpga/>

Durante el desarrollo del primer curso se estudiarán los conceptos básicos de los sistemas digitales como sistemas numéricos, operaciones aritméticas y lógicas, lógica combinatoria y secuencial. Se utilizarán lenguajes de descripción de hardware como VHDL y verilog como entrada de diseño a herramientas que realizan la síntesis digital. Para evitar crear dependencia, se enseñará la forma adecuada de implementar código re-utilizable que no utilice componentes específicos de un determinado fabricante.

1) *Diseño de aplicaciones utilizando HDL y PLDs*: Para generar las habilidades necesarias para concebir, diseñar, implementar y operar sistemas digitales, se realizarán prácticas sencillas que ayuden al estudiante a entender los mecanismos de implementación de tareas HW en un dispositivo lógico programable utilizando la plataforma de desarrollo SIE; como puede verse en la figura 5 la FPGA solo controla un conversor análogo digital serial de 8 canales, y proporciona 25 GPIOs, esto se hizo de forma intencional para que los estudiantes se vean forzados a realizar las conexiones eléctricas de los dispositivos externos a sus aplicaciones; las plataformas comerciales proporcionan una gran variedad de dispositivos conectados a los PLDs, lo que no es muy recomendable ya que el estudiante no aprende a leer la hoja de especificaciones del fabricante de un determinado dispositivo para determinar su forma de conexión, y/o las condiciones que se deben tener en cuenta para su correcto funcionamiento; afectando la generación de habilidades necesarias para la elección de componentes, realización y lectura de esquemáticos y diseño de layouts.

El procesador de la plataforma SIE será utilizado como camino de configuración de la FPGA; el archivo de configuración será descargado al sistema de archivos de la plataforma SIE utilizando el protocolo *ssh* y la interfaz de red USB; un programa en espacio de usuario se encarga de configurar la FPGA con el archivo deseado. Adicionalmente, existe una aplicación basada en el proyecto *urjtag*<sup>7</sup> ejecutándose en el procesador que permite la generación de vectores de prueba y recolección de resultados utilizando el puerto JTAG [Texas Instruments, 1996], lo que permite que el estudiante pueda probar su circuito sin instrumentos de medición adicionales.

### C. Arquitectura de Computadores

En este curso se trabajará en el dominio estructural comenzando desde los componentes básicos de una CPU hasta llegar a la arquitectura de un sistema sobre silicio (SoC), esto con el fin de conocer y entender la arquitectura y funcionamiento de los dispositivos en los que se ejecutan las tareas software. Se utilizarán periféricos conectados a través de buses a la CPU para implementar tareas hardware. Se realizará la implementación de un Sistema sobre silicio (SoC) y se trabajará con herramientas de libre distribución (cadena de herramientas GNU [Stallman, 1999]) para programar aplicaciones que involucren el uso de tareas HW y SW.

1) *Arquitectura de una CPU y tareas SW*: Utilizando procesadores *softcore* se estudiarán los componentes básicos de la CPU, el camino de datos: banco de registros, bloques

aritméticos y lógicos y buses internos, se analizarán las diferentes instrucciones que proporciona la arquitectura bajo estudio y se analizará el funcionamiento de la máquina de control, identificando los componentes que permiten el almacenamiento y ejecución secuencial de instrucciones definidas por el usuario. En este punto se introducirán los conceptos de llamado a funciones, atención de interrupciones, direccionamiento directo e indirecto y acceso a memoria externa.

Para este estudio, se utilizarán procesadores implementados en lenguajes de descripción de hardware que cuenten con herramientas de programación de bajo (assembler) y alto nivel (C, C++), con el fin de realizar simulaciones, aplicaciones y modificaciones. Al finalizar el curso se pretende que el estudiante entienda las diferencias entre tareas software y tareas hardware y podrá realizar experimentos que le permitan comparar las características de ambas implementaciones. En la actualidad se está trabajando con los SoC *plasma* basado en un procesador MIPS<sup>8</sup>, *MICO 32* de lattice<sup>9</sup> y *openrisc*<sup>10</sup> de OpenCores, implementados en VHDL o Verilog.

2) *Periféricos, tareas HW y comunicación HW - SW*: Se utilizarán los periféricos para la implementación de tareas HW y se estudiarán las diferentes formas que existen para comunicarse con las tareas SW (buses, interrupciones, polling) presentando los criterios de selección para la implementación de tareas SW y HW.

3) *Memorias*: Se introducirá el concepto de mapa de memoria, decodificador de direcciones, memorias volátiles (ejecución de programas y de uso general) y memorias no volátiles (almacenamiento de programas). Se introducirá el concepto de *bootloader* y su uso en la carga de aplicaciones a las memorias volátiles internas y externas del SoC. Y finalmente se mostrarán los diferentes métodos existentes para programar las memorias no volátiles.

4) *Programación*: Utilizando la cadena de herramientas GNU, se realizará el flujo de desarrollo para tareas SW desde la entrada de diseño utilizando el lenguaje C, hasta la generación del archivo binario que contiene las instrucciones (para la CPU en estudio) que implementan dicha tarea. Utilizando herramientas propias se generarán los archivos para inicializar la memoria de programa (implementada en la FPGA) con estas instrucciones. Se introducirá el concepto de *bootloader* y su uso en la carga de aplicaciones a la memoria interna del SoC.

5) *Diseño de aplicaciones que involucren co-diseño HW-SW*: Tanto la CPU, los periféricos (tareas HW), la memoria ram y la memoria de programa estarán implementados en la FPGA de SIE, por lo que es posible utilizar las mismas herramientas que se utilizaron en el curso anterior. Durante todo el periodo académico se desarrollará un proyecto de complejidad media que involucre el uso de tareas HW y SW, (en la página de la plataforma<sup>11</sup> se pueden observar los proyectos realizados hasta el momento); para esto se formarán equipos de trabajo de 3 personas que deben hacer una propuesta inicial (concepción y especificaciones), realizar la descripción funcional del sistema utilizando algoritmos (diseño y modelo del

<sup>8</sup><http://opencores.org/project,plasma>

<sup>9</sup><http://www.latticesemi.com/products/intellectualproperty/ipcores/mico32/index.cfm>

<sup>10</sup>[http://opencores.org/project,or1200\\_soc](http://opencores.org/project,or1200_soc)

<sup>11</sup><http://en.qi-hardware.com/wiki/2010-II/es>

<sup>7</sup><http://urjtag.org/>



sistema), realizar el particionamiento en funciones HW y SW, implementar estas funciones y diseñar una placa de circuito impreso con lo necesario para implementar la funcionalidad requerida (implementación). Se realizarán entregas periódicas para verificar el cumplimiento del cronograma propuesto por el equipo de trabajo y el proceso de diseño debe ser documentado en la página wiki del curso. Esta actividad generará habilidades de trabajo en equipo, elaboración de esquemáticos, fabricación de PCBs, escritura de documentos técnicos e implementación de sistemas digitales.

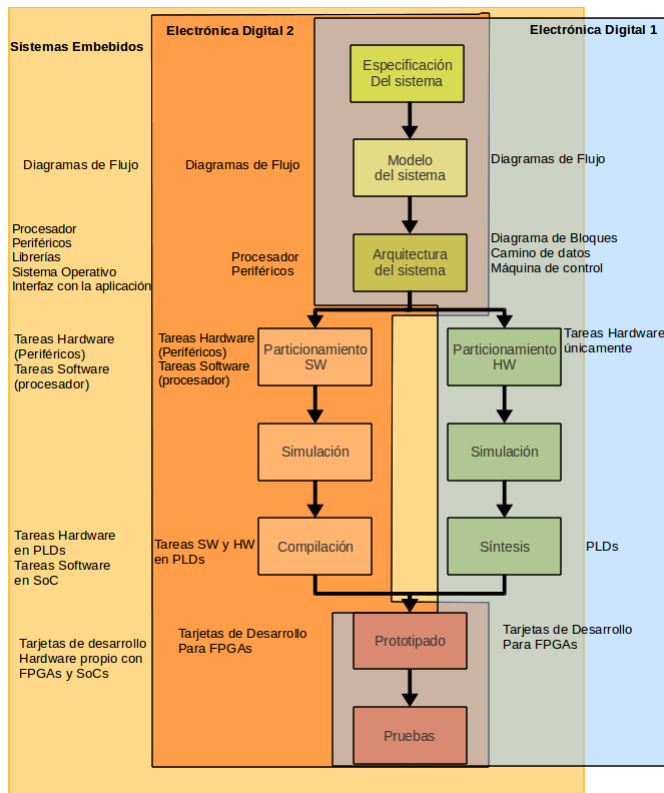


Fig. 7. Metodología de Diseño para el área de Sistemas Digitales

SIE permite conectar de forma fácil (usando dos jumpers) dos GPIOs de la FPGA a las señales de transmisión y recepción del procesador, lo que permite la comunicación serial entre el procesador implementado en la FPGA y el procesador MIPS, creando de esta forma un canal de depuración para las aplicaciones implementadas en la FPGA, eliminando la necesidad de equipo adicional.

#### D. Sistemas Embebidos

El tercer y último curso de la línea integra los conocimientos adquiridos en los cursos anteriores e introduce un elemento nuevo un SoC comercial, SIE posee un SoC basado en un procesador MIPS equipado con los recursos necesarios para ejecutar programas Linux; en este curso se estudiará la arquitectura de los SoC comerciales, las herramientas de programación abiertas disponibles (cadena de herramientas GNU, librerías GNU como libQT), cargadores de Linux (u-boot), el sistema operativo Linux, drivers de periféricos, distribuciones

para sistemas embebidos (openwrt, openembedded, debian). Con esto el estudiante estará en capacidad de crear aplicaciones que utilizan la gran variedad de librerías disponibles en el proyecto FOSS, las mismas que utilizan Nokia, Motorola, Dell, Sony.

1) *Creación de periféricos y drivers* : Además de cubrir el tema de programación de SoC utilizando GNU/Linux, es necesario que se entienda no solo el funcionamiento del sistema operativo Linux, sino como este se comunica y controla los periféricos (tareas HW); para esto, se implementarán periféricos en la FPGA y se estudiará la forma de acceder a ellos utilizando el protocolo establecido por el kernel.

2) *Concepción, Diseño, Implementación y Operación de Sistemas Embebidos*: Durante el periodo académico los estudiantes deben diseñar, implementar y operar un sistema embebido concebido por un equipo de trabajo de tres personas, utilizando como base la plataforma SIE, diseñarán una tarjeta hija (utilizando herramientas abiertas) que implemente la funcionalidad deseada, todos los proyectos deben implementar tareas HW en la FPGA y deben crearse módulos del kernel para controlarlos. Los proyectos realizados hasta el momento pueden encontrarse en la página del proyecto.

#### E. Comunidad hardware copyleft

Una parte importante de este método de enseñanza es la filosofía del proyecto hardware copyleft, por esta razón, cada grupo debe hacer un aporte, suministrando la información completa del proceso de desarrollo y suministrando los archivos necesarios para replicar y/o modificarlo, esto es una consecuencia de la licencia CC-BY-SA.

La experiencia del proyecto FOSS indica muchos miembros de estas comunidades ingresan para suplir necesidades, pero muchos de ellos continúan creando código y prestando servicios a la comunidad porque disfrutan programar. Estos *aficionados* realizan un papel muy importante dentro de la comunidad encargándose de tareas como mejora de la plataforma tecnológica, re-escribiendo secciones de código, documentándolo, respondiendo preguntas, preservando o mejorando la arquitectura. Otros consideran que la reciprocidad es vital para la contribución de código a la comunidad y que la forma de gobierno afecta dramáticamente la participación de programadores voluntarios [S. Shah, 2006]. Las actividades de documentación además de contribuir a mejorar las habilidades de escritura de reportes técnicos ayudan a formar una comunidad que contribuye al crecimiento del proyecto copyleft hardware, los estudiantes ingresan a la lista de desarrolladores de Qi-hardware y solo desde allí se resolverán las dudas, esto se hace con el objetivo de enseñarles a utilizar una herramienta muy poderosa en la que pueden compartir sus inquietudes con miembros más experimentados y mientras participan ayudan a crear un banco de preguntas que pueden ser útiles para futuros miembros. Adicionalmente se obliga a expresarse en un idioma diferente.

Crear estos hábitos ayuda a que los jóvenes sean conscientes de su papel dentro de una comunidad y piensen que sus acciones pueden ayudarla o perjudicarla, los proyectos realizados por ellos podrán ser parte de los recursos de la

comunidad (si la calidad del trabajo lo amerita) y pueden ser la continuación de un esfuerzo prolongado o el punto de partida de un nuevo conocimiento; la licencia CC-BY-SA garantiza que todos los trabajos derivados de este recurso serán parte del mismo, lo que garantiza su crecimiento, la labor de los estudiantes es vital para el uso del recurso común y puede crear miembros que en un futuro formularán políticas y reglas de uso del recurso. Por otro lado, participar en este tipo de proyectos permite crear reputación, la cual puede ser útil para establecer relaciones profesionales, de negocios o personales. El entorno académico es ideal para atraer nuevos miembros a la comunidad hardware copyleft, ya que se trabaja con jóvenes con deseos de ser parte de un grupo y de adquirir conocimientos.

Desde el punto de vista comercial este recurso es muy atractivo ya que permite ahorrar mucho tiempo, esfuerzo y dinero para la creación de nuevos productos. Por otro lado, el concepto de hardware copyleft es una herramienta poderosa para transferir tecnología y conocimientos a los países en vía de desarrollo donde la plataforma tecnológica no se lo suficientemente desarrollada.

#### IV. CONCLUSIONES

El hardware copyleft es una herramienta poderosa para la creación de estas habilidades, ya que proporciona la información necesaria para entender el ciclo completo de concepción, diseño, implementación y operación, (lo cual no es posible obtener cuando se trabaja con plataformas de desarrollo comerciales); proporcionando información detallada sobre el proceso de diseño de plataformas abiertas, que pueden ser utilizadas como referencia para generar nuevos productos comerciales; el acceso a aplicaciones software que permiten la creación de aplicaciones; un canal de comunicación que permite utilizar a la industria manufacturera asiática para la producción en masa; conocimiento de los procesos de fabricación y producción.

Las actividades propuestas en las tres asignaturas del área tienen como objetivo generar en el estudiante las habilidades necesarias que le permitan diseñar sistemas digitales de diferente grado de complejidad, hasta llegar a un sistema que puede ser comercializable y satisface una necesidad de una determinada comunidad, con esto, se evita que el último paso en el proceso de enseñanza sea la simulación; se ilustra el proceso que debe seguirse para que un prototipo se convierta en un producto comercial, lo que contribuirá con la creación de nuevos productos y la generación de empleo.

Se presentó un método de enseñanza para la línea de diseño digital que solo utiliza herramientas abiertas hardware y software, además de los beneficios económicos, este tipo de herramientas permiten estudiar, reproducir y modificar el software y el hardware utilizado, posibilitando su uso en aplicaciones comerciales.

La utilización de herramientas de bajo nivel permite que el estudiante conozca y controle los diferentes pasos de la metodología de diseño y sea capaz de ajustarlas para diferentes situaciones, esto hace que se adquiera un conocimiento sobre la tecnología sin crear dependencia hacia las herramientas

comerciales que realizan la mayoría de los pasos de la metodología de forma automática.

#### V. BIBLIOGRAFÍA

##### REFERENCES

- A. Gerstlauer, D. Gajski., . Technical Report CECS-02-17, and 2002. CECS, UC Irvine. System-level abstraction semantics, Technical Report CECS-02-17. Technical report, CECS, UC Irvine, 2002.
- A. Grove. How America Can Create Jobs. [http://www.businessweek.com/magazine/content/10\\_28/b418604835859](http://www.businessweek.com/magazine/content/10_28/b418604835859) May 2010.
- Creative Commons. Licencias Creative Commons. URL: <http://creativecommons.org/licenses.>, 2004.
- D. Servant. Combining amorphous computing and reactive agent-based systems: a paradigm for pervasive intelligence? In *First international joint conference on Autonomous agents and multiagent systems: part 1*, 2002.
- Edward F. Crawley. The CDIO Syllabus A Statement of Goals for Undergraduate Engineering Education. URL:<http://www.cdio.org>, 2001.
- Gajski D.D., Abdi S., Gerstlauer A., and Schirner G. *Embedded System Design: Modeling, Synthesis, Verification*. Springer, 2009.
- H. Mitsui, H. Kambe, and H. Koizumi. Use of Student Experiments for Teaching Embedded Software Development Including HW/SW Co-Design. *IEEE TRANSACTIONS ON EDUCATION*, 52(3), August 2009.
- Jon Hall. POR GRANDES QUE SEAN...: ASEGURE EL FUTURO DE SU NEGOCIO. *Linux magazine*, ISSN 1576-4079(58):92, 2009.
- Luis Alejandro Cortés. *Verification and Scheduling Techniques for Real-Time Embedded Systems*. PhD thesis, Linköpings universitet Institute of Technology, 2005.
- M. Weiser. Some computer science issues in ubiquitous computing. *Commun. ACM*, 1993.
- Qi Hardware. Qi Hardware Copyleft Hardware Project. URL: <http://en.qi-hardware.com/>.
- R. Stallman. Philosophy of the GNU project. URL: <http://www.gnu.org/philosophy/>, 2007.
- S. Shah. Motivation, Governance, and the Viability of Hybrid Forms in Open Source Software Development. *Management Science*, July 2006.
- R. M. Stallman. *The GNU Operating System and the Free Software Movement Voices from the Open Source Revolution*. O'Reilly and Associates, 1999.
- Texas Instruments. IEEE Std 1149.1 (JTAG) Testability. 1997 Semiconductor Group, 1996.
- W. Spraul, C. Camargo, and A. Wang. Proyecto SAKC. URL:<http://en.qi-hardware.com/wiki/SAKC>.
- Worldwide CDIO Initiative. "Benefits of CDIO" URL:<http://www.cdio.org/benefits-cdio> on November, 2009.