

**METODOLOGÍA PARA LA TRANSFERENCIA
TECNOLÓGICA Y DE CONOCIMIENTOS EN EL
DISEÑO DE SISTEMAS EMBEBIDOS**

UNIVERSIDAD NACIONAL DE COLOMBIA

Carlos Iván Camargo Bareño

7 de abril de 2011

**METODOLOGÍA PARA LA TRANSFERENCIA TECNOLÓGICA Y DE CONOCIMIENTOS EN EL
DISEÑO DE SISTEMAS EMBEBIDOS**

AUTOR: C. Camargo

E-MAIL: cicamargoba@unal.edu.co

Universidad Nacional de Colombia
<http://www.unal.edu.co>

Se garantiza el permiso para distribuir, y/o modificar este documento
bajo los términos de la licencia Creative Commons CC BY-SA

Publicado por la Universidad Nacional de Colombia

Índice general

1. Metodología Para la Enseñanza de Sistemas Embebidos	5
1.1. Introducción	5
1.1.1. Objetivos de la iniciativa CDIO	6
1.1.2. Estructura del plan de estudios CDIO	7
1.1.3. Implementación del Plan de Estudios CDIO	9
1.2. Definición e Identificación de las Habilidades CDIO	10
1.2.1. Introducir, Enseñar y Usar	10
1.2.2. Habilidades CDIO	11
1.2.3. Competencias de las Habilidades CDIO 2 y 3	14
1.2.4. Competencias de las Habilidades C.D.I.O. Sistemas en el contexto Empresarial, Social y Ambiental - Innovación	16
1.3. Integración de las Habilidades CDIO al Plan de Estudios	17
1.3.1. Metodología de diseño	17
1.3.2. Objetivo General	20
1.3.3. Objetivos Específicos	21
1.3.4. Ojbetivos comunes	21
1.3.5. Metodología	22
1.3.6. Integración de SIE con los cursos de la línea de electrónica digital	24
1.3.7. Sistemas Embebidos	28
1.3.8. Actividades	29
1.3.9. Plataforma de Desarrollo SIE	31
1.4. Desarrollo de Métodos de Evaluación	32

Capítulo 1

Metodología Para la Enseñanza de Sistemas Embebidos

1.1. Introducción

La disponibilidad de personal calificado que absorba, asimile y aplique los conocimientos asociados a una tecnología es, a nuestro modo de ver, el punto más importante en el proceso de transferencia tecnológica; como observamos anteriormente, los canales tradicionales de transferencia han demostrado ser poco efectivos en el área bajo estudio; podemos concluir sin temor a equivocarnos que la causa principal del atraso de la industria electrónica (relacionada con el diseño digital) es la poca oferta de personal calificado que permita generar un cambio y pasar de importadores a generadores de soluciones.

Analizando el caso del Departamento de Ingeniería Eléctrica y Electrónica (DIEE) de la Universidad Nacional de Colombia, observamos que los estudiantes son muy hábiles para analizar y para aplicar sus conocimientos en la solución a problemas académicos; sin embargo, la mayoría presenta una gran dificultad al momento de implementar físicamente estas soluciones; esto, porque en la mayoría de las asignaturas no se llega a este punto ya que el proceso termina en la simulación. La falta de contacto de muchos docentes con la industria hace que los programas de estas asignaturas no incluyan como parte de la metodología la experimentación práctica; algunos docentes argumentan que la implementación práctica no es labor de un ingeniero y por esta razón ignoran este proceso; aunque esta opinión es respetable, a nuestro modo de ver, en un país como Colombia donde no existe una oferta considerable de bienes y servicios relacionados con el diseño de sistemas electrónicos (comparado con países de la región) se le deben proporcionar al estudiante las herramientas necesarias para innovar y crear nuevos productos que den solución precisa a problemas locales. La falta de contacto por parte de los estudiantes a la solución de problemas reales origina una desconfianza por parte de la industria hacia los productos generados localmente.

Es importante que los estudiantes utilicen herramientas adecuadas cuando se enfrentan a la implementación física. Hasta hace poco en el DIEE se encontraban trabajos de grado que utilizaban placas de prototipos; se proporcionaban soluciones basadas en herramientas comerciales las cuales eran conseguidas de forma ilegal o se utilizaban demos. Este tipo de soluciones están muy lejos de una solución comercial real, primero por que no proporciona una plataforma física robusta y segundo porque es necesario adquirir software muy costoso. Adicionalmente, no es ético utilizar herramientas comerciales en la enseñanza si no se cuentan con las licencias necesarias; es muy común ignorar este tipo de violaciones a los derechos de autor cuando se trabaja en aplicaciones académicas, aunque el fin del uso sea muy loable no deja de ser ilegal; por esta

razón, una de las premisas de nuestra metodología de enseñanza será el uso de herramientas abiertas que produzcan resultados comparables con las herramientas comerciales.

La metodología que proponemos adopta los conocimientos generados en el proceso de transferencia descrito en el capítulo ?? y se constituye en su aplicación a la academia, con esto esperamos que los estudiantes tomen conciencia de la importancia del uso de esta tecnología y utilicen esta información para crear productos novedosos que den solución a problemas locales. El contenido del curso y metodología se ajustará a la iniciativa CDIO. El Departamento de Ingeniería Eléctrica y Electrónica de la Universidad Nacional de Colombia está realizando el proceso de adaptar la Iniciativa CDIO a su programa académico, y en la mayoría de las asignaturas no se contemplan actividades que ayuden a mejorar las habilidades en la implementación de sistemas, lo que confirma una vez más que la falta de docentes con experiencia en la industria afecta la generación de habilidades necesarias para crear nuevos productos.

1.1.1. Objetivos de la iniciativa CDIO

La iniciativa CDIO ¹ ha sido desarrollada por el MIT con ayuda de académicos, industriales, ingenieros y estudiantes[1] como respuesta a los caminos diferentes que están tomando la educación de la ingeniería y las demandas del mundo real ² Esta iniciativa ha sido adoptada por un creciente número de instituciones académicas a lo largo del mundo. Hacer parte de este esfuerzo mundial nos ayuda a mantener nuestros planes académicos actualizados con los cambios que se realizan en países más industrializados. En este capítulo mostraremos como esta iniciativa se adapta perfectamente a la metodología propuesta en este trabajo ya que adiciona dos componentes importantísimos para la aplicación de la tecnología en la creación de nuevos productos: la implementación y la operación.

La iniciativa CDIO se basa en la suposición de que los egresados de los centros de formación en ingeniería deben ser capaces de: **Concebir, Diseñar, Implementar y Operar** sistemas funcionales en el mundo real. Como mencionamos anteriormente, en Colombia, la mayoría de los centros de formación solo tienen en cuenta la concepción y el diseño, descuidando por completo la implementación y la operación, esto, impide que se generen las habilidades necesarias tenga una estrecha relación con la industria, la cual, requiere productos que pueda comercializar o soluciones a sus necesidades. La frase *en el mundo real* resalta la importancia de trabajar en la solución de problemas que pueden encontrarse en el ejercicio profesional, lo que es muy difícil de determinar cuando los docentes no han tenido contacto con el mundo real. La iniciativa CDIO se enfoca en preparar a los estudiantes con los conocimientos habilidades y aptitudes para ser ingenieros líder; y sus principales objetivos son: [1]:

- 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.

Estos objetivos se adaptan a los requerimientos que se exige a la plataforma tecnológica de un país para que pueda realizar una adecuada absorción del conocimiento transferido para posteriormente transformar ese conocimiento en nuevos productos adaptados a las necesidades del país. Las premisas que capturan la visión, objetivos y fundamentos pedagógicos de la iniciativa son:

- Es posible cumplir las necesidades propias de la profesión mientras se realiza el proceso de concebir, diseñar, implementar y operar sistemas en el contexto de los sistemas de ingeniería.

¹<http://www.cdio.org>

²lo que se aplica perfectamente al estado de la industria electrónica (y a otras industrias) en Colombia, donde la enseñanza en ingeniería se queda a medio camino y no se hace implementación física.

- 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.

Emprendimiento y liderazgo

La situación actual por la que atraviesa la Industria electrónica nacional, requiere que los profesionales en el área tengan las capacidades de emprendimiento y liderazgo necesarias para la creación de nuevas empresas o para la creación de nuevos productos, la Figura 1.1 muestra la relación entre emprendimiento³, liderazgo y las habilidades CDIO.

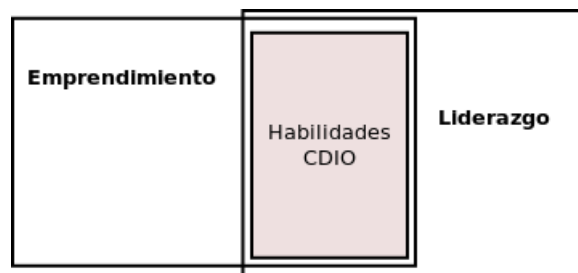


Figura 1.1:

fuelle:[2]

1.1.2. Estructura del plan de estudios CDIO

La figura 1.2 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. Para ser capaces de desarrollar sistemas complejos en ingeniería, los estudiantes deben dominar los fundamentos del *razonamiento y conocimiento técnico*; para trabajar en un entorno moderno basado en grupos de trabajo los estudiantes deben desarrollar *habilidades interpersonales* de comunicación y trabajo en equipo; finalmente para 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*[3]

³El concepto clásico de emprendimiento involucra el re-direccionamiento y movilización de capital y recursos humanos para crear una nueva actividad económica; actualmente, el emprendimiento está asociado a la creación de una nueva empresa con una nueva línea de negocios.



Figura 1.2: Bloques constructores de conocimiento, habilidades y actitudes necesarias para concebir, diseñar, implementar y operar sistemas en el contexto social y empresarial
fuente:[3]

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:

1. Fundamentos avanzados de ingeniería.
2. Fundamentos del núcleo de ingeniería.
3. 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 conocimiento profundo 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.

Habilidades personales, profesionales e interpersonales

Los niveles 2 y 3 se centran en las habilidades personales que debe poseer un individuo para que pueda cumplir con el objetivo de la iniciativa CDIO. El nivel 2 esta compuesto por:

1. Las habilidades profesionales que representan las tres formas de pensar más practicadas por los ingenieros: resolución de problemas; descubrimiento de conocimiento y pensamiento sistémico.
2. 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 equipo de trabajo hace referencia a

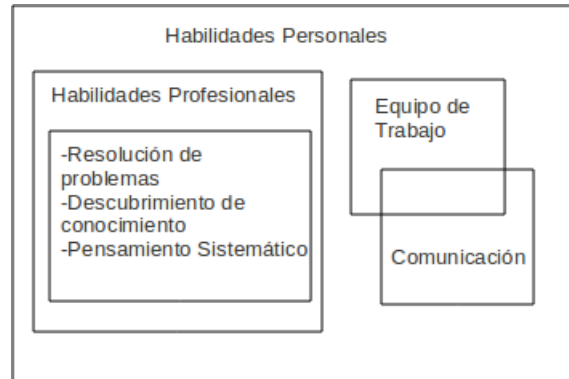


Figura 1.3: Relación entre las habilidades personales, profesionales e interpersonales
fuente:[3]

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 que utilizan los medios orales, escritos, electrónicos y gráficos y en el caso colombiano el uso del idioma Inglés. La Figura 1.3 muestra la relación entre las habilidades de nivel 2 (personales y profesionales) y nivel 3 (interpersonales).

Habilidades CDIO

Habilidades necesarias para concebir, diseñar, implementar y operar sistemas 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.

1.1.3. Implementación del Plan de Estudios CDIO

La Figura 1.4 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 qué 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.

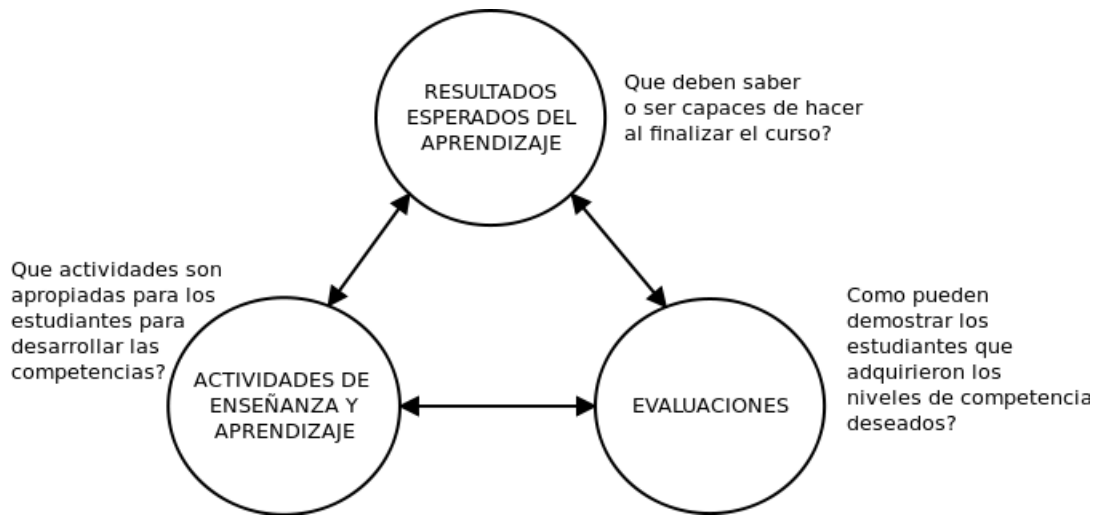


Figura 1.4: Objetivos, actividades, y evaluación:

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.

1.2. Definición e Identificación de las Habilidades CDIO

El primer paso en la implementación del plan de estudios CDIO es definir e identificar las habilidades requeridas en una determinada área del plan de estudios, en este caso en las asignaturas del área de electrónica digital. En el DICE de la UNAL el área de electrónica digital esta compuesta por tres asignaturas para la carrera de ingeniería electrónica: Electrónica Digital 1, Electrónica Digital 2 y Sistemas Embebidos; para la carrera de ingeniería eléctrica está compuesta por Electrónica Digital 1 (la misma en las dos carreras) únicamente.

1.2.1. 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 solo en una asignatura y es necesario que todo el plan académico contribuya a generarla, lo que requiere un consenso del personal académico. Los niveles de competencia seleccionados para indicar el grado 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.

1.2.2. Habilidades CDIO

A continuación se listan las habilidades CDIO que deben desarrollarse o reforzarse en el área de electrónica digital, algunas de las habilidades como la comunicación oral y escrita en inglés es común a la mayoría de las asignaturas, mientras que otras como la integración software - hardware es exclusiva de esta línea.

2 Aptitudes personales y profesionales

- 1 Razonamiento analítico y resolución de problemas
 - 1) Identificación y formulación del problema.
 - 2) Modelamiento.
 - 3) Solución y recomendación.
- 2 Experimentación investigación y descubrimiento de conocimiento
 - 1) Formulación de hipótesis.
 - 2) Investigación experimental.
- 3 Pensamiento sistemático
 - 1) Pensamiento global.
 - 2) Surgimiento e interacciones.
- 4 Pensamiento crítico y creativo y habilidades y actitudes personales
 - 1) Pensamiento creativo.
 - 2) Pensamiento crítico.
 - 3) Toma de conciencia de conocimientos propios y metacognición.
 - 4) Aprendizaje permanente y educar a otros.
- 5 Ética, responsabilidad profesional, equidad y otros valores personales.
 - 1) Ética, integridad y responsabilidad social.
 - 2) Comportamiento profesional y responsabilidad.
 - 3) Confianza y lealtad.

Las aptitudes personales y profesionales que se desean resaltar en esta metodología son:

- Pensamiento sistemático: Los conocimientos adquiridos en esta asignatura son utilizados para contribuir a una solución a un problema de la sociedad. Esta contribución está representada en un sistema digital que utiliza tecnología de punta y para su elaboración se aplicó una metodología de diseño que optimiza recursos económicos proporcionando la funcionalidad requerida.
- Toma de conciencia de conocimientos propios: El estudiante es el responsable de adquirir el conocimiento con la guía del profesor; por lo tanto, debe ser capaz de identificar temas que no ha asimilado completamente y debe realizar actividades que ayuden en la asimilación de nuevos conocimientos.
- Aprendizaje permanente y educar a otros: El aprendizaje debe ser visto por el estudiante como un proceso continuo que no termina al momento de presentar las evaluaciones, o al finalizar los ciclos de formación; la metodología en estos cursos debe forzar al estudiante a buscar fuentes de información para mejorar la asimilación de conocimientos. El trabajo en equipo ayuda a que esta búsqueda sea compartida, de tal forma que el aprendizaje sea colectivo.

Estas habilidades le permitirán al futuro profesional asimilar cambios en las tecnologías, aplicando técnicas de auto-aprendizaje, lo que hace innecesario que un profesional se vea obligado a tomar cursos de extensión para entender un cambio o una nueva característica de la tecnología que utiliza. A nuestro modo de ver los ingenieros deben estar en la capacidad de adaptarse a estos cambios por sí mismos.

3 Habilidades interpersonales, trabajo en equipo y comunicación

- 1 Equipo de trabajo
 - 1) Formar grupos efectivos.
 - 2) Equipo de liderazgo.
 - 3) Equipo técnico y multi-disciplinario.
- 2 Comunicaciones estructuradas
 - 1) Estrategia de comunicación.
 - 2) Estructura de la comunicación.
 - 3) Comunicación escrita.
 - 4) Comunicación electrónica.
 - 5) Presentación oral.
- 3 Comunicación en idioma extranjero
 - 1) Inglés.
- 4 Comunicaciones informales: Relacionarse con los demás
 - 1) Preguntar, escuchar y dialogar.
 - 2) Negociación, compromiso y resolución de conflictos.
 - 3) Establecimiento de conexiones.

Con las habilidades interpersonales se busca formar profesionales que puedan formar grupos de trabajo eficientes, identificando su papel dentro del mismo, facilitando el desarrollo de las actividades de los demás miembros y buscar soluciones conjuntas al problema a resolver y a problemas que surjan del desarrollo del proyecto. Esto es muy importante para el desarrollo de dispositivos digitales, ya que para su elaboración intervienen profesionales de diferentes áreas: diseñadores industriales, programadores, desarrolladores de hardware, personal de mercadeo, ventas y soporte. Por esto, es de suma importancia aprender a utilizar las diferentes formas de comunicación identificando cual es la más eficiente en un determinado caso; el uso de plataformas que permita a miembros de un equipo que se encuentran en diferentes puntos geográficos intercambiar información y discutir temas propios del diseño simplifica el trabajo y ahorra tiempo y dinero.

Por otro lado, es importante que el estudiante adquiera habilidades que le permitan documentar de forma adecuada los procesos de diseño y que el material que desarrolle pueda ser utilizado por otros miembros del equipo de forma rápida; por esto, es importante utilizar herramientas que faciliten la generación de documentos que permitan el acceso a múltiples usuarios.

Las comunicaciones informales son la mayor fuente de información en los proyectos abiertos, usuarios con un determinado problema escriben a una lista de correo o a canales *irc* (internet relay chat)⁴ que tiene como miembros a usuarios más avanzados que pueden contestar todo tipo de preguntas, y al mismo tiempo proporcionan una base de datos con todas las comunicaciones anteriores permitiendo la búsqueda de conversaciones relacionadas con un problema determinado. Estas listas de correo y canales *irc* también pueden ser utilizados para establecer relaciones profesionales que pueden terminar en nuevos proyectos, ofertas de empleo o colaboración.

⁴protocolo de comunicaciones en tiempo real basado en texto, permite conversaciones entre un grupo de personas; todos los usuarios que se encuentran en un canal pueden comunicarse entre sí

4 Concebir, diseñar, implementar y operar sistemas en el contexto social y empresarial

- 1 Contexto externo, social, económico y ambiental
 - 1) Rol y responsabilidad de los ingenieros.
 - 2) Impacto sobre la sociedad y el medio ambiente.
 - 3) Cuestiones y valores actuales.
 - 4) Sostenibilidad y necesidad de un desarrollo sostenible.
- 2 Empresa y contexto empresarial
 - 1) Interesados en la empresa, metas y objetivos.
 - 2) Espíritu empresarial técnico.
 - 3) Trabajo en organizaciones.
 - 4) Finanzas y economía de los proyectos de ingeniería
- 3 Concepción y administración de sistemas en ingeniería.
 - 1) Entender las necesidades y establecer las metas.
 - 2) Definir la función, concepto y arquitectura.
- 4 Diseño
 - 1) Proceso de Diseño.
 - 2) Fases del proceso de Diseño y enfoques.
 - 3) Utilización de conocimiento científico en el diseño.
 - 4) Diseño específico.
 - 5) Diseño multi-disciplinario.
- 5 Implementación
 - 1) Proceso de fabricación hardware.
 - 2) Proceso de Implementación de software.
 - 3) Integración software - hardware.
 - 4) Pruebas, verificación, validación y certificación.
- 6 Liderar esfuerzos en ingeniería
 - 1) Pensar creativamente e imaginar posibilidades.
 - 2) Definir la solución.
 - 3) Crear nuevas formas de solución.
 - 4) Construir y liderar una organización y una organización extendida.
 - 5) Planear y administrar un proyecto hasta su finalización.
 - 6) Innovar - la concepción, diseño e introducción de nuevos bienes y servicios.
 - 7) Innovar - el desarrollo de nuevos dispositivos, materiales o procesos que permitan nuevos bienes y servicios.
- 7 Emprendimiento en ingeniería
 - 1) Creación, formulación y organización de una empresa.
 - 2) Desarrollo del plan de negocios.
 - 3) Finanzas y capitalización.
 - 4) Mercadeo de productos innovadores.
 - 5) Concepción de productos y servicios alrededor de nuevas tecnologías.
 - 6) Sistema de innovación, redes, infraestructura y servicios.

- 7) Construyendo el equipo e iniciando el proceso de ingeniería.
- 8) Manejo de la propiedad intelectual.

Este grupo de habilidades son las que marcan diferencia con las otras asignaturas del plan de estudio ya que como se mencionó anteriormente, la mayoría de los cursos no contemplan la implementación de sistemas reales donde se apliquen los conocimientos que se intentan transmitir. Debido a esto, es importante recalcar el papel de los ingenieros en la sociedad, la facultad de ingeniería de la universidad nacional tiene como parte de sus objetivos:

- Crear y asimilar críticamente el conocimiento en los campos avanzados de las ciencias, la técnica, la tecnología, el arte y la filosofía.
- Estudiar y analizar los problemas nacionales y proponer, con independencia, formulaciones y soluciones pertinentes, convirtiéndose así en conciencia crítica de la nación.
- Hacer partícipes de los beneficios de su actividad académica e investigativa a los sectores sociales que conforman la nación Colombiana.

Como establecimos en el primer capítulo de este trabajo, el problema actual de la industria electrónica (relacionada con el diseño digital) en Colombia es el atraso y desactualización de las metodologías de diseño y fabricación, originado por la falta de personal calificado en el manejo de nuevas tecnologías y metodologías. Por esta razón, es importante que los estudiantes conozcan la realidad de este sector y sean concientes de su responsabilidad en la solución de sus problemas.

Como estudiantes pueden contribuir a difundir los resultados de su actividad académica proporcionando información que pueda ser utilizada por quien este interesado en adquirir conocimientos relacionados con el diseño digital; por esto, una de las actividades que se desarrollan en estas asignaturas es la documentación del proceso de diseño de un problema específico que deben resolver grupos de trabajo, esta metodología y esta información se almacena en el portal público *linuxenaja* y puede ser utilizada por quien esté interesado.

Una parte vital en el proceso de enseñanza es el uso de metodologías de diseño para la creación de soluciones a problemas específicos, por esta razón se trabaja con la misma metodología en los tres cursos y se explican los beneficios de usar todas sus etapas; esto con el objetivo de crear buenos hábitos de diseño en los estudiantes. A diferencia de otras asignaturas en esta línea es necesario implementar físicamente las soluciones propuestas, se resalta que la simulación es un paso importante del proceso de diseño pero no es punto final. El uso de plataformas con recursos limitados para la implementación física hace que los estudiantes optimicen la arquitectura de sus diseños y aprendan a buscar alternativas más económicas pero que cumplan con los requerimientos del sistema. El problema a solucionar es formulado por el equipo de trabajo y ellos deben proporcionar las especificaciones del sistema que resolverá este problema, los estudiantes poseen libertad total en la funcionalidad del mismo, lo que ayuda a la creación de habilidades en la definición de funciones y arquitecturas.

Con esto se espera que el estudiante sea conciente de la situación del país y que sienta la necesidad de crear nuevos productos que ayuden a solucionar un problema específico y de paso ayudar a la creación de empleo creando y comercializando nuevos productos diseñados localmente.

1.2.3. Competencias de las Habilidades CDIO 2 y 3

La tabla 1.1 muestra las competencias IEU para las *aptitudes personales y profesionales* de las tres asignaturas del área de electrónica digital.

Competencias de las habilidades CDIO nivel 2 y 3			
APTITUDES PERSONALES Y PROFESIONALES	Nivel 1		
	E. Dig1	E. Dig2	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 sistemático</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. Digital1	E. Digital1	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

Cuadro 1.1: Competencias para los niveles 2 y 3 CDIO

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

La tabla 1.2 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.

HABILIDADES CDIO	Nivel 1		
	E. Digital1	E. Digital1	Sist. Emb.
Contexto Externo, Social, Económico y Ambiental	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
Empresa y contexto empresarial	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
Concepción y Administración de Sistemas en Ingeniería.	IEU		
28 Entender las necesidades y establecer las metas	IEU	EU	U
29 Definir la función, concepto y arquitectura	IEU	EU	U
Diseño	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
Implementación	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

Cuadro 1.2: Competencias para CDIO

Como puede verse en la tabla 1.2 el componente distintivo del plan de estudios propuesto es el nivel de competencias de las habilidades relacionadas con diseño e implementación; en casi todas ellas se utilizan estas habilidades, lo que significa que se realizarán actividades que impliquen diseño e implementación de sistemas digitales utilizando tecnología de punta y técnicas de diseño y fabricación modernas; lo que es imprescindible si se espera que nuestros profesionales lideren esfuerzos que deriven en la creación de empresas y de nuevos productos bienes y servicios.

1.3. Integración de las Habilidades CDIO al Plan de Estudios

1.3.1. Metodología de diseño

El proceso de diseño de un Sistema Embebido comienza con la *especificación del sistema* (ver figura 1.5), en este punto se describe la funcionalidad y se definen las restricciones físicas, eléctricas y económicas. Esta especificación debe ser muy general y no deben existir dependencias (tecnológicas, metodológicas) de ningún tipo, se suele utilizar lenguajes de alto nivel, como UML, C++ para realizar la descripción. La especificación puede ser verificada a través de una serie de pasos de análisis cuyo objetivo es determinar la validez de los algoritmos seleccionados, por ejemplo, determinar si el algoritmo siempre termina, los resultados satisfacen las especificaciones. Desde el punto de vista de la re-utilización, algunas partes del funcionamiento global deben tomarse de una librería de algoritmos existentes.

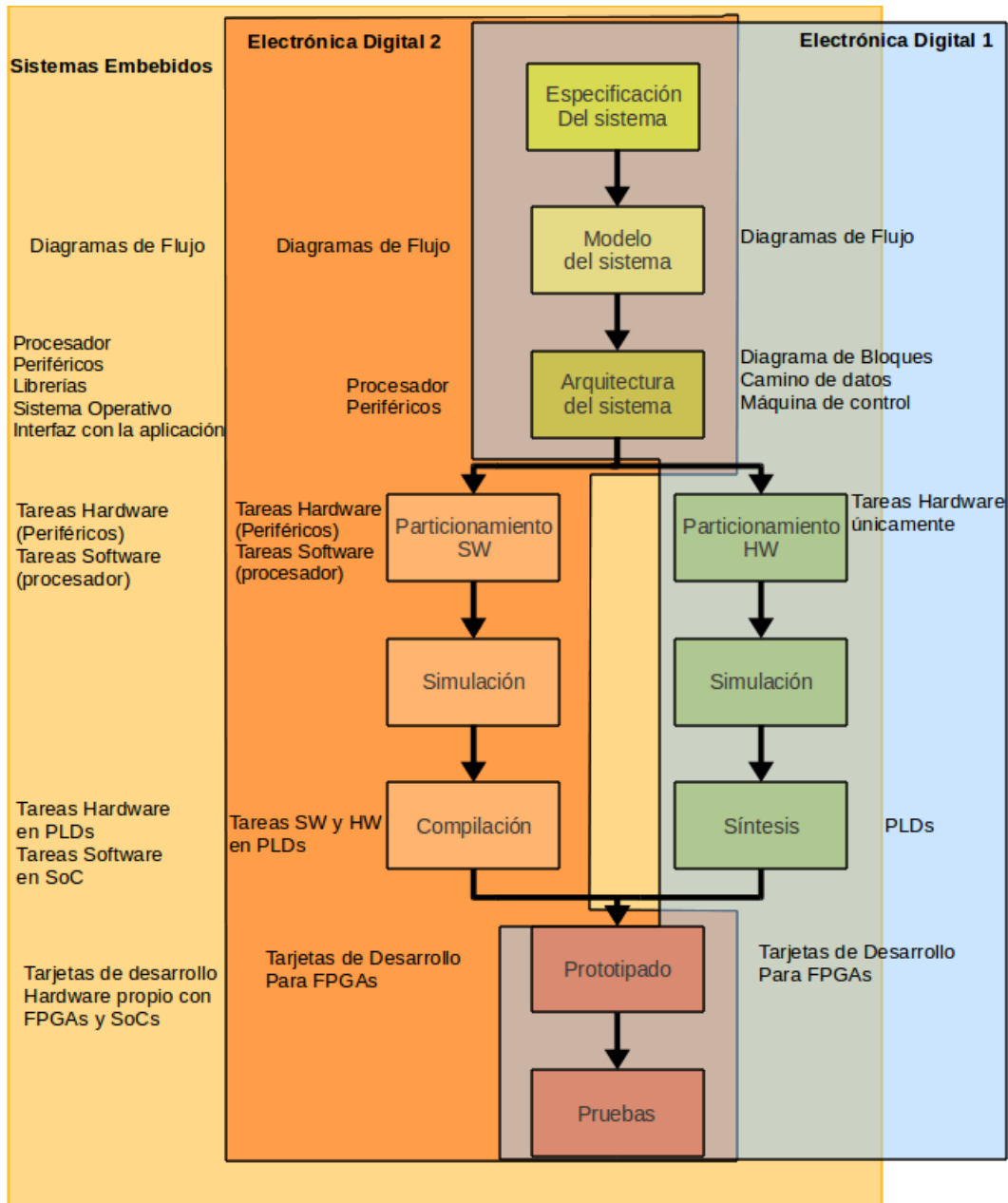


Figura 1.5: Metodología de Diseño para el área de Sistemas Digitales

Una vez definidas las especificaciones del sistema se debe realizar un modelamiento que permita extraer de estas la funcionalidad. El modelamiento es crucial en el diseño ya que de él depende el paso exitoso de la especificación a la implementación. Es importante definir que modelo matemático debe soportar el entorno de diseño. Los modelos más utilizados son: máquinas de estados finitos, diagramas de flujos de datos, sistemas de eventos discretos y redes de petri. Cada modelo posee propiedades matemáticas que

pueden explotarse de forma eficiente para responder preguntas sobre la funcionalidad del sistema sin llevar a cabo dispendiosas tareas de verificación. Todo modelo obtenido debe ser verificado para comprobar que cumple con las restricciones del sistema.

Una vez se ha obtenido el modelo del sistema se procede a determinar su *arquitectura*, esto es, el número y tipo de componentes y su inter-conexión. Este paso no es más que una exploración del espacio de diseño en búsqueda de soluciones que permitan la implementación de una funcionalidad dada, y puede realizarse con varios criterios en mente: costos, confiabilidad, viabilidad comercial. Las opciones de implementación que se estudiarán son:

- Componente HW y SW integrado en un dispositivo semiconductor (SoC): En la actualidad existen muchas compañías que fabrican procesadores de 32 bits integrados a una gran variedad de periféricos, lo cual simplifica el diseño y reduce costos. Este tipo de implementación es muy popular en los dispositivos de consumo masivo (reproductores de MP3, consolas de juego, etc), debido a los grandes niveles de producción resulta más económico contar con un dispositivo que integre el mayor número de funcionalidades, esto disminuye el costo de componentes, el área de circuito impreso.
- Componente SW en un SoC y componente HW en un dispositivo lógico programable (PLD): Cuando no existen en el mercado un SoC con la cantidad de periféricos requerida para una determinada aplicación, o con una funcionalidad específica, es necesario recurrir a la utilización de dispositivos comerciales que implementen dicha operación; en algunas ocasiones el periférico puede realizar funciones poco comunes y no se proporciona comercialmente, por lo que es necesario implementar estas funcionalidades en una FPGA y conectarla al SoC de tal forma que este la vea como uno más de sus periféricos.
- Componente SW y HW en una FPGA: Esta es tal vez la opción más flexible, pero la de menor desempeño, ya que al utilizar los recursos lógicos de la FPGA para la implementación del procesador (softcore) la longitud de los caminos de interconexión entre los bloques lógicos aumentan el retardo de las señales, lo cual disminuye la máxima velocidad de funcionamiento. Los procesadores *softcore* más populares en la actualidad son:
 - Microblaze y Picoblaze de Xilinx.
 - NIOS de Altera.
 - Leon de Gaisler Research.
 - LatticeMico32 (LM32) de Lattice Semiconductors.
 - OpenRisc de opencores.

De los anteriores procesadores el Leon, el LM32 y el Openrisc proporcionan el código fuente en un lenguaje de descripción de hardware, los de las empresas Xilinx y Altera entregan un código a nivel de compuertas que no permite su estudio y modificación; por esas razones y debido a que no se necesita una FPGA con mucha capacidad lógica en estos cursos se trabajará con el LM32 de Lattice.

Utilizando como base la arquitectura obtenida en el paso anterior las tareas del modelo del sistemas son mapeadas dentro de los componentes. Esto es, asignación de funciones a los componentes de la arquitectura. Existen dos opciones a la hora de implementar las tareas o procesos:

- 1 Implementación software: La tarea se va a ejecutar en un procesador ya sea el del SoC o el softcore LM32.
- 2 Implementación hardware: La tarea se va a ejecutar en un sistema digital dedicado implementado en un PLD.

Para cumplir las especificaciones del sistema algunas tareas deben ser implementadas en hardware, esto con el fin de no ocupar al procesador en tareas cíclicas, un ejemplo típico de estas tareas es la generación de bases de tiempos. La decisión de que tareas se implementan en SW y que tareas se implementan en HW recibe el nombre de *particionamiento*, esta selección es fuertemente dependiente de restricciones económicas y temporales.

Las tareas software deben compartir los recursos que existan en el sistema (procesador y memoria), por lo tanto se deben hacer decisiones sobre el orden de ejecución y la prioridad de estas. Este proceso recibe el nombre de *planificación*. En este punto del diseño el modelo debe incluir información sobre el mapeo, el particionamiento y la planificación del sistema.

Las siguientes fases corresponden a la implementación del modelo, para esto las tareas hardware deben ser llevadas al dispositivo elegido (ASIC o FPGA) y se debe obtener el "ejecutable" de las tareas software, este proceso recibe el nombre de *síntesis* HW y SW respectivamente, así mismo se deben sintetizar los mecanismos de comunicación.

El proceso de prototipado consiste en la realización física del sistema, esto es, la fabricación de la placa de circuito impreso, el montaje de los componentes y la fabricación de la caja. Finalmente, el sistema físico debe someterse a pruebas para verificar que se cumplen con las especificaciones iniciales y con las normas propias de su aplicación. Como puede verse en el flujo de diseño existen realimentaciones que permiten depurar el resultado de pasos anteriores en el caso de no cumplirse con las especificaciones iniciales.

Esta metodología de diseño se utilizará en todos los cursos de la línea, pero se utilizarán diferentes opciones de implementación en los diferentes cursos, como se puede apreciar en la figura 1.5; en electrónica digital 1 solo se implementarán tareas hardware en una FPGA, esto con el fin de enseñar al estudiante los conceptos básicos de la electrónica digital, y el uso de máquinas de estado algorítmicas para la implementación de tareas hardware; en electrónica digital 2 se estudiará la arquitectura de los SoC (utilizando el LM32 de lattice) se utilizarán las herramientas de compilación GNU para escribir las tareas software y se crearán periféricos que implementen tareas hardware y finalmente se estudiarán las diferentes formas de comunicación entre las tareas hardware y software; finalmente en el curso sistemas embebidos, se utilizará un SoC comercial junto con la cadena de herramientas GNU y el sistema operativo Linux para estudiar la arquitectura y programación de los sistemas embebidos modernos.

1.3.2. Objetivo General

Generar en el estudiante las habilidades necesarias para concebir, diseñar, implementar y operar sistemas digitales complejos que satisfagan necesidades de la sociedad y proporcionar un canal para la transferencia de tecnología y conocimiento a la Industria Colombiana.

Electrónica Digital 1

Concebir y definir las especificaciones y requerimientos de un sistema digital, modelar su funcionamiento, y realizar la implementación siguiendo la metodología de diseño de sistemas embebidos utilizando únicamente tareas hardware.

Electrónica Digital 2

Concebir, definir las especificaciones, modelar, diseñar un sistema digital siguiendo la metodología de diseño de sistemas embebidos y realizar su implementación óptima utilizando tareas hardware (implemen-

tadas en un PLD) y tareas software (que se ejecutan en un procesador).

Sistemas Embebidos

Concebir, diseñar, e implementar un sistema digital complejo utilizando la metodología de diseño de sistemas embebidos, un SoC comercial y un sistema operativo para su implementación.

1.3.3. Objetivos Específicos

1.3.4. Ojbetivos comunes

- Identificar las especificaciones funcionales del sistema, su arquitectura de alto nivel y definir su descomposición en elementos
- Explicar las actividades en las etapas del proceso de diseño,
- Desarrollar el pensamiento sistémico.
- Modelar funcionalmente sistemas digitales.
- Diseñar pruebas para comprobar el correcto funcionamiento de los sistemas implementados.
- Leer y entender material técnico escrito en inglés.
- Implementar un sistemas embebido (hardware o hardware/software) para cumplir una tarea determinada que cumpla con una necesidad real (Obtener e interpretar las necesidades del consumidor) utilizando técnicas, herramientas y procesos adecuados.
- Estudiar y aplicar el concepto de la re-utilización de código.
- Desarrollar trabajo en equipo incluyendo presentaciones, describiendo los diversos roles y responsabilidades.
- Documentar los diseños realizados para crear una base de datos que contribuya a la difusión del conocimiento adquirido.

Electrónica Digital 1

- Estudiar las fases de la metodología de diseño para sistemas embebidos.
- Estudiar los dominios de diseño estructural, funcional y físico.
- Estudiar los lenguajes de descripción de hardware.
- Estudiar los componentes básicos de la lógica combinatoria y secuencial.
- Estudio de las máquinas de estado algorítmicas.

Electrónica Digital 2

- Estudiar los requisitos para un particionamiento Hardware / Software óptimo.
- Estudiar la arquitectura de un procesador, micro-arquitectura, set de instrucciones, interrupciones, direccionamiento, memorias.
- Estudiar el proceso de implementación de tareas software.
- Estudiar la integración Software-Hardware.
- Diseñar pruebas para comprobar el correcto funcionamiento de los sistemas implementados.

Sistemas Embebidos

- Realizar aplicaciones que requieran diseño multi-disciplinario.
- Estudiar y realizar el proceso de fabricación hardware.
- Estudiar el principio básico de los sistemas operativos.
- Describir la integración de software en hardware electrónico
- Entender diagramas de circuitos electrónicos de sistemas digitales, identificar sus componentes y su función.
- Estudiar diseños software y hardware existentes para entender su funcionamiento, arquitectura y adquirir experiencia en el diseño.
- Hacer parte de listas de discusión de temas técnicos que usen el inglés como lenguaje.
- Utilizar medios electrónicos para documentar procesos de diseño.

Con estos objetivos se busca que al cursar todas las asignaturas del área el estudiante adquiera los conocimientos que se resumen en la figura 1.6.

1.3.5. Metodología

Todas las actividades que se realizarán en estos cursos están encaminadas a generar habilidades necesarias para concebir, diseñar, e implementar sistemas digitales complejos, y están articuladas alrededor de una única metodología de diseño. Los tres cursos tienen un carácter teórico-práctico, el componente teórico tratará los diferentes temas de forma general, con el fin de no crear dependencia con las herramientas utilizadas (lo que permitirá realizar actualizaciones de forma fácil). En el componente práctico se tratarán temas específicos de manejo de las herramientas (lenguajes de descripción de hardware, lenguajes de programación, manejo de plataformas de desarrollo) y como estas se relacionan con la metodología de diseño utilizada.

El estudiante debe estudiar, profundizar y comprobar algunos temas tratados en clase y debe leer previamente la documentación que se encuentra disponible en el sitio web de los cursos. Adicionalmente, debe formar grupos de trabajo para realizar actividades a lo largo del semestre.

Durante el semestre se trabajará para definir las especificaciones, diseñar e implementar un dispositivo que resuelva una determinada necesidad (con la complejidad adecuada para cada curso), en la sesión teórica

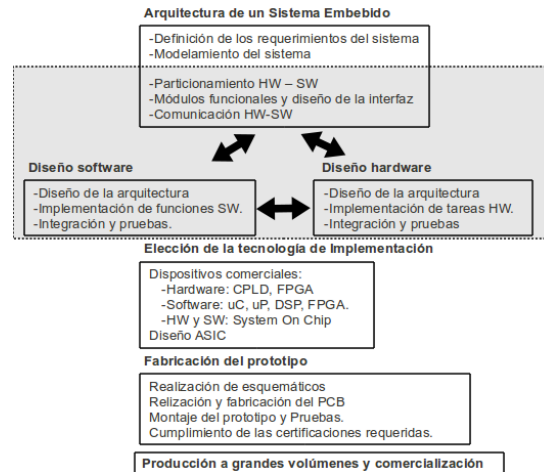


Figura 1.6: Educación de sistemas embebidos. Tomada de:[4] y modificada

se tratarán aspectos relacionados con la concepción, diseño, identificación y definición de las funciones de los componentes del sistema, mientras que en el laboratorio se tratarán temas relacionados con la implementación de dichos componentes sobre PLDs o SoC. Se deben realizar presentaciones del avance, indicando las razones que se tuvieron en cuenta en cada decisión y como se resolvieron los problemas encontrados, todo este proceso debe documentarse en la wiki del portal *linuxenaja*, esto último para formar un banco de proyectos que pueda ser utilizado como referencia por quien este interesado.

SIE: Plataforma abierta para el desarrollo de sistemas embebidos

En el mercado existe una gran variedad de plataformas que pueden ser utilizadas en el estudio de sistemas embebidos, sin embargo, no todas son adecuadas para la implementación del método que proponemos ya que la plataforma que se utilice debe proporcionar toda la información necesaria para poder entenderla, programarla, replicarla y modificarla; esto con el fin de proporcionar al estudiante una herramienta que pueda ser utilizada en el desarrollo de nuevos productos comerciales, para que se cumplan estas condiciones 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*), pudiera ser utilizada en los tres cursos del área y se convirtiera en una herramienta más para la difusión de los conocimientos adquiridos en este estudio a quien se encuentre interesado.

La Figura 1.7 muestra el diagrama de bloques de la plataforma SIE, en ella podemos encontrar un proce-

sador que posee periféricos para 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.

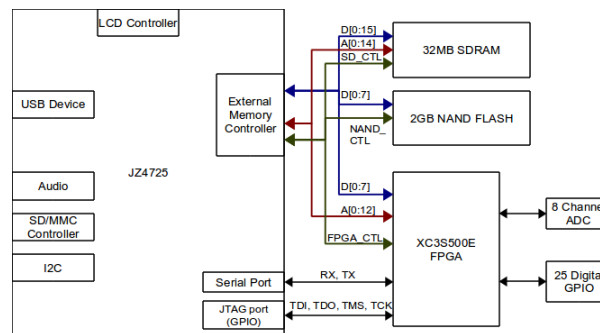


Figura 1.7: Estructura de la plataforma de desarrollo SIE

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. SIE posee un sistema de archivos basado en el proyecto *openwrt* y dispone de una gran cantidad de aplicaciones y librerías que pueden ser compiladas en un computador tradicional, siguiendo las instrucciones contenidas en la wiki del proyecto.

1.3.6. Integración de SIE con los cursos de la línea de electrónica digital

En la actualidad SIE está siendo utilizada en los cursos de arquitectura de computadores y sistemas embebidos de la Universidad Nacional de Colombia sede Bogotá, la metodología utilizada se centra en la realización de un proyecto durante el período académico, los estudiantes deben realizar el proceso de concepción, diseño e implementación de una aplicación que da solución a un determinado problema, cada grupo integrado por tres estudiantes está encargado de llevar una *bitácora* del proyecto en la wiki servidor del portal *linuxencaja*, incluyendo toda la información generada durante el proceso de diseño (diagramas de flujo, diagramas de bloque, simulaciones) el único requisito es la construcción de una placa hija (diseñada en la herramienta abierta Kicad) con la funcionalidad que se le desea dar a la plataforma SIE.

Electrónica digital 1

En el primer curso del área de diseño digital en la UNAL se realiza el estudio, diseño e implementación de máquinas de estado algorítmicas utilizando la metodología de diseño presentada anteriormente y la

herramienta gratuita de Xilinx *Webpack*, los estudiantes implementarán sus diseños utilizando lenguajes de descripción de hardware (VHDL, verilog), como resultado de este proceso se obtendrán 3 archivos uno con extensión **.bit** que se utiliza para configurar a la FPGA con la funcionalidad deseada; un archivo con extensión **pad.csv** que indica la función asignada por el diseñador a todos los pines de la FPGA; y un archivo con extensión **.vcd** que contiene los resultados de la simulación del sistema ⁵. SIE proporciona un canal de comunicación entre el procesador y el puerto JTAG de la FPGA que puede ser utilizado para:

- Configuración: Carga del archivo con extensión **.bit** con la funcionalidad deseada a la FPGA; este archivo puede ser transferido a la FPGA utilizando el cable usb y el protocolo de comunicaciones ssh, cuando este archivo de configuración se encuentre en el sistema de archivos de SIE se puede utilizar la aplicación *xc3sprog* para configurar la FPGA.
- Prueba a baja frecuencia: Como mencionamos anteriormente, el protocolo JTAG permite la aplicación de vectores de prueba a un dispositivo semiconductor y la recolección de la respuesta a estos estímulos utilizando 4 señales (TDL, TDO, TMS, TCK).

Una aplicación (propia y abierta) que tiene como entrada los archivos que contienen la información de los pines (**pad.csv**) y los resultados de la simulación (ver figura 1.8) extrae la información necesaria para aplicar los vectores de prueba al circuito implementado en la FPGA y capturar su respuesta a estos estímulos ⁶, utilizando la instrucción **INTEST** del protocolo JTAG. Los resultados de esta prueba son graficados en el LCD de SIE y pueden ser exportados a un archivo tipo imagen; esta herramienta puede verse como una combinación de un analizador lógico y un generador de vectores de prueba de bajo costo. Con esto, se proporcionan las herramientas necesarias para realizar todas las etapas del flujo de diseño con solo una plataforma.

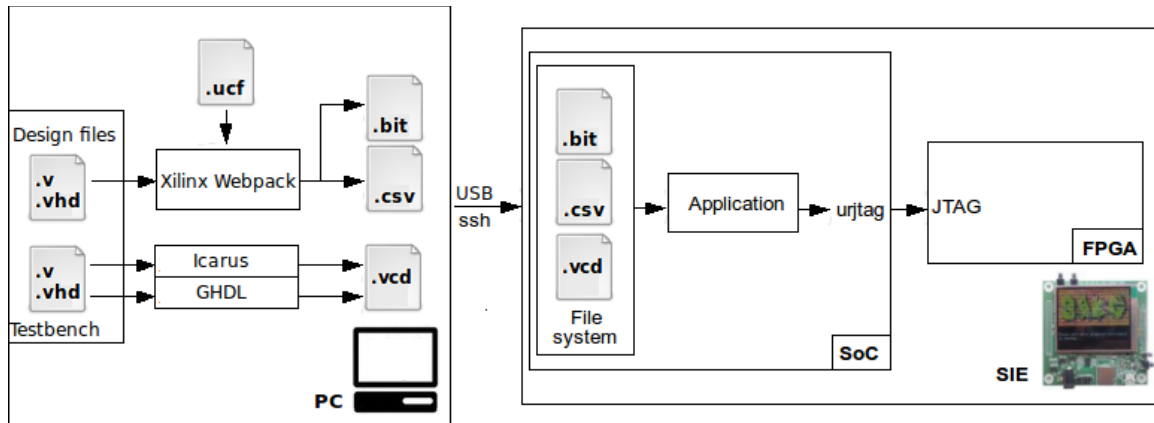


Figura 1.8: Flujo de diseño hardware

De forma intensional SIE no cuenta con memorias, displays de 7 segmentoso conectores conectados a la FPGA; experiencias con el manejo de las plataformas de desarrollo que poseen una gran cantidad de periféricos indican que los estudiantes no adquieren habilidades necesarias para la lectura de esquemáticos ni de hojas de especificaciones de componentes al utilizarlas, ya que normalmente se suministran ejemplos funcionales que son utilizados por los estudiantes; la falta de este tipo de recursos en SIE obliga a los

⁵Este archivo lo generan las herramientas de simulación abiertas *icarus* y *ghdl*

⁶para el control del puerto JTAG se modificó la herramienta *urtag*

estudiantes a investigar sobre la adecuada conexión de estos componentes y a la construcción de una tarjeta hija que implemente la funcionalidad deseada.

Electrónica digital 2

Como se mencionó anteriormente, en este curso se busca que el estudiante entienda la diferencia entre tareas hardware y software, y los canales de comunicación entre ellas; para esto se implementarán las tareas software en el procesador *softcore* LM32 de Lattice (o cualquiera con el código fuente disponible); para esto, es necesario estudiar la arquitectura de la unidad de procesamiento, su set de instrucciones, manejo de interrupciones, comunicación con las memorias de datos y de instrucciones y su forma de programación. El código fuente facilita el estudio de las operaciones internas del procesador al permitir el seguimiento de señales específicas, a manera de ejemplo en la figura 1.9 se muestra el acceso a una memoria externa para tipos de datos *unsigned char*

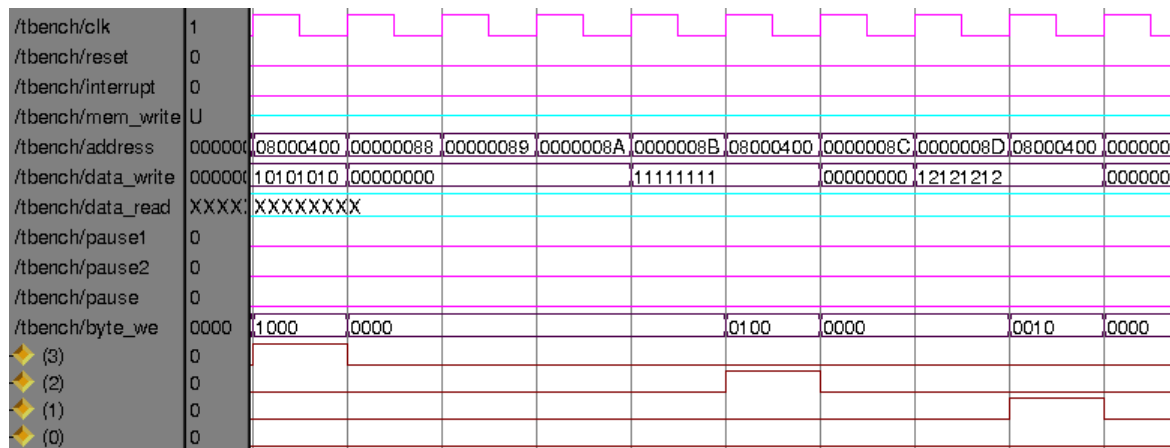


Figura 1.9: Simulación del acceso de datos a memoria externa en un procesador mips

Este tipo de simulaciones permiten entender perfectamente conceptos relacionados con el funcionamiento del sistema (diferencia de tipos de datos en este ejemplo) lo que no es posible realizar si se utilizan dispositivos comerciales en el aprendizaje de la arquitectura de computadores ya que los simuladores existentes se enfocan en el contenido del banco de registros y en la memoria y no pueden mostrar (hasta el momento) las señales internas. Pero, ¿Porqué no implementar un procesador propio para la realización de este estudio? esta metodología se utilizó en el pasado en muchos cursos de arquitectura de computadores y es utilizada en la actualidad, su principal desventaja radica en la falta de herramientas para realizar la programación de estas nuevas arquitecturas, por lo que es necesario programarlas en lenguaje de máquina lo que resulta ser muy tedioso aún para aplicaciones sencillas. La metodología que proponemos aquí permite utilizar la cadena de herramientas GNU existente para el procesador LM32, esta cadena de herramientas permite trabajar con lenguaje ensamblador, C y C++ y proporciona un flujo de diseño software que puede ser utilizado otros procesadores comerciales. En la figura 1.10 se muestra el flujo de diseño al utilizar un procesador *softcore*

Una pregunta válida en este punto (y ya ha sido realizada por los estudiantes) es: ¿Cuándo se enseñará el manejo de microcontroladores y microprocesadores comerciales? La pregunta es muy importante ya que estos dispositivos son muy utilizados y en una gran gama de aplicaciones no es necesario utilizar complejos SoC. La respuesta a la pregunta es que no es necesario enseñar el manejo de estos dispositivos ya que con los conocimientos adquiridos en este curso se cuenta con la información necesaria para que los estudiantes

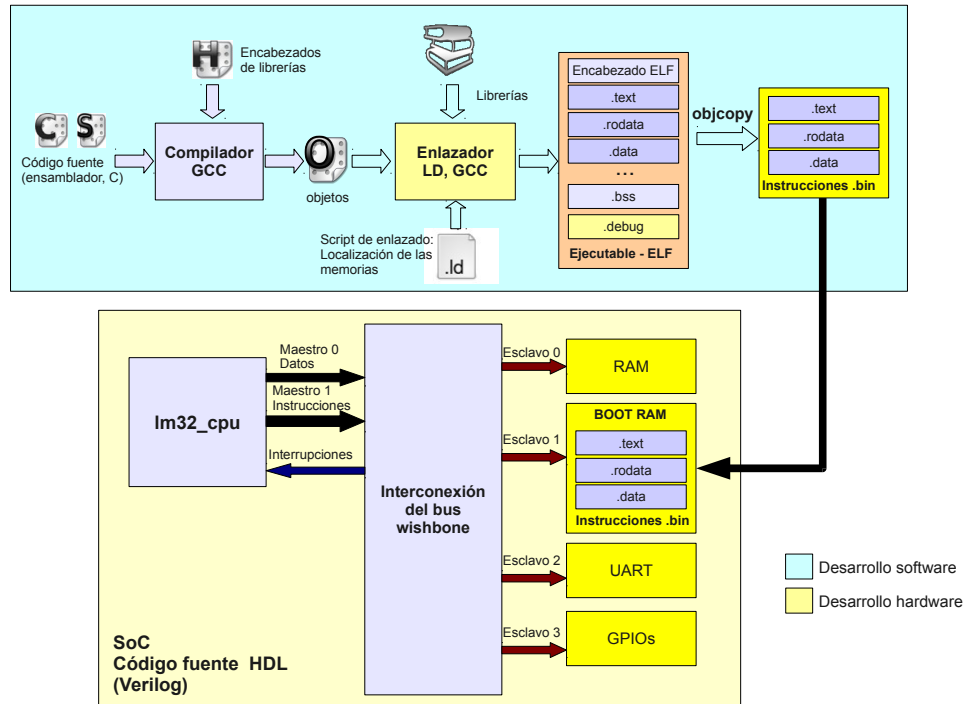


Figura 1.10: Flujo de diseño hardware/software al utilizar un procesador *softcore*

puedan hacerlo por ellos mismos; experiencias con los estudiantes del DIEE de la UNAL muestran que gracias a que la cadena de herramientas de compilación GNU soportan una gran variedad de procesadores de 8, 16 y 32 bits, es posible cambiar de CPU de forma fácil y si el código fué escrito de forma adecuada, utilizarlo con pequeñas modificaciones en diferentes procesadores.

En la página de descargas del proyecto SIE⁷ se pueden obtener implementaciones en VHDL para el procesador MIPS MLITE y en verilog para el procesador de lattice LM32, se suministran los archivos necesarios para el desarrollo de aplicaciones software utilizando la cadena de herramientas GNU y los archivos para realizar la simulación del softcore utilizando *ghdl* e *icarus verilog*; A diferencia de Microblaze de Xilinx (suministra un formato tipo netlist que utiliza los bloques básicos de Xilinx) *plasma* y *lm32* permiten realizar modificaciones en la arquitectura del procesador, permitiendo la creación de nuevas instrucciones y co-procesadores. Adicionalmente, se proporcionan herramientas que inicializan las memorias internas de los procesadores softcore con el código generado por la cadena de compilación; lo que permite realizar un estudio completo del funcionamiento del procesador. En este curso se estudia la arquitectura de un sistema sobre silicio (SoC) y los estudiantes deben concebir, diseñar e implementar un periférico dedicado, conectarlo al SoC y escribir el código en C que lo controla e implementa la funcionalidad. La FPGA de SIE posee recursos limitados, lo que obliga a los estudiantes a optimizarlos y a buscar en el mercado circuitos integrados que les permita cumplir con la funcionalidad requerida.

En este curso el procesador es utilizado como herramienta de configuración de la FPGA, los archivos *.bit* son transferidos al sistema de archivos de SIE utilizando el protocolo *ssh* y desde allí son transferidos a la FPGA utilizando *xc3sprog* o *urjtag*. Ejecutando la aplicación *minicom* en el procesador de SIE, es posible utilizar el puerto serie del procesador implementado en la FPGA como canal de depuración; por lo que no

⁷<http://projects.qi-hardware.com/index.php/p/nn-usb-fpga/source/tree/master/>

es necesario utilizar programadores o aplicaciones externas.

1.3.7. Sistemas Embebidos

QI hardware, proporciona las herramientas software necesarias para el desarrollo de aplicaciones software en los procesadores MIPS de Ingenic JZ4720/25/40; en la actualidad existen aplicaciones abiertas para: programar las memorias no volátiles *usbboot*; inicializar la plataforma y cargar la imagen de linux. *u-boot* crear la imagen del kernel de Linux; crear un sistema de archivos *openwrt*; crear instaladores de un gran número de aplicaciones *opkg-openwrt*; desarrollo software utilizando librerías gráficas como QT o GTK. *openwrt*.

SIE utiliza un procesador JZ4725 y tiene una arquitectura compatible con el producto de QI-hardware *BEN NANONOTE*, por esta razón, todo el desarrollo realizado para BEN es aplicable a SIE, esto garantiza el mantenimiento, actualización y disponibilidad del software necesario para el desarrollo de aplicaciones.

En este curso se utiliza el procesador MIPS para ejecutar tareas de visualización, comunicación, control e interfaz con el usuario. Se utilizan librerías gráficas de alto nivel como QT (de Nokia) para realizar la interfaz, se desarrollan módulos del kernel y programas en espacio de usuario para el control de periféricos dedicados (implementados en la FPGA). Con esto se proporciona a los estudiantes herramientas que están siendo utilizadas en la actualidad por los grandes fabricantes de dispositivos digitales como Nokia, Dell, Hewlett Packard.

La figura ?? muestra la arquitectura de la etapa de comunicación entre el procesador y la FPGA, las señales *DATA*, *NCS2*, *we0_n*, *address* provenientes del procesador no se encuentran en fase con la señal de reloj de la FPGA, por lo que deben ser sincronizadas con este; (el módulo SYNC se encarga de esta tarea). Debido a que no se pueden implementar buses tri-estado en el interior de la FPGA (sólo se pueden utilizar en los pines de la FPGA), por esta razón, los periféricos deben tener dos buses de datos uno de entrada (Color Naranja) y otro de salida (color verde). Se debe proporcionar un circuito a la salida de los buses de los periféricos que permita el paso de la información del periférico seleccionado (en la figura se representa este elemento como un Multiplexor). Debido a que la FPGA es mucho más rápida que el procesador, es necesario generar un pulso con la duración de un ciclo de reloj de la FPGA para la señal de escritura *we0_n*; esto se hace para evitar que se realice más de una operación de escritura (esta tarea es realizada por el módulo Write Pulse Generator, en la figura puede verse el diagrama de tiempos que representa su funcionamiento). Un buffer tri-estado debe controlar el acceso al bus de Datos, este buffer debe presentar un estado de alta impedancia cuando no se selecciona ningún periférico o cuando se realiza una operación de escritura y debe permitir el paso de información desde el periférico seleccionado en las operaciones de lectura.

Se utilizarán los siguientes métodos de calificación:

- Pruebas escritas donde se verificará la asimilación de conocimiento.
- Sustentación oral de procesos de diseño e Implementación.
- Evaluación del avance del proceso de Concepción, Diseño e Implementación del Proyecto Final.

1.3.8. Actividades

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.

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.

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 (9) de alto nivel (9), modelarla (2) a nivel de sistema(6), verificar el cumplimiento de los requerimientos(33,38). Proporciona métodos para determinar su arquitectura óptima (particionamiento HW/SW) y definir la función e interacción(37,7) de sus componentes software (36) y hardware (35).

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. 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, por lo que se deben tener la siguiente estructura(17):

- Un diagrama de caja negra que indique las entradas y salidas del sistema.
- Una descripción de alto nivel del algoritmo que implementa 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)

Proyecto Final 1,2,3, 14, 15, 30, 31, 32, 33, 34, 35, 22, 23, 24, 25, 27

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.

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 uno de estos grupos (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 electro-magnética, 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.

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

1.3.9. Plataforma de Desarrollo SIE

La plataforma SAKC fué diseñada para ser utilizada como herramienta de implementación para los tres cursos de esta área. Está compuesta (ver Figura 1.11) por una FPGA y un Procesador, lo que permite la implementación de tareas Hardware y Software utilizando únicamente la FPGA o el procesador y la FPGA.

Durante el primer curso, el procesador de la plataforma se utilizará para configurar a la FPGA con las tareas hardware sintetizadas por los estudiantes. esta plataforma carece (intencionalmente) de elementos comunmente utilizados en las plataformas de desarrollo para PLDs como Pulsadores, Leds, Displays o conectores espaciales, lo que obliga a los estudiantes a investigar la forma de conexión de estos y a construir circuitos que permitan la conexión de la plataforma con estos elementos de interfaz con el usuario o con otros sistemas.

La capacidad de la FPGA de SAKC permite la implementación de procesadores soft-core (como plasma y microblaze), posibilitando la implementación de tareas Hardware y software en ella. El procesador será utilizado al finalizar el curso para comparar el desempeño (velocidad, consumo de potencia, costo) entre un procesador soft-core y un procesador comercial.

El ultimo curso utilizará el procesador para la ejecución de tareas software y la FPGA para ejecutar las tareas Software, se hará uso de herramientas utilizadas actualmente en el diseño de Aplicaciones comerciales de multinacionales como Nokia (QT), Motorola (Linux).

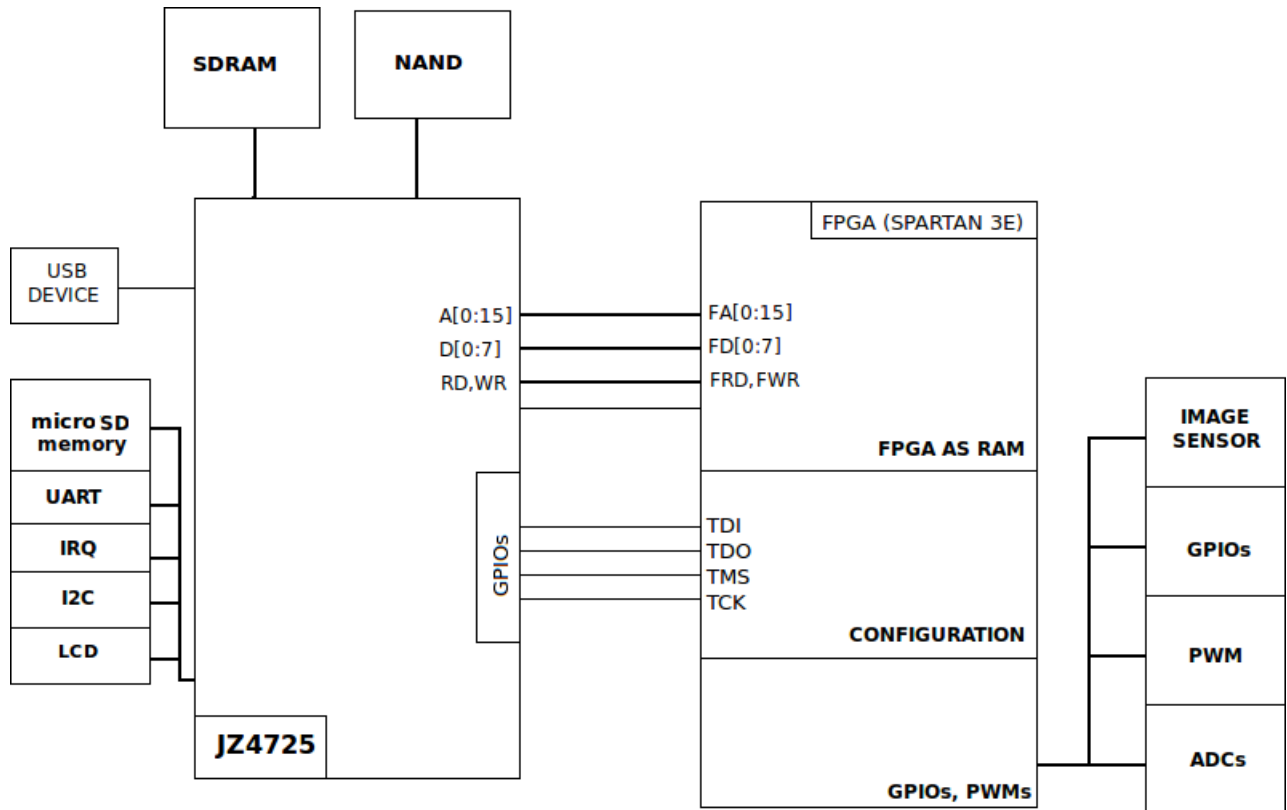


Figura 1.11: Diagrama de Bloques de la Plataforma de Desarrollo SAKC

Hardware y Software Copy-Left

El conocimiento debe ser considerado un bien común y se debe garantizar el acceso a todo el mundo. Por esta razón SAKC proporciona la documentación necesaria para:

- Estudiar, entender, y reproducir o modificar su Arquitectura.
- Conocer su proceso de fabricación.
- Entender su funcionamiento global y la interacción de sus componentes.
- Estudiar tutoriales que explican su programación.
- Descargar, estudiar y modificar el código fuente de todas las aplicaciones existentes actualmente.
- Realizar consultas con los creadores de las aplicaciones y de la plataforma de desarrollo.
- Contribuir a mejorar la calidad de la documentación y crear nueva información.

SAKC está distribuido bajo la licencia Creative Commons (CC) BY - SA, 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.

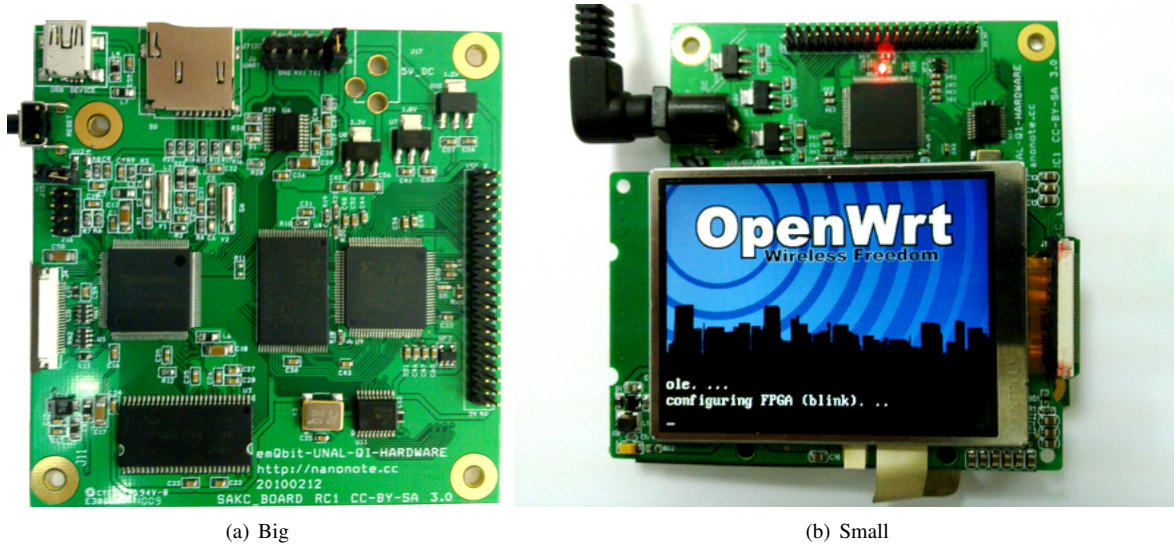


Figura 1.12: Plataforma de Desarrollo SAKC

1.4. Desarrollo de Métodos de Evaluación

Bibliografía

- [1] Worldwide CDIO Initiative. "Benefits of CDIO"URL:<http://www.cdio.org/benefits-cdio> on November, 2009.
- [2] E. Crawley, J. Malmqvist, D. Brodeur, and B. Lucas. CDIO Syllabus, Leadership and Entrepreneurship. *5th International CDIO conference*, 2009.
- [3] Edward F. Crawley. The CDIO Syllabus A Statement of Goals for Undergraduate Engineering Education. URL:<http://www.cdio.org>, 2001.
- [4] 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.