

Duración del curso: 16 clases de 3hs, iniciando el 11 de enero y finalizando el 26 de abril.

Días y horarios de cursada: miércoles de 14 a 17hs.

Docentes a cargo: Ing. Federico Zacchigna, Lic. Miguel García, a través de la ACSE.

Ubicación: SUR Emprendimientos Tecnológicos S.R.L., Rivadavia 611 Piso 11 CABA.



Curso de Procesamiento Digital de Señales e Imágenes en FPGA

El objetivo principal del curso es introducir a los participantes en el procesamiento digital de señales utilizando tecnología FPGA y técnicas de co-diseño hardware/software. El curso pondrá especial foco en el procesamiento de imágenes en tiempo real para aplicaciones de detección y seguimiento de objetos en imágenes. Se dictará el curso en dos partes.

Durante la “**Parte 1: Diseño digital en FPGA**” se tratarán conceptos básicos del diseño de sistemas de procesamiento digital en FPGA utilizando VHDL. En esta parte del curso los asistentes aprenderán a diseñar un sistema de procesamiento digital básico, diseñar un banco de simulación para verificar el sistema creado, implementar el sistema en FPGA, y diseñar el banco de ensayos necesario para caracterizar y validar dicho sistema.

Durante la “**Parte 2: Co-Diseño Hardware/Software**” se introducirá a los asistentes en la utilización de técnicas de co-diseño hardware/software en el desarrollo de sistemas de procesamiento digital. En esta parte del curso los asistentes aprenderán a diseñar un sistema utilizando técnicas que logran un compromiso adecuado entre las ventajas e inconvenientes de implementar un sistema sólo en hardware (es decir, utilizar un circuito microelectrónico específicamente construido para una aplicación) e implementarlo sólo en software (es decir, emplear componentes estándares programando la aplicación requerida en software). Los asistentes abordarán el diseño, implementación y validación de un sistema complejo de procesamiento digital de imágenes, que incluirá la programación de un microprocesador, la integración del sistema implementado en la primera parte del curso, y la integración de componentes externos de lógica programable.



PROYECTO INTEGRADOR

A lo largo del curso se abordará el desarrollo de un proyecto integrador, que tendrá por objetivo lograr el diseño, implementación y validación de un sistema de filtrado de imágenes.

En la primera parte del curso se implementará el bloque principal de un sistema de filtrado de imágenes en hardware, que calculará la convolución en dos dimensiones, para la detección de bordes de objetos presentes en las imágenes.

Mientras que en la segunda parte del curso se abordará la implementación de un sistema de filtrado de imágenes en software, que luego será optimizado delegando parte del trabajo al componente desarrollado en la primera parte del curso.

PRE REQUISITOS

Los participantes deberán estar familiarizados con los conceptos básicos de las técnicas digitales, tales como la representación binaria, compuertas, etc.

Se espera que tengan conocimientos básicos de técnicas de procesamiento de imágenes tales como segmentación de imágenes, detección de bordes, detección de puntos característicos, etc.

También será necesario que sean capaces de programar software embebido en microprocesadores, fundamentalmente para la segunda parte del curso.

HERRAMIENTAS DE DESARROLLO

Cada participante deberá contar con una computadora personal para el trabajo en el curso. En caso de no contar con ello deberá avisar a los organizadores, previo al inicio del curso.

Las herramientas para desarrollo, simulación y depuración serán facilitadas en el curso, así como los kits de desarrollo en FPGA (para uso exclusivo en las instalaciones de SUR).

Ejemplos de filtros de detección de bordes



Figura 1 - Imagen de Lena



Figura 2 - Lena con filtro tipo A



Figura 3- Lena con filtro tipo B

CRONOGRAMA DETALLADO

Parte 1: Diseño digital en FPGA

CLASE	CONTENIDOS
Clase 1: Introducción a los sistemas digitales y el diseño digital	<ul style="list-style-type: none"> • Representación binaria, punto fijo y punto flotante. Aritmética en los sistemas de representación. • Elementos del diseño digital: <ul style="list-style-type: none"> ○ Circuitos combinacionales: genéricos (funciones lógicas), multiplexores, comparadores, sumadores. ○ Circuitos secuenciales: flip-flop, registro, memorias. ○ Complejos: Multiplicadores, FIFO, Máquinas de estado (FSM). • Diagramas temporales de circuitos combinacionales y secuenciales. • Ejemplos combinando elementos básicos para formar detectores de flanco, contadores, y otros elementos complejos.
Clase 2: Introducción a VHDL y simulación de circuitos combinacionales	<ul style="list-style-type: none"> • Repaso rápido de los elementos de diseño. • Introducción a VHDL. Práctica: <ul style="list-style-type: none"> • Diseño del primer elemento y su banco de pruebas para la simulación. • VHDL mediante ejemplos. Diseño concurrente de otros elementos combinacionales básicos: funciones lógicas, multiplexores. • Diseño de elementos combinacionales mediante process. • Circuitos de ancho de palabra genéricos.
Clase 3: Introducción a FPGA y simulación de circuitos secuenciales	<ul style="list-style-type: none"> • Circuitos sincrónicos y RTL. • Descripción de la estructura interna de una FPGA. Práctica: <ul style="list-style-type: none"> • Diseño de circuitos secuenciales: flip-flop, registro, registro de N bits. Luego, sumadores y contadores. • Utilización de generic para la implementación de un registro-N y una FIFO-N. Más ejemplos.
Clase 4: DSP. Implementación en FPGA	<ul style="list-style-type: none"> • Arquitecturas de procesamiento de datos digitales. Interfaz de <i>streaming</i> de datos, <i>delay</i> y <i>throughput</i> del sistema. • Arquitecturas pipeline e iterativas. Práctica: <ul style="list-style-type: none"> • Implementación del primer circuito en FPGA: Contador con enable y reset sincrónico. • Implementación del multiplicador y multiplicador pipeline.
Clase 5: Presentación del trabajo final	<ul style="list-style-type: none"> • Filtrado de imágenes mediante filtro de convolución. Su aplicación para detección de bordes. Práctica: <ul style="list-style-type: none"> • FIFO en RAM de largo genérico. • Diagrama en bloque del trabajo final. • Inicio del desarrollo del trabajo final.
Clase 6, 7 y 8: Trabajo final	<ul style="list-style-type: none"> • Desarrollo y presentación del trabajo final.

Parte 2: Co-Diseño Hardware / Software

CLASE	CONTENIDOS
Clase 1: Introducción	<ul style="list-style-type: none"> Introducción al co-diseño hardware/software y metodología de desarrollo. Arquitectura de los All Programmable Systems on a Chip. Herramientas: Vivado IP Integrator y Vivado SDK. Práctica: <ul style="list-style-type: none"> "Hola mundo", totalmente en software. Debug de software desde Vivado SDK.
Clase 2: Comunicación PS-PL	<ul style="list-style-type: none"> Introducción a la comunicación del procesador con componentes en lógica programable: bus AXI, espacios de direcciones. Práctica: <ul style="list-style-type: none"> Lectura de switches o botones y salida en leds, utilizando IP Core GPIO: Mapeo de pines, consumo de recursos.
Clase 3: DMA	<ul style="list-style-type: none"> AXI High performance. IP Core AXI Data mover: su interfaz hacia RAM y hacia streams. Protocolo AXI Stream. Interrupciones: conexiones hardware y su atención en software. Práctica: <ul style="list-style-type: none"> Copias de RAM a RAM usando DMA. Chequeo desde PC vía UART.
Clase 4: Integración del componente filtrador de imágenes	<ul style="list-style-type: none"> Conexión del componente al DMA. Driver para control del componente. Práctica: <ul style="list-style-type: none"> Prueba con imágenes provistas por UART. Medición de performance del componente usando timers.
Clase 5: Captura y reproducción de imágenes	<ul style="list-style-type: none"> Formato de imagen en memoria. Entrada HDMI: presentación de los bloques necesarios. Práctica: <ul style="list-style-type: none"> Implementación de un sistema que capture y procese video vía HDMI.
Clase 6, 7 y 8: Trabajo final	<ul style="list-style-type: none"> Presentación del trabajo final. Implementación y optimización en software. Estimación de rendimiento. Optimización mediante delegación del trabajo al componente de filtrado implementado en Parte 1 del curso. Práctica: <ul style="list-style-type: none"> Desarrollo y presentación del trabajo final.