



CARRERA: Ing. Electrónica

PROGRAMA ANALÍTICO DE ASIGNATURA

NOMBRE DE LA ACTIVIDAD CURRICULAR: TÉCNICAS DIGITALES II

Año Académico: 2024

Área: Técnicas Digitales

Bloque: Tecnologías Básicas

Nivel: 4

Tipo: obligatoria

Modalidad: anual

Cargas horarias totales:

<i>Horas reloj</i>	<i>Horas cátedra</i>	<i>Horas cátedra semanales</i>
120	160	5

COMPOSICIÓN DEL EQUIPO DE CÁTEDRA:

Profesor Titular: Ing. Alejandro Furfaro

JTP: Dr. Ing. Luciano Ferreyro

FUNDAMENTACIÓN

Esta asignatura con un adecuado balance de actividades teóricas y prácticas, introducirá al alumno en las tecnologías y metodologías actualizadas al estado del arte, de forma que esté en condiciones de proyectar, diseñar, calcular, simular, depurar y producir dispositivos electrónicos de procesamiento de señales analógicas y digitales y el hardware, firmware y software de sistemas embebidos y el firmware y software a él asociados de sistemas de cómputo de propósito o general basados en Microprocesadores, con el objetivo de destinarlos a las tecnologías de la información y las comunicaciones, a las necesidades de la industria, a la automatización y control de procesos y la robótica, a los sistemas informáticos y digitales, y, en general, a cualquier sector de actividad económica y de la sociedad que necesite aplicar sistemas de cómputo en la mejora de su productividad, competitividad, o de la sociedad en sí.

Para poder cumplirlo, el alumno desarrollará e implementará su hardware en base a una plataforma comercial SOC-FPGA estándar confiable y robusta sobre la cual



desarrollará el firmware desde cero (bare metal) para comprender los recursos de hardware que utiliza un sistema operativo, diseñar un dispositivo de hardware para entrada / salida en FPGA conectable al Core, y una aplicación real que se apoyará en las capas de firmware de interfaz con el hardware apoyándose en los conocimientos de programación y descripción de hardware previamente adquiridos en las asignaturas del área digital, y accediendo a los demás dispositivos de entrada salida a través de los drivers del Sistema Operativo Base.

Al finalizar el curso, el alumno tendrá la capacidad para diseñar soluciones sobre actuar de manera responsable y competente en cualquier escenario nacional e internacional desarrollando la capacidad de innovación, al servicio de un conocimiento productivo, generando empleos y posibilitando el desarrollo social.

COMPETENCIAS DE EGRESO ESPECÍFICAS A LAS QUE CONTRIBUYE:

Competencia	Competencias de Actividades reservadas			Competencias de Alcances
	Baja	Media	Alta	
CE 1.1 - Diseñar, proyectar y calcular sistemas, equipos y dispositivos de generación, transmisión y/o procesamiento de campos y señales analógicos y digitales; circuitos integrados; hardware de sistemas de cómputo de propósito general y/o específico y el software a él asociado; hardware y software de sistemas embebidos y dispositivos lógicos programables; sistemas de automatización y control; sistemas de procesamiento y de comunicación de datos y sistemas irradiantes, para brindar soluciones óptimas de acuerdo a las condiciones técnicas, legales, económicas, humanas y ambientales.			X	
CE 1.2 - Plantear, interpretar, modelar y resolver los problemas de ingeniería descritos en CE 1.1.			X	
CE 1.4 - Diseñar, proyectar y calcular circuitos y sistemas digitales.			X	

COMPETENCIAS DE EGRESO GENÉRICAS A LAS QUE CONTRIBUYE:

Competencia	Baja	Media	Alta
Identificar, Formular y resolver problemas de Ingeniería Electrónica.		X	
Concebir, diseñar y desarrollar proyectos de Ingeniería Electrónica.			X



Utilizar de manera efectiva las técnicas y herramientas de aplicación de la ingeniería electrónica.			X
Desempeñarse de manera efectiva en equipos de trabajo.		X	

OBJETIVOS (APRENDIZAJE/LOGROS A ALCANZAR)

- Diseñar hardware de sistemas embebidos y sus interfaces con el mundo real, para un rango de aplicaciones amplio.
- Desarrollar firmware para sistemas embebidos, que permitan comprender las bases de funcionamiento de un Sistema Operativo de Tiempo Real (RTOS), y su relación con los recursos de soporte de la CPU.
- Desarrollar proyectos de software que utilicen los recursos disponibles a través de un Sistema Operativo, para acceder al hardware del Core, y sus requerimientos particulares.
- Introducir las tecnologías y metodologías de forma de poder seleccionar componentes, herramientas y técnicas para el desarrollo de sistemas embebidos.
- Proyectar, calcular, simular y depurar dispositivos electrónicos de procesamiento de señales analógicas y digitales y el hardware, firmware y software de sistemas embebidos y el firmware y software a él asociados de sistemas de cómputo.

CONTENIDOS

Contenidos mínimos

- Conversores analógicos/Digitales y Digitales/ Analógicos.
- Memorias Electrónicas.
- Arquitectura y organización de procesadores.
- Hardware y software de sistemas embebidos.
- Comunicaciones y protocolos digitales seriales de baja y alta velocidad.
- Introducción a los sistemas operativos en tiempo real.

Contenidos analíticos

Unidad 1: Introducción a la Arquitectura y Organización de Computadores

Introducción a la Arquitectura y Organización de procesadores. Diferentes generaciones de procesadores. Modelo de programa almacenado.

Modelos Von Neumann y Harvard.

Organización y Hardware. Pipeline: Funcionamiento, riesgos y Obstáculos.

Procesadores CISC. Paradigma RISC: Sus postulados. Análisis de los procesadores



actuales con esta perspectiva: Intel, Motorola, ARM y RISC V. Primeros programas en assembler.

Ejemplos homogéneos para comparación de código entre diferentes plataformas: ARM e Intel. Análisis crítico de cada caso. Cantidad de instrucciones versus cantidad de micro operaciones. Tamaño del código resultante.

Unidad 2: Microprocesadores de 32 bits.

Arquitectura ARMv7.

Orígenes de ARM. Procesador ARM7TDMI: Su influencia en la industria y posterior evolución. Pipeline de tres etapas. Línea Cortex. Perfiles A, R, y M: Características generales de c/u. Limitaciones y diferencias del perfil Cortex-M, respecto de los anteriores. Análisis crítico desde el punto de vista de lo aprendido: ¿Una arquitectura homogénea? ¿o tres sub arquitecturas?. Notas de Time to market vs. Rigurosidad técnica.

Generalidades de Arquitectura y Organización: La Pila. Funcionamiento. Registros asociados con el manejo de la pila. Pila en ARMv7. Representación de números. Endianess. Su implementación en ARM.

Perfil Cortex-A: Descripción detallada. Registros de Propósito general. Registros dedicados: Stack Pointer, Link Register, Program Counter. Modos de funcionamiento. Estados del set de instrucciones.

Set de Instrucciones ARM32, y T32.

Instrucciones de ejecución condicional. Ventajas respecto de la integridad del pipeline. Instrucciones de Acceso a memoria Load - Store, y Load Store Múltiple. Manejo de Constantes, y referencias a direcciones inmediatas de 32 bits. Resolución por literal pools. Loops y Branches.

Ejemplos de programación:

Estructura de un programa en Lenguaje ensamblador. Uso de GNU Assembler.

ABI ARM: Pasaje de argumentos a funciones, por registros y por pila. Set de instrucciones: Conceptos preliminares. Uso de salida de archivos listing en el GNU assembler. Análisis de su información. Análisis de un literal pool, generado por el ensamblador mediante el archivo listing.

Arranque de un Cortex A9. Arranque de una Zynq-7000. Donde se ubica el código de inicialización de nuestro sistema.

Análisis de los formatos de los objetos generados por el assembler y el linker: ELF. Análisis del formato con comandos de consola: readelf, objdump. Análisis de etiquetas de código y datos.Linker. Introducción a los linker scripts. Relación con las etiquetas del programa y su uso en el linker script.

Debugging. gdb como core de debugging. Dashboard: Herramienta para su mejor visualización. Uso de estas herramientas para depuración de programas.

Interrupciones. Definiciones generales. Tipos de interrupción y sus características mas distintivas. Identificación de las diferentes fuentes de interrupción. Proceso de general de Vectorización.

Sistema de E/S: Acceso a los dispositivos periféricos. Técnicas de mapeo. Espacio de direccionamiento exclusivo para E/S vs. Dispositivos Memory mapped.



Interrupciones de Hardware. Necesidad de un Controlador de interrupciones. Prioridades. Máscaras. Disparo por flanco o por nivel. Diferencias fundamentales entre el sistema de interrupciones de Hardware del perfil M y el de los perfiles A y R.

Características del sistema de interrupciones del Cortex-M. El NVIC.

Características del sistema de interrupciones de los perfiles Cortex-A y Cortex-R. El GIC. Diagrama en bloques. Identificación de interrupciones: Interrupt ID. SGI: Software Generated Interrupts, PPI: Private Peripheral Interrupts, y SPI: Shared Peripheral Interrupts. Señalización de interrupciones. Estados de una interrupción. Mecanismo de procesamiento de interrupciones. Configuración. Inicialización. Registros Memory Mapped.

Implementaciones propietarias a criterio de los fabricantes. Caso Sitara 3358

Gestión de interrupciones IRQ y FIQ. Handler de Interrupción Simple y Anidamiento. Recaudos a tomar con los registros banqueados en atención anidada. Resguardo del stack y del estado de máquina.

Unidad 3: Soporte para un Sistema Operativo: Recursos para Context Switch

Concepto de Tarea como unidad de ejecución. Recursos de hardware para soporte de ésta función. Concepto de Contexto, Espacio de contexto, y Context switch. Estados de tareas.

Primeros conceptos de System Programming: Planificación de Tareas, Espacios de memoria para tarea y de administración.

Planificación de tareas: Estructuras de control genéricas. Políticas de scheduling. Restricciones temporales y como resolverlas.

Implementación general de un scheduler (planificador) para ARM Cortex-A. Resguardo de los registros core en el stack. Resguardo de los registros FPU y SIMD: concepto de Lazy FPU context switch. Soporte de Hardware para su implementación.

Unidad 4: Soporte para un Sistema Operativo: Recursos para System Programming

Generalidades para el desarrollo de Embedded Systems. Conceptos fundamentales de System Programming. Arranque de un SoC. El Firmware. Inicialización del Hardware del sistema, funciones de acceso al Hardware transparentes al usuario: Hardware Abstraction Layer (HAL).

Modelo de System Programming para ARMv7 perfiles A y R. Descripción de los Modos de ejecución, estados de ejecución y Niveles de privilegio. Recursos de arquitectura para su control y configuración.

Manejo de excepciones en Modo Privilegiado. Resignificación del Banqueo de Registros en los diferentes modos de ejecución.

Coprocesadores. Modelo general de ARM basado en Coprocesadores. Los Coprocesadores CP14 y CP15. Funciones, modos de acceso. Instrucciones para acceso a los Coprocesadores. Formato y técnicas generales de acceso. Guía de supervivencia para entender el críptico manual de System Programming de ARM.

Aplicación del Modelo de Programación de Sistemas. Sistemas Operativos para Embedded Systems. Evolución desde el modelo de Lazo de control, Sistemas Event Driven, Sistemas Interrupt Driven, Eventos sincrónicos y Asíncrónicos, Priorización de eventos. Modelo de procesos. Sistemas Mono y Multiprocesador (Simétricos y Asíncrónicos), espacio de



direccionamiento Real o Virtual, Gestión estática o dinámica, Preemption. Sistemas derivados de estos modelos: Real Time Systems

Unidad 5 : Sistemas Operativos para Embedded Systems

Funciones de un Sistema Operativo. Arquitectura de un Sistema Operativo End to End. Rango de Sistemas Operativos.

Scheduling de Procesos. IO-Bound vs. CPU-Bound, Tiempo de respuesta. Throughput.

Algoritmos de scheduling: Round Robin, Dynamic Priority, Preemption. Scheduler Time sharing vs Real Time. Determinismo y Latency.

Sincronización de Procesos. Técnicas: Wake-up. Semáforos: Operaciones, Semáforos Binarios, Mutexes. Problema Productor-Consumidor. Problema Escritor-Lector. Starvation. Dead Locks, características, estrategias de prevención. Intercomunicación de procesos de alto nivel: colas, pipes, etc.

Gestión de memoria en ARMv7: VMSA y PMSA. Privilegios.

Sistemas Operativos de propósito General: Una somera introducción para el siguiente curso.

Sistemas Operativos de Tiempo Real. Características imprescindibles. Principales diferencias con los de propósito general.

Algoritmos de Scheduling clásicos para RTOS: RDS y EDF. Descripción detallada de ambos algoritmos. Diagramas temporales. Características y limitaciones de c/u.

Características y escenarios complejos para RTOS: Inversión de prioridades y su prevención, Deadlines, Prioridad Techo, Herencia de prioridad.

Casos de estudio de RTOS's: Free-RTOS, MicroC-OS, NuttX, QNX, RT-Linux. En cada caso se repasan arquitectura, características, rangos de aplicación, apego a estándares abiertos (con sus innegables ventajas).

Diseño de un RTOS: Lineamientos fundamentales: Gestión de tareas, de memoria, políticas de scheduling, carácter estático o dinámico. Diseño en base a restricciones temporales o de performance.

Unidad 6: Buses de Inter conexión con la CPU.

Principios de diseño de un Bus. Introducción general Especificación del Bus AMBAv2: Alta performance (AHB), System (ASB), y Periféricos (APB). Diagrama general y características de cada Bus. Conceptos de Ciclo de Bus, Transferencia de Bus y Operación de Bus. AHB: Master, Slave y Arbitro. APB, Bridge con el ASB. Características.

Organización del Bus. Dispositivos que se conectan del lado AHB, y APB. Señales AHB y APB.

Arquitectura detallada. Operación AHB. Protocolo Request Grant. Acceso Directo a Memoria, como mecanismo de acceso directo a un Bus sin intervención de una CPU de propósito general. Ciclos de address y ciclos de control. Transferencias en modo Burst. Wait States. Control de errores.

Transferencias AHB. Diagramas de timing para su operación. Transferencias Múltiples, Bust, interrupción de transferencias Burst. Busr Wrapping. Burst Incremental. Burst de longitud desconocida. Mención a la interfaz DRAM LLC que se analizará en el curso siguiente (TDIII)

Señales de control.

Decodificación de Direcciones.



Respuesta desde el slave a las diferentes transacciones: Retry, Error, Split and Retry. Diagramas temporales para cada una.

Armado de las líneas de datos: Endianess.

Arbitraje del AHB. Diagrama de señales detallado. Diagramas de timing para : Bus Grant, Bus Grant con Wait State, Hand Over, Hand Over sobre un Burst, Manejo de HGrant, Split.

Evolución en AMBAv3: AXI. Características nuevas y compatibilidad con AMBAv2.

Unidad 7: Memorias.

El sistema de Memoria. Rol de la Memoria en la Performance de un computador. Capacidad, ancho de banda de bus, tiempo de acceso. Clasificación: Volátiles y No volátiles.

Jerarquía. Localidad. Performance. Energía y Potencia. Costos. Casos prácticos de memorias jerárquicas.

Estudio detallado de energía y potencia. Corrientes de fuga, energía y potencia estática. Estudio de la energía y potencia en conmutación. Estrategias tecnológicas para reducir la potencia y la energía.

Memorias volátiles. Memorias dinámicas. Visión introductoria. Refresco. Características salientes.

Memorias estáticas. Análisis detallado. Comparación de parámetros con respecto a la memoria dinámica. Breve introducción a la memoria cache.

Funcionamiento de una celda de SRAM. Decodificación interna. Decodificación de direcciones. Organizaciones: Predecoding, No particionado, División de wordline, Decodificación de wordline jerárquica.

Circuitos periféricos. Circuitos de pre carga y Ecuilibración: Diferentes configuraciones. Multiplexado de lectura/Escritura, Amplificadores de Detección: Diferentes configuraciones. Estructuras jerárquicas de detección. Amplificadores de escritura.

Timing y conexión física. Conexión sincrónica. Acceso burst. Casos prácticos.

Potencia y Energía. Dynamic Voltage Scaling. Powering Down. Gating Vdd. Potencia de leakage en SRAM.

Memorias no volátiles. Evolución desde la ROM grabada por el fabricante del chip. Memorias Flash. Tecnologías NAND Flash y NOR Flash. Características de ambas tecnologías y aplicación en cada caso en función de sus diferencias. Operación de escritura y borrado.

Funcionamiento del dispositivo FGPMOS como memoria de almacenamiento no volátil.

Lectura de una Memoria Flash. Estructura circuital de una Flash NAND y de una Flash NOR. Implementación en Array de celdas.

Acceso por bus paralelo o serie (SPI). Estudio de modelos reales.

Organización interna y configuración. Registro de estados. JEDEC Common Flash Interface.

Unidad 8: Interfaz serie de baja velocidad

Interfaces de un computador: Paralelo y Serie. Interfaces serie de baja velocidad. RS-232, RS-422, y RS-485, Breve reseña de sus características principales.

SPI: Diagrama funcional y características. Configuración Master Multi-Slave: Descripción funcional y diagramas de temporización. Señalización CLCK, SS, MISO y MOSI. Timing de operación Master Slave. Clock Sincrónico: Polaridad y Fase. Implementación SPI en BBB: SITARA 3358, y en Zync-7000.



I2C: Diagrama Funcional y características. Formato del mensaje. Direccionamiento. Read/Write Bit. Data Frame: Trama e una comunicación. Análisis temporal. Configuración Single Maater Multi Slave. Configuración Multi Master Multi Slave. Sensado de la línea SDA. Multimaster: Sincronización de clock, Arbitración del bus. Sumario de ventajas y desventajas. Implementación I2C en Zynq 7000.

CAN: Características, formato de frame. Implementación CAN en Zynq 7000.

Estándares adicionales. 1-Wire: Características principales, Topología de bus. Timing de Bus. Ejemplos.

Unidad 9: Interfaz serie de alta velocidad USB.

Descripción del Bus USB. Especificaciones generales. Genealogía 1.0, 1.1, 2.0, 3.0, 3.1, 3.2.

Características mecánicas. Conectores. Características Eléctricas. Ventajas de la señal diferencial. Niveles de Tensión y corriente característicos. Detección de la conexión de un dispositivo al bus. Topología: Dispositivo Hub Root, Hubs, y Nodos y sus funciones. Codificación de la información. Modelo de Arquitectura en capas. Evolución. Host, device y On-the-go. Transferencias. Endpoint, pipes, tipos de transferencias. Inicio de una transferencia. Bloques constitutivos de la misma. Fases de la transferencia. Handshake y procesamiento de errores. Conexión de dispositivos y su detección.

Transferencias de control, Burst, por interrupción e isócronas. Tramas y división de tiempos. Demoras (latency) y requerimientos al host.

El proceso de enumeración. Incorporación y remoción de un dispositivo. Descriptores de dispositivo, de configuración, de interfaz, de string, de endpoint.

Clases de dispositivos, principales características. La clase de los dispositivos de interfaz con el ser humano (HID). Su caracterización. Configuración de un controlador para ser interpretado por un sistema operativo.

Enumeración. Análisis detallado de la secuencia completa. Análisis de un Bus físico real.

Unidad 10: Conversión Analógica a Digital y Digital a Analógica.

Breve reseña del análisis de Fourier. Representación de señales en el dominio de la frecuencia. Digitalización de una señal. Muestreo Cuantificación Codificación. Circuito de Muestreo y Retención. Análisis funcional. Determinación del error de muestreo. Funciones de un Conversor AD: Cuantificación y Codificación: Error de Cuantificación en LSB.

Implementación de un conversor AD: Diagrama general. Función transferencia. Implementación de un Sistema general de Adquisición de Datos. Filtrado analógico de entrada y digital de salida.

Conversores de Alta Velocidad: Flash Converter: Diagrama General Código Termómetro. Problemas de diseño: Timing y Distorsión.

Conversores de Alta Precisión. Simple y Doble Rampa. Breve descripción. Características fundamentales, fortalezas, y limitaciones tecnológicas.

Conversores de Aproximaciones Sucesivas. Diagrama. Registro SAR. Funcionamiento. Características.

Conversores Sigma-Delta. Diagrama funcional Modulador Sigma Delta. Funcionamiento. Principios teóricos que lo rigen Noise Shaping. Función del Filtro Digital decimador.

Fundamentos de un sistema de Muestreo de datos; Relación Señal a Ruido en la entrada de un conversor. Performance Dinámica de un ADC de N-bit. Ruido en ADCs. Performance dinámica



de un Conversor. Distorsión Armónica, Peor Armónica, Distorsión Armónica Total (THD), Distorsión Armónica Total mas Ruido (THD+N). Signal-to-Noise-and-Distortion Ratio (SINAD), Signal-to-Noise Ratio (SNR), y Effective Number of Bits (ENOB).

Unidad 11: Introducción al Diseño de Circuitos Impresos.

Elementos de diseño de Circuitos Impresos. Ruteo de señales analógicas, digitales y alimentación. Conceptos de Integridad de señal.

Sustratos. Diferentes Tipos. Criterios de selección. Características físicas principales. Catálogos normalizados de materiales.

Gestión de Temperatura. Ensayos.Tooling Holes. Lado Primario y Lado secundario.

Layout. Capas. Redes de Alimentación. Layout óptimo. Capacitores de descople. Distribución de circuitos según frecuencia. Dimensionamiento de los conductores. Designadores. Sistemas de Grilla. Leyendas. Componentes y Montaje. Distancias entre componentes. Assembly Drawing. Ubicación de componentes fiduciales. Ubicación de puentes y jumpers. Panelización. Tipos de orificios y sus tolerancias. Vías. Edge conectors. Máscara antisoldante y coating.

Documentación.

Herramientas de Software en el Diseño de Circuitos Impresos

DISTRIBUCIÓN DE CARGA HORARIA ENTRE ACTIVIDADES TEÓRICAS Y PRÁCTICAS

Modalidad organizativa de las clases	Horas Reloj totales presenciales	Horas reloj virtuales totales	Horas totales
Teórica	48		48
Formación práctica	72		72

Tipo de prácticas	Horas Reloj totales presenciales	Horas reloj virtuales	Lugar donde se desarrolla la práctica
Formación experimental	24		Laboratorios 105, 109 y 110 del Dpto. Ing. Electrónica UTN.BA Sede Medrano
Problemas abiertos de Ingeniería (ABP)			
Proyecto y diseño	48		Laboratorios 105, 109 y 110 del Dpto. Ing. Electrónica UTN.BA Sede Medrano
Otras:			
Práctica supervisada			



Total de horas	72		
-----------------------	-----------	--	--

ESTRATEGIAS DE ENSEÑANZA Y ACTIVIDADES DE APRENDIZAJE

Se procura transmitir a los alumnos una impronta general del estado del arte de la actividad profesional, tanto en ámbitos empresariales como científicos. Se trata de un abordaje general, evitando la impronta personal del docente, ya que se pretende una mirada sin sesgo personal. El esfuerzo es entender la industria de sistemas de cómputos mas allá de los muros de nuestra propia actividad y de las fronteras de la Argentina, proporcionando al estudiante una formación que lo prepare para superar las limitaciones evidentes de la industria local, y convertirlo en un agente de cambio capaz de impulsar el desarrollo que el país requiere.

Por otra parte se procura proporcionar capacidad para trabajo por proyectos desarrollados por equipos multidisciplinarios conformados por especialistas en las diferentes áreas disciplinares en virtud de la complejidad tecnológica que cada área ha alcanzado de la mano de la Investigación científica y el desarrollo tecnológico consecuente. Se considera importante que la asignatura pueda desarrollar esta competencias en los estudiantes a través de proyectos y aplicaciones que eventualmente puedan propender a la Integración Horizontal de contenidos con el resto de las asignaturas del 4to. Nivel del plan de estudios, brindándoles una visión mayor de las perspectivas de aplicación de las tecnologías digitales de alta prestación en otras áreas de la ingeniería Electrónica cuyos contenidos son compatibles con ésta asignatura para la implementación de proyectos de espectro mas amplio: como así también con las asignaturas relacionadas de los niveles previos en base a un proyecto de Área consistente que pueda hilar competencias y conocimientos a lo largo de la formación a medida que el estudiante transita por los diferentes niveles de la carrera.

Soporte tecnológico

Las consultas y comunicaciones se pueden efectuar en cualquier momento de cualquier día de la semana. Para ello utilizamos un server en Discord que nos permite a los docentes responder consultas y resolver dudas de manera no presencial. De este modo un estudiante puede consultar a los docentes cualquier problema que lo estanca en el avance de su trabajo.

Combinado con Discord utilizamos el servidor gitlab de la Facultad, para que los estudiantes entreguen sus trabajos. Además se podrán subir versiones con problemas para que los docentes podamos verificar los reportes de problemas y consultas en Discord y podamos orientarlos en su resolución entre clases, evitándoles esperar a la clase siguiente para resolver sus dudas.

Teniendo en cuenta el requisito de evaluación continua para la promoción, establecido en el nuevo reglamento de estudios de la UTN implementado en el ciclo lectivo 2017, se implementa un sistema de evaluación continúa. Se divide la asignatura en dos



instancias de evaluación, una por cada cuatrimestre. Las dos se basarán en la metodología de entrega individual de un Trabajo Práctico integrador que ha sido desarrollado en etapas incrementales en las cuales los docentes pueden evaluar a cada estudiante de manera continua. Al final del cuatrimestre el Trabajo Práctico final será la resultante de dicho desarrollo incremental y debe estar en condiciones de funcionamiento en el repositorio git. Por cada entrega parcial durante el cuatrimestre se realiza un examen escrito que permita verificar que el estudiante comprende los conceptos teóricos detrás del código que termina de entregar.

Trabajos Prácticos

Los Trabajos Prácticos se diseñan como secuencias incrementales de funcionalidades. Se parte de un esquema minimalista de arranque al que en cada paso se le agregan funciones y prestaciones. El trabajo práctico final de cada secuencia o trayecto formativo será la resultante de un proceso de desarrollo gradual en el que cada paso significa aplicar en la práctica un nuevo concepto teórico.

Los trabajos prácticos se entregan en el repositorio gitlab. La parte práctica debe ser desarrollada por el alumno en forma individual, parte del trabajo se ejecuta en clase en presencia de los docentes y otra parte fuera del horario de clase asistido por los docentes a demanda mediante el soporte tecnológico de discord y gitlab.

Se espera de esta forma que el Trabajo Práctico final de cada trayecto formativo obre como elemento de aprendizaje efectivo ya que requiere del alumno tiempo extra al de clase (es una estrategia que le exige el necesario tiempo de aplicación y estudio sin el cual es imposible aspirar a aprobar los trabajos prácticos como mínimo o promocionar la asignatura como meta superior).

La guía de trabajos integradores incluyen los siguientes hitos de desarrollo:

Primer Cuatrimestre:

Trayecto 1 . Parte A

- Assembler ARMv7. Programas básicos con el fin de explorar el set de instrucciones y familiarizarse con la estructura de un programa escrito en Assembler puro.
- ABI ARM. Interfaz entre C y Assembler. Comparación de código y performance entre funciones escritas en Assembler y llamadas a la misma función escrita en C.

Trayecto 1 . Parte B



- Interrupciones Básicas - Luego del arranque comenzar a establecer el sistema de Interrupciones básico para dispositivos de Hardware y Excepciones. Notas sobre consumo mínimo en la espera de eventos: SWI
- Interrupciones de HW y Excepciones – Timer tick entre los dispositivos mas salientes que se requiere controlar. Concepto de base de tiempo e importancia frente al paso siguiente.

Trayecto 1 . Parte C

- Task Scheduler Simple. Implementar dos tareas triviales basadas en el código generado hasta el momento. El eje de esta parte del trabajo es escribir el context switch.
- Manejo de Modo Usuario y Modos Privilegiados. Problemática cuando las dos tareas se establecen en modo User.

Trayecto 2

- Instalación de kicad y experimentación con las principales funciones. Comenzar a comprender la diferencia entre una PCB de desarrollo y uno de producto final.
- Diseño de un PCB conectable a uno de los boards de desarrollo con algún sensor conectable por interfaz serie I2C o SPI, y una serie de electrónica de mejoramiento de la señal de entrada que se vaya a ingresar por el Conversor AD/DA.
- Modificcación de un board base (ejemplo clásico de Beagle Bone Black), con alguna adaptación para customizarlo a un requerimiento específico.

Segundo Cuatrimestre:

Trayecto 3

- Desarrollo de aplicaciones utilizando los periféricos estudiados en clase bajo la administración de un sistema operativo. Conexión y control de sensores. Aplicaciones de Adquisición de datos con el conversor AD del Zynq-7000. Uso del sysfs para acceder desde programas de modo user al hardware.

Trayecto 4

- Puesta en marcha de la FPGA y del SoC Zync-7000 bajo Linux.
- Diseño de un dispositivo customizado en Verilog con acceso por DMA y conexión al Bus mediante conectividad AXI provista dentro de la IP de Xilinx.



Trayecto 5

- Analizador de Protocolo USB. Uso de Wireshark. Detección de los paquetes mas singulares en la enumeración
- Uso de una tarjeta redpitaya para analizar buses serie de baja velocidad.

Evaluación

Por cada trayecto de formación práctica se evalúa además como mínimo un examen escrito.

De este modo se evalúa cada componente del trabajo práctico no solo desde su implementación sino desde la comprensión cabal de los conceptos teóricos que le dan al desarrollo de cada competencia el necesario sólido sustento del conocimiento, condición indispensable para un profesional de excelencia.

Al final de cada cuatrimestre las calificaciones surgen de la combinación ponderada de las evaluaciones de cada trayecto en sus aspecto práctico y teórico (examen escrito asociado).

En caso de obtener 8 o mas puntos en cada examen parcial el participante promocionará la asignatura.

Puede recuperar uno de los dos parciales en caso que éste no llegue a esa calificación. Nunca ambos. La nota definitiva será la del recuperatorio, ***aun si esta resultase menor a la calificación que se quiso recuperar. Ejemplo: Si se busca recuperar un 7 para convertirlo en 8 y promocionar y la nota del recuperatorio es menor de 7, queda la del recuperatorio como definitiva. Aun si desaprueba.***

ARTICULACIÓN HORIZONTAL Y VERTICAL CON OTRAS MATERIAS

Articulación con las materias del 4to. Nivel

El 4to. Nivel de la carrera se caracteriza por la cantidad de asignaturas que requieren de un proyecto anual como condición para regularizar o promocionar la asignatura.

En diferentes relevamientos realizados con estudiantes, hemos podido constatar las siguientes opiniones representativas de la mayor parte de los encuestados:

1. Los proyectos son interesantes y contribuyen a una mejor formación práctica y al ejercicio de la Ingeniería en condiciones asimilables a las de un entorno laboral.
2. El tener proyecto en todas las asignaturas del 4to. Nivel termina imprimiendo al segundo cuatrimestre del ciclo lectivo una carga de trabajo excesiva que conspira con el tiempo de estudio para los parciales, ya que los contenidos del segundo cuatrimestre resultan determinantes para poder iniciar el proyecto mientras que los del primer cuatrimestre son menos determinantes para al menos iniciar el proyecto.
3. Por lo general un sistema de cómputo termina siendo un componente esencial en cualquier proyecto que se realice en el resto de las asignaturas.



En base a estos datos concretos en lugar de cargar a los estudiantes con un proyecto adicional que contribuya a la saturación mencionada, les damos soporte a los que lo requieran o los orientamos a utilizar los trabajos de clase como bloques constitutivos de la solución de los demás proyectos.

En particular trabajamos con mas énfasis las interfaces de Adquisición de datos ya que las mismas pueden ser aplicadas en proyectos que involucren acondicionamiento de señales débiles a la entrada de un sistema (Electrónica Aplicada II), o pueden limitar en banda un espectro de señal de entrada para desarrollar filtros digitales (Teoría de Circuitos II), en donde además adelantamos eventualmente a demanda de los estudiantes que lo necesiten para su proyecto el modelo de Ejecución SIMD (NEON). En el caso de Sistemas de Comunicaciones no solo aportamos los temas mencionados anteriormente, sino el manejo de la FPGA utilizables en proyectos de SDR, y su interconexión con el SoC para algoritmos adicionales de procesamiento que enriquezcan el proyecto. En el caso de Medidas Electrónicas I, nos concentramos especialmente en las interfaces ya descritas a la entrada de los conversores AD.

Articulación con el Área

Uno de los principales objetivos de este programa de estudio y de los contenidos propuestos tanto para los tópicos teóricos como los recursos seleccionados para su implementación práctica es convertir a esta asignatura en un eslabón fundamental dentro del Área Técnicas Digitales.

En las asignaturas Informática I e Informática II se desarrollan competencias de programación con lenguajes estructurados (C), y orientados a Objetos (C++), con base en SoCs de gama media y baja, gestionados con Microcontroladores. Se trabaja con la Arquitectura ARMV7 perfil M (que en rigor en algunos modelos como el Cortex M0 es ARMv6).

No obstante los estudiantes comprenden como programar cualquier dispositivos mas o menos básico de Entrada Salida, y programan sus handlers de interrupción de manera fluida. Los proyectos de Informática II dan prueba sobrada de estas competencias.

Informática I está trabajando en una modificación en la implementación práctica de la primer parte de la asignatura que trabaja exclusivamente en lenguaje C, sobre la misma plataforma de microcontrolador que se utiliza en Informática II de modo de darle a esta segunda asignatura del área un mejor vuelo de aplicación.

Aun sin esta modificación que aun se trabaja a nivel conceptual en Informática I, no tiene sentido seguir trabajando con microcontroladores en la carrera, ya que los microprocesadores tienen un rango de aplicaciones muy superior y además una complejidad para su comprensión y aplicación que justifica ampliamente su abordaje a lo largo de dos cursos.

Por tal motivo Técnicas Digitales II introduce al estudiante en el primer curso de Microprocesadores de la carrera (De hecho el programa analítico se refiere a “Arquitectura y organización de procesadores” , textual).

Por otra parte en la asignatura Técnicas Digitales I de la cual Técnicas Digitales II es correlativa, se aborda el diseño de los bloques básicos de sistemas lógicos con el enfoque de Diseño RTL de modo de poderlos utilizar como bloques constitutivos de sistemas lógicos de complejidad creciente, y su descripción se efectúa a través de Lenguaje de Descripción de Hardware



(Actualmente VHDL, aunque se plantea en 2024 migrar a Verilog). La filosofía de construir sistemas mas complejos partiendo de los anteriores permite que en varios de sus cursos se llegue a diseñar un core sencillo de procesador, su interfaz con memoria y su pipeline de ejecución.

Todas estas competencias y conocimientos adquiridos con esfuerzo a lo largo de los tres niveles iniciales de la carrera deben ser honrados por Técnicas Digitales II.

Esta es la razón por la cual hemos desarrollado el plan de trabajos prácticos de la asignatura a través de un SoC FPGA.

Por otra parte en el quinto nivel espera Técnicas Digitales III que si no se proporcionan bases suficientes en Técnicas Digitales II se convierte en la única asignatura que aborde Microprocesadores con la complejidad que el tema tiene en un solo año. Esta situación la convierte en una materia muy extensa e intensa. Muy dificultosa y lo peor, se pierde una chance de poder darle mas tiempo a temas de gran importancia conceptual y de amplia aplicación práctica.

Este curso propuesto para Técnicas Digitales II se compone de un primer cuatrimestre en el que se estudia el procesador Cortex-A (en este caso A9), y todo el hardware que da soporte a los sistemas operativos multitarea, permitiendo llegar hasta un punto y luego continuarlo naturalmente en Técnicas Digitales III.

Al momento de estudiar la entrada salida, lo volcamos especialmente en la FPGA diseñando un dispositivo personalizado de E/S cuyo driver para Linux se desarrollará en Técnicas Digitales III.

Como puede verse esta asignatura esta concebida para integrarse plenamente con las asignaturas del mismo nivel pero mas aun fuertemente con las asignaturas del área.

Clase N°	Descripción	Modalidad
Clase 1:	Presentación de la asignatura, objetivos. Modalidad. Teórica: Material de soporte Audiovisual: Video Clase Teórica 1 Temario: Introducción a la Arquitectura y Organización de procesadores. Diferentes generaciones de procesadores. Modelo de programa almacenado. Modelos Von Newmann y Harvard. Organización y Hardware. Pipeline: Funcionamiento, riesgos y Obstáculos. Procesadores CISC. Paradigma RISC: Sus postulados. Análisis de los procesadores actuales con esta perspectiva: Intel, Motorola, ARM y RISC V. Primeros programas en assembler. Ejemplos homogéneos para comparación de código entre diferentes plataformas: ARM e Intel. Análisis crítico de cada caso.	Presencial
Clase 2:	Teórica: Material de soporte Audiovisual: Video Clases Teóricas 02 y 03	Presencial



	<p>Temario: ARM. Orígenes. ARM7TDMI: Su influencia en la industria y posterior evolución. Arquitectura ARMv7. Línea Cortex. Perfiles A, R, y M: Características generales de c/u. Limitaciones y diferencias del perfil Cortex-M, respecto de los anteriores. Análisis crítico: ¿Una arquitectura homogénea? ¿o tres sub arquitecturas?. Time to market vs. Rigurosidad técnica. Generalidades de Arquitectura y Organización: La Pila. Funcionamiento. Registros asociados con el manejo de la pila. Pila en ARMv7 Representación de números. Endianess. Sus implementación en ARM. Cortex-A: Descripción detallada. Registros, Modos de funcionamiento. Estados del set de instrucciones. Estructura de un programa en Lenguaje ensamblador. Uso de GNU Assembler. ABI ARM. Pasaje de argumentos a funciones, por registros y por pila. Set de instrucciones: Conceptos preliminares Instrucciones ARM32, Thumb, Thumb-2(T32).</p>	
Clase 3:	<p>Laboratorio. Taller de programación. Presentación del software de soporte QEMU para Zynq 7000. Toolchain de desarrollo. Crosscompiler. Primeros ejercicios de assembler. Objetivo: Comprender la estructura de un programa en assembler, recorrer las instrucciones mas sencillas. Familiarizarse con el Toolchain de desarrollo.</p>	Virtualidad Sincrónica
Clase 4:	<p>Teórica: Material de soporte Audiovisual: Video Clase Teórica 04 Temario: Set de Instrucciones ARM32, y T32. Instrucciones de ejecución condicional. Ventajas respecto de la integridad del pipeline. Instrucciones de Acceso a memoria Load - Store, y Load Store Múltiple. Manejo de Constantes, y referencias a direcciones inmediatas de 32 bits. Resolución por literal pools. Loops y Branches. Ejemplos de programación: Revisión de sintaxis de Makefile. Uso de salida de archivos listing en el GNU assembler para entender como se termina armando un literal pool, por el ensamblador. Arranque de un Cortex A9. Arranque de una Zynq 7000. Donde se ubica el código de inicialización de nuestro sistema. Análisis de los formatos de los objetos generados por el</p>	Presencial



	<p>assembler y el linker: ELF. Análisis del formato con comandos de consola: readelf, objdump.</p> <p>Debuging. gdb como core de debuging. Dashboard: Herramienta para su mejor visualización. Uso de estas herramientas para depuración de programas.</p> <p>Acceso a las syscalls de Linux en Qemu Zync7000, o Beagle Bone Black (lo que resulte mas cómodo).</p>	
Clase 5:	<p>Laboratorio:</p> <p>Taller de programación. Assembler ARM. Ejemplos. Programa de arranque del CORTEX A9. Práctica sobre Qemu emulando Zynq-7000.</p>	Presencial
Clase 6:	<p>Laboratorio:</p> <p>Taller de programación C y Assembler. ABI. Exploración del set de instrucciones. Práctica sobre Qemu emulando Zynq-7000. Los programas se ejecutan en Qemu en el momento de arranque del sistema. Identificar inconvenientes y restricciones. Plantear estrategia de arranque.</p>	Presencial
Clase 7:	<p>Evaluación Teórica Trayecto 1 Parte A (20 minutos)</p> <p>Teórica:</p> <p>Material de soporte Audiovisual: Video Clase Teórica 05</p> <p>Temario:</p> <p>Interrupciones. Definiciones generales. Tipos de interrupción y sus características mas distintivas. Identificación de las diferentes fuentes de interrupción. Proceso de general de Vectorización.</p> <p>Sistema de E/S: Acceso a los dispositivos periféricos. Técnicas de mapeo. Espacio de direccionamiento exclusivo para E/S vs. Dispositivos Memory mapped.</p> <p>Interrupciones de Hardware en ARM: Controlador de interrupciones. Prioridades. Máscaras. Disparo por flanco o por nivel.</p> <p>Diferencias fundamentales entre el perfil M y los perfiles A y R. Características del sistema de interrupciones del Cortex-M. El NVIC. Principales registros Memory Mapped.</p> <p>Características del sistema de interrupciones del Cortex-A y R. GIC. Implementaciones personalizadas de los fabricantes. Registros Memory Mapped.</p> <p>Gestión de interrupciones IRQ y FIQ. Handler de Interrupción Simple y Anidamiento. Recaudos a tomar con los registros banqueados en atención anidada. Resguardo del stack y del estado de máquina.</p>	Presencial
Clase 8:	<p>Laboratorio:</p> <p>Taller de programación Incorporar el vector de interrupciones y</p>	Presencial



	excepciones al Inicio del Sistema Bare Metal. Manejo del linker, definición de secciones de memoria, linker scripting básico para relocación de secciones de NVRAM a RAM	
Clase 9:	Teórica: Material de soporte Audiovisual: Video Clase Teórica 06 Temario: Concepto de Tarea como unidad de ejecución. Recursos. Contexto. Espacio de contexto. Context switch. Concepto general, estados de tareas. Primeros conceptos de System Programming. Planificación de Tareas. Espacios de tarea y de administración. Planificación de tareas: Estructuras de control genéricas. Políticas de scheduling. Restricciones temporales y como resolverlas. Implementación general de un scheduler (planificador) en ARM Cortex-A. Resguardo de los registros core en el stack. Resguardo de los registros FPU y SIMD: concepto de Lazy FPU context switch.	Presencial
Clase 10:	Evaluación Teórica Trayecto 1 Parte B (20 minutos) Teórica: Material de soporte Audiovisual: Video Clase Teórica 07 Temario: Desarrollo de Embedded Systems. Conceptos de System Programming. Firmware. Arranque de un SoC. Inicialización del Hardware. Hardware Abstraction Layer (HAL). ARMv7 perfiles A y R. System Programming. Descripción de Modos de ejecución, estados de ejecución y Niveles de privilegio. Recursos de arquitectura para su control y configuración. Manejo de excepciones en Modo Privilegiado. Banqueo de Registros en los diferentes modos de ejecución. Coprocesadores. Modelo general de ARM basado en Coprocesadores. CP14 y CP15. Funciones, modos de acceso. Instrucciones para acceso a los Coprocesadores. Formato y técnicas generales: Como entender el críptico manual de System Programming de ARM. Aplicación del Modelo de Programación de Sistemas. Sistemas Operativos para Embedded Systems. Evolución desde el modelo de Lazo de control, Sistemas Event Driven, Sistemas Interrupt Driven, Eventos sincrónicos y Asincrónicos, Priorización de eventos. Modelo de procesos. Sistemas Mono y Multiprocesador, espacio de direccionamiento Real o Virtual, Gestión estática o dinámica, Preemption. Sistemas derivados de estos. Real Time Systems	Presencial



Clase 11:	Laboratorio: Implementación de un scheduler estático básico. Contexto del procesador. Intercambio de stack.	Presencial
Clase 12:	Teórica: Material de soporte Audiovisual: Video Clase Teórica 08 Temario: Sistemas Operativos para Embedded Systems. Funciones de un Sistema Operativo. Arquitectura de un Sistema Operativo End to End. Rango de un Sistema Operativo. Scheduling de Procesos. IO-Bound vs. CPU-Bound, Tiempo de respuesta. Throughput. Algoritmos de scheduling: Round Robin, Dynamic Priority, Preemption. Scheduler Time sharing vs Real Time. Determinismo y Latency. Sincronización de Procesos. Técnicas: Wake-up. Semáforos: Operaciones, Semáforos Binarios, Mutexes. Problema productor consumidor. Escritor - Lector. Starvation. Dead Locks, características, estrategias de prevención. Intercomunicación de procesos de alto nivel: colas, pipes, etc. Gestión de memoria en ARMv7: VMSA y PMSA. Privilegios. Sistemas Operativos de propósito General: Una somera introducción para el siguiente curso. Sistemas Operativos de Tiempo Real. Características imprescindibles. Principales diferencias con los de propósito general. Algoritmos de Scheduling clásicos para RTOS: RDS y EDF. Descripción detallada de ambos algoritmos. Diagramas temporales. Características y limitaciones de c/u Características y escenarios complejos para RTOS: Inversión de prioridades y su prevención, Deadlines, Prioridad Techo, Herencia de prioridad. Casos de estudio: RTOS's: Free-RTOS, MicroC-OS, NuttX, QNX, RT-Linux. En cada caso se repasan arquitectura, características, rangos de aplicación, apego a estándares abiertos (con sus innegables ventajas). Diseño de un RTOS: Lineamientos fundamentales: Gestión de tareas, de memoria, políticas de scheduling, carácter estático o dinámico	Presencial
Clase 13:	Laboratorio: Pasaje de las tareas al espacio User. Implementación de algún control sencillo de protección de memoria. Análisis de las ventajas del banqueo de registros en cada cambio de modo	Presencial
Clase 14:	Evaluación Teórica Trayecto 1 Parte C (20 minutos)	Virtualidad



	<p>Teórica: Material de soporte Audiovisual: Video Clase Teórica 09 Temario: Elementos de diseño de Circuitos Impresos. Ruteo de señales analógicas, digitales y alimentación. Conceptos de Integridad de señal. Sustratos. Diferentes Tipos. Criterios de selección. Características físicas principales. Catálogos normalizados de materiales. Gestión de Temperatura. Ensayos.Tooling Holes. Lado Primario y Lado secundario Layout Capas. Redes de Alimentación. Layout óptimo. Capacitores de descope. Distribución de circuitos según frecuencia. Dimensionamiento de los conductores. Designadores. Sistemas de Grilla. Leyendas. Componentes y Montaje.Distancias entre componentes. Assembly Drawing. Ubicación de componentes fiduciales. Ubicación de puentes y jumpers. Panelización. Tipos de orificios y sus tolerancias. Vías. Edge connectors. Máscara antisoldante y coating. Documentación.</p>	Sincrónica
Clase 15:	<p>Laboratorio: Taller de diseño de PCB. Resolución del trabajo Práctico.</p>	Virtualidad Sincrónica
Clase 16:	<p>Laboratorio: Taller de diseño de PCB. Resolución del trabajo Práctico.</p>	Virtualidad Sincrónica
Clase 17:	<p>Evaluación Teórica Trayecto 2 (20 minutos) Teórica: Material de soporte Audiovisual: Video Clase Teórica 11 Temario: Principios de diseño de un Bus. Introducción general Especificación del Bus AMBAv2: Alta performance (AHB), System (ASB), y Periféricos (APB). Diagrama general y características de cada Bus. Conceptos de Ciclo de Bus, Transferencia de Bus y Operación de Bus. AHB: Master, Slave y Arbitro. APB, Bridge con el ASB. Características. Organización del Bus. Dispositivos que se conectan del lado AHB, y APB. Señales AHB y APB. Arquitectura detallada. Operación AHB. Protocolo Request Grant. Concepto de DMA como forma de ganar acceso directo a un Bus sin intervención de una CPU de propósito general. Ciclos de address y ciclos de control. Transferencias en modo Burst. Wait States. Control de errores. Transferencias AHB. Diagramas de timing para su operación. Transferencias Múltiples, Bust, interrupción de transferencias Burst. Busrt Wrapping. Burst Incremental. Burst de longitud</p>	Presencial



	<p>desconocida. Mención a la interfaz DRAM LLC que se analizará en el curso siguiente (TDIII) Señales de control. Decodificación de Direcciones. Respuesta desde el slave a las diferentes transacciones: Retry, Error, Split and Retry. Diagramas temporales para cada una. Armado de las líneas de datos: Endianess. Arbitraje del AHB. Diagrama de señales detallado. Diagramas de timing para : Bus Grant, Bus Grant con Wait State, Hand Over, Hand Over sobre un Burst, Manejo de HGrant, Split. Evolución en AMBAv3: AXI. Características nuevas y compatibilidad con AMBAv2.</p>	
Clase 18:	<p>Laboratorio: Inicialización de la tarjeta de desarrollo Zync 7000 del segundo cuatrimestre. Conexión a Vivado. Ejemplos minimalistas para asegurar funcionalidad. Uso de la IP para Protocolo de Bus AXI.</p>	Presencial
Clase 19:	<p>Teórica: Material de soporte Audiovisual: Video Clase Teórica 10 Temario: El sistema de Memoria. Rol de la Memoria en la performance de un computador. Capacidad, ancho de banda de bus, tiempo de acceso. Clasificación: Volátiles y No volátiles. Jerarquía. Localidad. Performance. Energía y Potencia. Costos. Casos prácticos de memorias jerárquicas. Estudio detallado de energía y potencia. Corrientes de fuga, energía y potencia estática. Estudio de la energía y potencia en conmutación. Estrategias tecnológicas para reducir la potencia y la energía. Memorias volátiles. Memorias dinámicas. Visión introductoria. Refresco. Características salientes. Memorias estáticas. Análisis detallado. Comparación de parámetros con respecto a la memoria dinámica. Breve introducción a la memoria cache. Funcionamiento de una celda de SRAM. Decodificación interna. Decodificación de direcciones. Organizaciones: Predecoding, No particionado, División de wordline, Decodificación de wordline jerárquica. Circuitos periféricos. Circuitos de pre carga y Ecuilibración: Diferentes configuraciones. Multiplexado de lectura/Escritura, Amplificadores de Detección: Diferentes configuraciones. Estructuras jerárquicas de detección. Amplificadores de escritura. Timing y conexión física. Conexión sincrónica. Acceso burst.</p>	Presencial



	Casos prácticos. Potencia y Energía. Dynamic Voltage Scaling. Powering Down. Gating Vdd. Potencia de leakage en SRAM.	
Clase 20:	Laboratorio: Implementación de una Block RAM en la FPGA de la Zync-7000. Acceso desde los Cortex A9 del SOC. Énfasis en decodificación del espacio de direcciones.	Presencial
Clase 21:	Teórica: Material de soporte Audiovisual: Video Clase Teórica 13 Temario: Conversión AD/DA Breve reseña del análisis de Fourier. Representación de señales en el dominio de la frecuencia. Digitalización de una señal. Muestreo Cuantificación Codificación. Circuito de Muestreo y Retención. Análisis funcional. Determinación del error de muestreo. Funciones de un Conversor AD: Cuantificación y Codificación: Error de Cuantificación en LSB. Implementación de un conversor AD: Diagrama general. Función transferencia. Implementación de un Sistema general de Adquisición de Datos. Filtrado analógico de entrada y digital de salida. Conversores de Alta Velocidad: Flash Converter: Diagrama General Código Termómetro. Problemas de diseño: Timing y Distorsión. Conversores de Alta Precisión. Simple y Doble Rampa. Breve descripción. Características fundamentales, fortalezas, y limitaciones tecnológicas. Conversores de Aproximaciones Sucesivas. Diagrama. Registro SAR. Funcionamiento. Características. Conversores Sigma-Delta. Diagrama funcional Modulador Sigma Delta. Funcionamiento. Principios teóricos que lo rigen Noise Shaping. Función del Filtro Digital decimador. Fundamentos de un sistema de Muestreo de datos; Relación Señal a Ruido en la entrada de un conversor. Performance Dinámica de un ADC de N-bit. Ruido en ADCs. Performance dinámica de un Conversor. Distorsión Armónica, Peor Armónica, Distorsión Armónica Total (THD), Distorsión Armónica Total mas Ruido (THD+N). Signal-to-Noise-and- Distortion Ratio (SINAD), Signal-to-Noise Ratio (SNR), y Effective Number of Bits (ENOB) ----- Laboratorio. Acceso al AD en Zynq-7000. Problema de adquisición de señales. Experimentar con el ADC de la tarjeta redpitaya.	Presencial



Clase 22:	Laboratorio: Práctica sobre Bus AXI. Diseño de un periférico trivial conectado como slave en el AXI de un SoC-FPGA basado en Xilinx Zynq-7010/7020. Acceso por DMA.	Presencial
Clase 23:	Evaluación Teórica Trayecto 4(20 minutos) Teórica: Material de soporte Audiovisual: Video Clase Teórica 12. Temario: Comunicación serie de baja velocidad Interfaces de un computador: Paralelo y Serie. Interfaces serie de baja velocidad. RS-232, RS-422, y RS-485, Breve reseña de sus características principales SPI: Diagrama funcional y características. Configuración Master Multi-Slave: Descripción funcional y diagramas de temporización. Señalización CLCK, SS, MISO y MOSI. Timing de operación Master Slave. Clock Sincrónico: Polaridad y Fase. Implementación SPI en BBB: SITARA 3358. I2C: Diagrama Funcional y características. Formato del mensaje. Direccionamiento. Read/Write Bit. Data Frame: Trama e una comunicación. Análisis temporal. Configuración Single Maater Multi Slave. Configuración Multi Master Multi Slave. Sensado de la línea SDA. Multimaster: Sincronización de clock, Arbitración del bus. Sumario de ventajas y desventajas. Implementación SPI en Zynq 7000. CAN: Características, formato de frame. Implementación CAN en Zynq 7000. Estándares adicionales. 1-Wire: Características principales, Topología de bus. Timing de Bus. Ejemplos. ----- Laboratorio: Ejemplos de Acceso al I2C y SPI de la SoC-FPGA: Conexionado de sensores para enriquecer la práctica.	Presencial
Clase 24:	Laboratorio: Continuación del Dispositivo personalizado en la FPGA que interactúa con los sensores y el ADC.	Presencial
Clase 25:	Teórica: Material de soporte Audiovisual: Video Clase Teórica 15 Temario: Memorias Flash. Funcionamiento del dispositivo FGMOS. Tecnologías NAND y NOR Flash. Características de ambas tecnologías y aplicación en cada caso en función de sus diferencias. Operación de escritura y borrado. Funcionamiento del dispositivo FGMOS como memoria de almacenamiento no volátil.	Presencial



	Lectura de una Memoria Flash. Estructura circuital de una Flash NAND y de una Flash NOR. Implementación en Array de celdas. Acceso por bus paralelo o serie (SPI). Estudio de modelos reales. Organización interna y configuración. Registro de estados. JEDEC Common Flash Interface.	
Clase 26:	Laboratorio: Conexión de un sensor I2C a la BBB. Acceso desde el sistema bare metal.	Presencial
Clase 27:	Laboratorio: Continuar con la resolución del acceso al Conversor AD de la Zynq-7000.	Presencial
Clase 28:	Teórica: Evaluación Teórica Trayecto 3 (20 minutos) Material de soporte Audiovisual: Video Clase Teórica 14. Temario: Universal Serial Bus Especificaciones generales. Genealogía 1.0, 1.1, 2.0, 3.0, 3.1, 3.2. Características mecánicas. Conectores. Características Eléctricas. Ventajas de la señal diferencial. Niveles de Tensión y corriente característicos. Detección de la conexión de un dispositivo al bus. Topología: Dispositivo Hub Root, Hubs, y Nodos y sus funciones. Codificación de la información. Protocolo. Endpoints de transmisión y Pipes. Frames. Tipos de transferencias: Características y aplicaciones de c/u. Transacciones: su estructura. Paquetes de datos y de handshake. Descomposición de las transferencias en paquetes de datos y handshake. Estructuras de Configuración: Descriptor de dispositivo, descriptor de Configuración, Descriptor de Interfaz, Descriptor de endpoint, Descriptor de string. Device requests. Enumeración. Análisis detallado de la secuencia completa. Análisis de un Bus físico real. Clases de dispositivos.	Presencial
Clase 29:	Laboratorio: Análisis de transacciones USB con Wireshark	Presencial
Clase 30:	Evaluación Teórica Trayecto 5 (20 minutos) Laboratorio: Análisis de transacciones USB con Wireshark	Presencial
Clase 31:	Laboratorio: Cierre de los TPs pendientes de entrega. Consultas	Presencial



Clase 30:	Laboratorio: Cierre de los TPs pendientes de entrega. Consultas	Presencial
-----------	---	------------

CRONOGRAMA ESTIMADO DE CLASES

BIBLIOGRAFÍA OBLIGATORIA

- Hennesy John, Patterson David (2006). Computer Organization and Design. The Hardware Software Interface. 5th. Ed. Morgan Kauffman, Elsevier.
- ARM Assembly Language. Fundamentals and Technics. 2nd. Ed. CRC Press. William Hohl , Christopher Hinds.
- Arm Limited. (2014). Instruction Set Assembly Guide for Armv7 and earlier Arm architectures Reference Guide Disponible en formato electrónico en <https://developer.arm.com/documentation/100076/latest/>
- Arm Limited. (2011). Cortex-A9 Technical Reference Manual. Disponible en formato electrónico en <https://developer.arm.com/documentation/ddi0388/latest/>
- ARM. (2010). ARM Generic Interrupt Controller. Retrieved from http://infocenter.arm.com/help/topic/com.arm.doc.ihl0048a/IHL0048A_gic_architecture_spec_v1_0.pdf
- Sloss, A., Symes, C., & Wright, D. (2004). ARM System Developer's Guide: Designing and Optimizing System Software. Morgan Kaufmann. ISBN-13: 978-1558608740.
- K. C. Wang (2015). Design and Implementation of the MTX Operating System. Springer
- K. C. Wang (2017). Embedded and Real-Time Operating Systems. Springer
- Bruce Jacob, Spencer W. Ng, David T. Wang. Memory Systems Cache, DRAM, Disk. Elsevier, Morgan Kaufman.
- Axelson, J. (2000). USB Complete: The Developer's Guide (Complete Guides series). Lakeview Research. ISBN-13: 978-1931448086.
- Data Conversion Handbook. Walt Kester, Editor with the technical staff of Analog Devices. Analog Devices. 2005.
- Louise H. Crockett, Ross A. Elliot, Martin A. Enderwitz, Robert W. Stewart. The Zynq Book. Embedded Processing with the ARM® Cortex®-A9 on the Xilinx® Zynq®-7000 All Programmable SoC. 1st Edition. Descargable desde: <http://www.zynqbook.com/download-book.php>



- Louise H. Crockett, Ross A. Elliot, Martin A. Enderwitz, David Northcote. The Zynq® Book Tutorials for Zybo and ZedBoard. Descargable desde: <http://www.zynqbook.com/download-book.php>

BIBLIOGRAFÍA COMPLEMENTARIA

- Li, Q., & Yao, C. (2003). Real-Time Concepts for Embedded Systems. CMP. ISBN-13: 978-1578201242.
- Hyde, J. (n.d.). USB Design by Example: A Practical Guide to Building I/O Devices. Intel University Press. ISBN-13: 978-0970284655.
- Bruce Smith (2017). ARM A32 Assembly Language. 32-bit, Neon, VFP, Thumb2.