

## SCHEDA DELL'INSEGNAMENTO (SI)

### "CIRCUITI PER DSP"

SSD ING-INF/01

DENOMINAZIONE DEL CORSO DI STUDIO: CORSO DI LAUREA MAGISTRALE IN INGEGNERIA  
ELETTRONICA

ANNO ACCADEMICO 2023-2024

#### INFORMAZIONI GENERALI - DOCENTE

DOCENTE: DAVIDE DE CARO

TELEFONO: 081-7683136

EMAIL: [DAVIDE.DECARO@UNINA.IT](mailto:DAVIDE.DECARO@UNINA.IT)

#### INFORMAZIONI GENERALI - ATTIVITÀ

ANNO DI CORSO (I, II, III): II

SEMESTRE (I, II): I

CFU: 9

## **INSEGNAMENTI PROPEDEUTICI (se previsti dall'Ordinamento del CdS)**

Nessuna

### **EVENTUALI PREREQUISITI**

Conoscenza di base del funzionamento dei circuiti digitali e del linguaggio C.

### **OBIETTIVI FORMATIVI**

L'insegnamento si pone l'obiettivo di fornire agli studenti le nozioni avanzate e di alto livello relative alla implementazione degli algoritmi di elaborazione digitale dei segnali e, inoltre, di acquisire le conoscenze, sia di base sia avanzate, relative alle architetture dei circuiti DSP disponibili commercialmente, delle problematiche, sia teoriche che pratiche, relative alla implementazione ottimale, in tempo reale, su DSP, dei principali algoritmi di elaborazione digitale dei segnali e dell'ambiente di sviluppo per la loro programmazione.

### **RISULTATI DI APPRENDIMENTO ATTESI (DESCRITTORI DI DUBLINO)**

#### **Conoscenza e capacità di comprensione**

Al termine del processo di apprendimento lo studente è edotto sulle principali tecniche di elaborazione dei dati in virgola fissa e mobile, sugli strumenti di analisi degli errori di rappresentazione e delle problematiche di overflow in un sistema lineare realizzato in aritmetica a virgola fissa, sulle principali tecniche di prevenzione/gestione dell'overflow, sulle problematiche teoriche relative alla descrizione in linguaggio C di algoritmi di elaborazione digitale dei segnali, sulle caratteristiche architettoniche di minima che identificano un circuito DSP (Digital Signal Processor), sulle problematiche relative alla pipeline nei DSP, sia nel caso di pipeline protetta che nel caso di pipeline non-protetta, sulle caratteristiche dei DSP ad elevato parallelismo di elaborazione -in particolar modo basati su architettura di tipo Very Long Instruction Word (VLIW)-, alle forme di parallelismo al livello dell'istruzione set del DSP -istruzioni Single-Instruction-Multiple-Data-, sulle tecniche di ottimizzazione del codice per le architettura VLIW -in particolar modo utilizzando tecniche di Loop Unrolling e Software Pipelining-, sulle caratteristiche di base e avanzate dei sistemi di interruzione e DMA (Direct-Memory-Access) nei DSP, sulle caratteristiche delle interfacce di comunicazione del DSP -in particolar modo quelle di tipo seriale sincrono-, sulle problematiche relative al realizzazione su un DSP di sistemi di elaborazione in tempo reale -in particolar modo utilizzando l'elaborazione in streaming oppure a blocchi-.

#### **Capacità di applicare conoscenza e comprensione**

Al termine del processo di apprendimento lo studente è in grado di progettare, autonomamente, l'implementazione in aritmetica a virgola fissa e mobile, di un algoritmo di elaborazione dei segnali tenendo conto sia degli errori di rappresentazione che delle problematiche connesse all'overflow e di determinare l'approccio di prevenzione/gestione dell'overflow che meglio si adatta alla specifica applicazione.

Lo studente acquisisce inoltre la capacità di descrivere in linguaggio C un algoritmo di elaborazione digitale dei segnali con particolare riferimento alle problematiche relative alla efficiente implementazione su DSP sia a virgola fissa che a virgola mobile. Lo studente padroneggia inoltre le tecniche di ottimizzazione del codice, con particolare riferimento alle architetture VLIW ed è in grado di gestire nel modo migliore possibile, in relazione alla specifica applicazione, il trade-off tra tempo di calcolo e dimensione del codice applicando le tecniche di Loop Unrolling e Software Pipelining.

In fine, lo studente padroneggia gli approcci per l'elaborazione in tempo reale di un algoritmo di elaborazione digitale su un DSP, e, in relazione alle caratteristiche della specifica applicazione, può risolvere nel modo migliore possibile il trade-off tra massima frequenza di elaborazione e latenza adottando approcci di elaborazione in streaming oppure a blocchi. Lo studente inoltre padroneggia le problematiche inerenti l'impiego delle unità DMA sia a supporto dell'elaborazione, sia a servizio delle interfacce del DSP nella elaborazione a blocchi.

#### **Autonomia di giudizio, Abilità comunicative, Capacità di apprendimento**

In relazione a tutte le capacità descritte in precedenza, lo studente acquisisce inoltre capacità autonoma di giudizio essendo, questa, tra l'altro, nel particolare contesto, un presupposto imprescindibile dell'attività creativa di tipo progettuale che rientra nelle capacità di applicare la conoscenza che sono sviluppate.

Durante l'insegnamento gli studenti vengono inoltre stimolati nella acquisizione degli strumenti che consentono l'approfondimento in modo autonomo degli argomenti trattati, mentre le metodologie di verifica dell'apprendimento da parte degli studenti tendono a sviluppare anche le loro abilità comunicativa.

## PROGRAMMA-SYLLABUS

**Introduzione:** Circuiti per l'Elaborazione Digitale: ASIC, FPGA, DSP. Elaborazioni Real-Time.

**Problematiche numeriche nella rappresentazione dei segnali:** Richiami sulle rappresentazione Fixed-Point e Floating-Point; Precisione ed Errori di Quantizzazione; Errori di Quantizzazione al livello di Sistema; Overflow e tecniche di prevenzione: Variazione della Rappresentazione, Scaling, Scaling per segnali a banda stretta, Scaling Statistico, Aritmetica Saturata, Bit di Guardia; Applicazione delle tecniche di prevenzione dell'Overflow ad un Filtro FIR. Descrizione degli algoritmi di DSP in linguaggio C: rappresentazione Fixed-Point Generalizzata, Regole di Promozione del Tipo, Esempio (filtro FIR con tecnica dei bit di Guardia).

**Architetture di base dei circuiti DSP:** Datapath; Architettura Harvard ed Harvard modificata; Memoria nei DSP: Banchi Separati, Banchi in Interweaving, Memorie Dual-ported, Allineamento dei Dati ed Accessi Multipli, Gerarchie delle memorie, Caching coerenza e predicibilità; Repeat Buffer; Indirizzamenti per DSP ed Unità di generazione degli indirizzi: indirizzamento Circolare e Bit-Reversal; Tecniche di Zero-Overhead-Looping; Semplici periferiche nei DSP: Timers/PWM generators; Esempi di DSP della Texas Instruments: architetture C2xx, C54xx e C55xx.

**Architetture avanzate dei DSP:** Trade-off tra Ortogonalità e Dimensione del Codice; Pipelining nei DSP: Pipeline non-Protetta o Visibile, Pipeline Protetta o Trasparente, Hazards Strutturali, Hazard sui Dati e Dipendenze, Hazard di Controllo, Delayed-Branch; Scheduling Statico e Dinamico delle Istruzioni; Architetture Superscalari (cenni); Architetture Very-Long-Instruction-Word (VLIW); Pipeline non-Protetta ed Interruzioni (trade-off tra interrompibilità del codice e tempo di calcolo); Istruzioni ed Aritmetica Single-Instruction-Multiple-Data (SIMD); DSP VLIW della Texas Instruments: architetture C64xx, C67xx, C66xx. Cenni ai principali DSP di Analog Devices (Blackfin, SigmaDSP, SHARC e TigerSHARC) e Freescale (Symphony, StarCore SC3400 e SC3850). DSP con Multi-threading Hardware: architettura Hexagon di Qualcomm. SoC basati su DSP di Texas Instruments: DaVinci Digital Media Processor, OMAP, Keystone e Keystone II.

**Architettura dei DSP Texas Instruments C64xx e C67xx:** Architettura VelociTI; Set Istruzioni; Architettura delle Memorie e Caching su due livelli; Esecuzione Condizionale; Indirizzamenti con offset; Indirizzamenti Circolari e Registro AMR; Pipeline del DSP: Delay-Slots e Latenza delle istruzioni; Istruzioni SIMD.

**Sviluppo del Codice ed Ottimizzazione per DSP VLIW:** Sviluppo in Assembly: Grafi delle Dipendenze, Parallelizzazione delle Istruzioni, Eliminazione dei NOP, Loop Unrolling; Software Pipelining: Minimum-safe-trip-count, Resource-Bound, Loop-Carried-Resource-Bound, Esecuzione Speculativa, Problemi di Live-too-Long, Utilizzo congiunto di Loop Unrolling e Software Pipelining; Sviluppo del codice in Linear-Assembly; Sviluppo del Codice in C: direttive per il Software Pipelining e l'Unrolling, Aliasing dei Puntatori, Direttive per l'Allocazione in Memoria, Livelli di Ottimizzazione, Funzioni Static, Interrompibilità del Codice, Software Pipelined Loop Buffer (SPLOOP); Linker, Variabili Globali e Static, Allocazione delle variabili Near e Far, Start-up del DSP.

**Interrupt e DMA nei DSP Texas Instruments C6x:** Interrupt: funzionalità di base, registri per la gestione degli interrupt, funzionamento dettagliato in Hardware, scrittura della Interrupt-Service-Routine (ISR), tempi caratteristici; DMA: funzionalità di base, parametri e tipologie di trasferimento, sincronizzazione, Quick-DMA, generazione di interrupt di completamento, esempi.

**Implementazione Real-Time degli Algoritmi di DSP:** Interfacce Seriali Sincrone ed interfaccia McBSP dei DSP Texas C6x; Gestione delle comunicazioni in Polling, Interrupt, DMA; Elaborazioni in Streaming di tipo Interrompibile e non-Interrompibile, Controllo di Flusso nelle elaborazioni in Ricezione/Trasmissione; Sistemi Operativi Real-Time e Elaborazioni con Prelazione; Elaborazioni a Blocchi: Ping-Pong Buffering; Tecniche di Buffering per gli Algoritmi con Memoria; Debugging off-line e real-time nei DSP, Tecniche di In-System Debugging tramite interfaccia JTAG.

**Esercitazioni:** Le esercitazioni sfruttano l'ambiente di sviluppo Code Composer Studio di Texas Instruments e le schede per la prototipizzazione rapida su DSP DSK-C6713.

Implementazioni di Filtri FIR: Calcolo dei Coefficienti, Progetto Numerico e Problematiche di Overflow, Ottimizzazione del Codice con esempi di Unrolling, Software Pipelining, Buffering dei dati; Prove sperimentali con I/O in Real-time dal Codec audio presente sulla board DSK-C6713: filtro FIR, equalizzatore audio a 5 bande, equalizzatore audio a 5 bande mediante Sistema Operativo real-time, equalizzatore audio a 9 bande mediante Sistema Operativo real-time.

## MATERIALE DIDATTICO

- John G. Proakis, Dimitris G. Manolakis, "Digital Signal Processing: Principles, Algorithms and Applications", 4° edition, Prentice Hall 2007
- Sen M. Kuo, Woon-Seng Gan, "Digital Signal Processors: Architectures, Implementations, and Applications", Prentice Hall 2005
- Appunti delle lezioni
- Testi delle esercitazioni

## MODALITÀ DI SVOLGIMENTO DELL'INSEGNAMENTO

Lo svolgimento dell'insegnamento prevede lezioni frontali, esercitazioni e, compatibilmente con gli aspetti organizzativi, lezioni di laboratorio.

Per lo svolgimento delle esercitazioni gli studenti adottano l'ambiente di sviluppo Code Composer Studio della Texas Instruments. Il laboratorio, oltre a Code Composer Studio, prevede l'impiego di board per il prototyping rapido (DSK6713 della Texas Instruments) e la strumentazione di base di un laboratorio di elettronica (oscilloscopio, generatore di segnali).

## VERIFICA DI APPRENDIMENTO E CRITERI DI VALUTAZIONE

a) Modalità di esame:

L'esame si articola in prova	
scritta e orale	
solo scritta	
solo orale	✓
discussione di elaborato progettuale	
altro (discussione esercitazioni)	✓