



UNIVERSITÀ DEGLI STUDI DI PALERMO

DIPARTIMENTO	Matematica e Informatica
ANNO ACCADEMICO OFFERTA	2018/2019
ANNO ACCADEMICO EROGAZIONE	2018/2019
CORSO DILAUREA	INFORMATICA
INSEGNAMENTO	ARCHITETTURE DEGLI ELABORATORI
TIPO DI ATTIVITA'	A
AMBITO	50168-Formazione informatica di base
CODICE INSEGNAMENTO	16450
SETTORI SCIENTIFICO-DISCIPLINARI	INF/01
DOCENTE RESPONSABILE	ROMBO SIMONA ESTER Professore Ordinario Univ. di PALERMO
ALTRI DOCENTI	
CFU	6
NUMERO DI ORE RISERVATE ALLO STUDIO PERSONALE	102
NUMERO DI ORE RISERVATE ALLA DIDATTICA ASSISTITA	48
PROPEDEUTICITA'	
MUTUAZIONI	
ANNO DI CORSO	1
PERIODO DELLE LEZIONI	2° semestre
MODALITA' DI FREQUENZA	Facoltativa
TIPO DI VALUTAZIONE	Voto in trentesimi
ORARIO DI RICEVIMENTO DEGLI STUDENTI	ROMBO SIMONA ESTER Lunedì 9:30 13:30 In presenza: Via Archirafi 34, Piano II, Stanza 220 - Telematico: via Microsoft Teams o altro canale - In entrambi i casi, e' consigliabile prenotarsi tramite email alla docente

DOCENTE: Prof.ssa SIMONA ESTER ROMBO

PREREQUISITI	Nessuno.
RISULTATI DI APPRENDIMENTO ATTESI	<p>Conoscenza e capacita' di comprensione Acquisizione delle conoscenze fondamentali di base sull'organizzazione e il funzionamento degli elaboratori elettronici.</p> <p>Capacita' di applicare conoscenza e comprensione Capacita' di valutare le prestazioni dei sistemi di elaborazione sulla base della piena comprensione delle soluzioni tecniche utilizzate. Programmazione in linguaggio Assembly.</p> <p>Autonomia di giudizio Capacita' di analizzare e valutare l'architettura di un calcolatore secondo i componenti che lo costituiscono.</p> <p>Abilita' comunicative Capacita' di descrivere soluzioni di progettazione dei sistemi di calcolo mediante l'analisi delle specifiche tecniche fornite. Capacita' di cooperare e lavorare in team per determinare delle soluzioni di progettazione appropriate in alcuni casi specifici.</p> <p>Capacita' d'apprendimento Capacita' di aggiornamento con la consultazione di testi avanzati. Capacita' di seguire, utilizzando le conoscenze acquisite nel corso, sia corsi di master di primo livello, che corsi di laurea magistrali.</p>
VALUTAZIONE DELL'APPRENDIMENTO	<p>Prova scritta e prova orale. Nel semestre in cui viene tenuto il corso, e in particolare durante l'apposita pausa dalle attivita' didattiche, verra' effettuata una prova in itinere che, se superata, integrera' la prova scritta, sostituendo il primo esercizio della stessa.</p> <p>Prova scritta: sara' composta da tre esercizi, il primo volto a verificare la capacita' di effettuare la sintesi di reti sequenziali attraverso l'uso di automi a stati finiti, il secondo necessario per valutare la capacita' di progettare una macchina microprogrammata, infine il terzo servira' a verificare che lo studente sia in grado di programmare in Assembly. La prova verra' superata se lo studente dimostrera' una conoscenza almeno di base su ciascuna delle tre parti del programma. Il numero e la gravita' degli errori negli esercizi svolti determineranno lo scostamento dal voto massimo (30).</p> <p>Prova orale: Approfondira' le conoscenze relative a ciascuna parte del programma, completando il quadro del giudizio attraverso la valutazione della capacita' critica e dell'autonomia nel trovare soluzioni. Si potra' richiedere lo svolgimento di alcuni esercizi per compensare eventuali lacune evidenziate dall'esito del test scritto, e si proporranno problemi di progettazione anche complessi agli studenti che mostreranno ottima padronanza degli argomenti svolti a lezione. La valutazione della prova orale andra' a integrare quella della prova scritta, determinando il voto finale attraverso una media tra le due prove. Il voto minimo per il superamento dell'esame (18/30) verra' attribuito se si dimostrera' una conoscenza minima degli argomenti che ricoprono tutto il programma del corso. Quanto piu' le conoscenze saranno raffinate, tanto piu' il voto sara' alto. In particolare:</p> <p>18-21: Conoscenza sufficiente di tutte le parti del programma. 22-24: Conoscenza discreta di tutte le parti del programma. 25-27: Conoscenza buona di tutte le parti del programma. 28-30: Conoscenza ottima di tutte le parti del programma.</p> <p>La lode verra' attribuita se lo studente avra' dimostrato capacita' di applicare le conoscenze acquisite durante il corso per risolvere autonomamente problemi complessi la cui soluzione non e' stata discussa durante le lezioni.</p>
OBIETTIVI FORMATIVI	<p>L'obiettivo del corso di Architetture degli Elaboratori e' quello di fornire agli studenti di Informatica le conoscenze fondamentali sull'organizzazione degli elaboratori elettronici e sui principi alla base del loro funzionamento. Si studiera' la teoria dei circuiti digitali (analisi e sintesi di reti combinatorie e macchine sequenziali), come anche alcuni aspetti piu' avanzati dell'organizzazione e delle architetture dei sistemi di elaborazione. Inoltre, si forniranno gli strumenti essenziali per la comprensione dei principi di un linguaggio assemblativo e per il suo utilizzo.</p>
ORGANIZZAZIONE DELLA DIDATTICA	Lezioni frontali.
TESTI CONSIGLIATI	<p>F. Luccio, L. Pagli – Reti logiche e calcolatore, Bollati Boringhieri. Dispense su Assembly (Manuscript on Assembly). Esercizi svolti e tracce di esercizi da svolgere su tutte le parti del programma (Exercises and Tests on all the parts of the program).</p>

PROGRAMMA

ORE	Lezioni
2	NOZIONI INTRODUTTIVE Rappresentazione delle informazioni. Aritmetica binaria. Organizzazione, struttura e livelli di progettazione degli elaboratori elettronici.
3	RETI LOGICHE (I) Concetti di base della sintesi combinatoria e sequenziale. Algebra booleana: funzioni, forme minime, forme normali.
4	RETI LOGICHE (II) Progettazione di reti e moduli combinatori: tecniche di minimizzazione, codificatori-decodificatori, multiplexer-demultiplexer. Unita' aritmetico-logica (ALU).
6	SINTESI DI RETI SEQUENZIALI Reti sequenziali asincrone. Reti sequenziali sincrone. Sintesi di reti sequenziali attraverso automi a stati finiti.
4	SINTESI DI RETI SEQUENZIALI: Esercizi.
4	PROGETTAZIONE DI SISTEMI (I) Parte operativa e parte controllo; interconnessione tra componenti. Macchina di Von-Neumann. Microsequenze di FETCH ed EXECUTE. Introduzione alla microprogrammazione e sua implementazione.
4	PROGETTAZIONE DI SISTEMI (II) Progettazione di sistemi a microprocessore: progettazione della parte operativa, interfacciamento con la memoria, progettazione della unita' di controllo.
4	PROGETTAZIONE DI SISTEMI: Esercizi.
3	ASSEMBLER (I) Assemblatori, collegatori e cenni alle relazioni con la compilazione. Il linguaggio Assembly. Il processore 80386: gestione della memoria, tipi di dato, istruzioni. Programmazione in Assembly utilizzando il Netwide Assembler (NASM): istruzione MOV, stampa su video. Utilizzo del debugger GDB. Esercizi.
4	ASSEMBLER (II) Istruzioni elementari in NASM (operazioni logico-aritmetiche, SHIFT, CMP, istruzioni di salto). Esercizi.
5	ASSEMBLER (III) Gestione degli array in NASM. Esercizi.
5	Gerarchie di memoria, cache, parallelismo e pipeline. Dettagli su FETCH in situazioni piu' specifiche e complesse.