



UNIVERSITÀ DEGLI STUDI DI PALERMO

| | | | |
|---|---|---------------------------------|------------------|
| DIPARTIMENTO | Matematica e Informatica | | |
| ANNO ACCADEMICO OFFERTA | 2019/2020 | | |
| ANNO ACCADEMICO EROGAZIONE | 2019/2020 | | |
| 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 | BELLAVIA FABIO | Ricercatore a tempo determinato | Univ. di PALERMO |
| ALTRI DOCENTI | | | |
| CFU | 6 | | |
| NUMERO DI ORE RISERVATE ALLO STUDIO PERSONALE | 86 | | |
| NUMERO DI ORE RISERVATE ALLA DIDATTICA ASSISTITA | 64 | | |
| 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 | BELLAVIA FABIO Giovedì 14:00 16:00 DMI - Stanza 203 | | |

| | |
|--|--|
| PREREQUISITI | Nessun prerequisito |
| RISULTATI DI APPRENDIMENTO ATTESI | <p>Conoscenza e capacita' di comprensione Gli studenti acquisiranno una buona conoscenza dei fondamenti scientifici e tecnologici delle architetture dei sistemi di elaborazione, delle caratteristiche dei diversi livelli e moduli che li compongono e le loro relazioni con gli strumenti di programmazione.</p> <p>Capacita' di applicare conoscenza e comprensione Gli studenti dimostreranno le loro capacita' di utilizzare le conoscenze acquisite attraverso l'analisi, la progettazione, la programmazione, e l'ottimizzazione di moduli e unita' di controllo a diversi livelli dell'architettura degli elaboratori. Gli studenti acquisiranno la capacita' di valutare le prestazioni dei sistemi di elaborazione, la conoscenza del linguaggio di programmazione assembly della macchina e la capacita' di usare tale linguaggio per scrivere programmi anche complessi.</p> <p>Autonomia di giudizio Gli studenti acquisiranno la capacita' di valutare le esigenze tecniche che si presentano nella progettazione di circuiti (costo di progettazione, velocita' di elaborazione) e di prendere decisioni metodologiche in merito alla progettazione dell'architettura di un calcolatore.</p> <p>Abilita' comunicative Gli studenti acquisiranno la conoscenza dei formalismi e della terminologia usati per la descrizione di circuiti e parti dei calcolatori.</p> <p>Capacita' di apprendimento Le conoscenze acquisite durante il corso permetteranno allo studente di maturare nuove capacita' di apprendimento e di affinare le tecniche, i metodi e gli strumenti in maniera autonoma. Gli studenti acquisiranno la capacita' di rapportare e confrontare le caratteristiche dell'organizzazione e della progettazione degli elaboratori.</p> |
| VALUTAZIONE DELL'APPRENDIMENTO | <p>I metodi di verifica con cui gli studenti dimostreranno di aver acquisito capacita' e conoscenze apprese durante il corso, consistono in una prova finale scritta e in una prova finale orale.</p> <p>Prova finale scritta: mira a verificare le capacita' dello studente relativamente alla presentazione dei concetti di base e all'utilizzo delle metodologie di progettazione analizzate nel corso. Essa consiste di un certo numero di quesiti e nella risoluzione di un certo numero di esercizi analoghi a quelli presentati nel corso delle lezioni in aula e del laboratorio. Tale prova servira' a verificare l'apprendimento di specifiche capacita':</p> <ul style="list-style-type: none"> - conoscere i componenti principali dell'architettura di un calcolatore; - risolvere problemi relativi alla progettazione di semplici moduli e alla determinazione delle loro performance; - saper programmare in assembly interagendo correttamente con i componenti del calcolatore; - saper utilizzare le conoscenze acquisite per risolvere problemi applicativi presentati sotto forma di esercizi, domande e progetti. <p>Alla prova verra' assegnato un voto in trentesimi (al piu' 30) che sara' inversamente proporzionale al numero e alla gravita' degli errori fatti dallo studente. Affinche' la prova venga considerata superata e' necessario che lo studente dimostri di aver acquisito la conoscenza almeno di base di ciascuna delle parti trattate, raggiungendo un voto minimo di 18/30.</p> <p>Durante il semestre di svolgimento del corso, e in particolare durante l'apposito periodo dedicato dal calendario didattico allo svolgimento di esami, e' prevista una sola prova di verifica intermedia (o prova in itinere) che, se superata con esito positivo, integrera' la prova scritta, sostituendo una parte degli esercizi presenti nella stessa.</p> <p>Prova finale orale: unica e obbligatoria per tutti, servira' a verificare e approfondire le conoscenze generali acquisite dallo studente e la padronanza degli argomenti trattati nel programma. Essa consiste in alcune domande relative ai concetti che non sono stati direttamente affrontati durante la prova scritta. Inoltre, qualora la prova scritta metta in evidenza eventuali lacune, al fine di compensare le stesse, si potra' richiedere lo svolgimento su carta di alcuni esercizi e problemi di progettazione. Per poter sostenere la prova orale e' necessario aver superato la prova scritta con un voto maggiore o uguale a 18.</p> <p>Voto Finale: La valutazione finale e' espressa in trentesimi e il voto minimo per il superamento dell'esame e' 18/30. Il voto finale e' il risultato della media tra il voto raggiunto nella prova scritta e il</p> |

| | |
|---------------------------------------|--|
| | <p>voto raggiunto nella prova orale e tiene conto della griglia di valutazione riportata di seguito:</p> <p>18-20 -> Conoscenza minima di base sugli argomenti principali del corso e utilizzo di un linguaggio tecnico sufficiente con poca capacita' di applicare in modo autonomo le conoscenze e le metodologie acquisite.</p> <p>21-23 -> Discreta conoscenza di base sugli argomenti principali del corso e soddisfacente utilizzo della terminologia specifica con incertezze in alcuni passaggi e nell'esposizione di alcuni argomenti.</p> <p>24-26 -> Buona conoscenza dei concetti e delle metodologie necessarie per la soluzione dei quesiti proposti e ampia conoscenza delle parti del programma con qualche incertezza nell'esposizione di alcuni argomenti.</p> <p>27-29 -> Ottima conoscenza degli argomenti trattati nel programma e utilizzo di una terminologia adeguata.</p> <p>Capacita' di fornire soluzioni complete e corrette in maniera autonoma e dimostrando sicurezza e padronanza nella descrizione dei passaggi e nell'esposizione degli argomenti.</p> <p>30-30L-> Eccellente conoscenza degli argomenti e della terminologia adottata. Competenze tecniche e capacita' di approfondire sotto vari aspetti gli argomenti trattati. Ottime capacita' di utilizzo delle conoscenze acquisite, di analisi, e di collegamento tra i concetti visti a lezione.</p> |
| OBIETTIVI FORMATIVI | <p>Il corso si propone di fornire una buona conoscenza dei fondamenti scientifici e tecnologici delle architetture dei sistemi di elaborazione, le conoscenze e le abilita' minime necessarie al progetto e alla programmazione degli elementi costruttivi degli elaboratori. Gli argomenti in programma sono selezionati in modo da fornire:</p> <ul style="list-style-type: none"> - I fondamenti dei circuiti digitali e le tecniche di progettazione - La conoscenza dell'organizzazione dell'architettura dei sistemi di elaborazione - La conoscenza degli strumenti di programmazione e le capacita' di sviluppare componenti a diversi livelli dell'architettura dei sistemi di elaborazione |
| ORGANIZZAZIONE DELLA DIDATTICA | <p>La didattica e' organizzata mediante lezioni frontali in aula e in laboratorio. E' previsto inoltre un approccio didattico guidato da obiettivi, problemi e requisiti, che permette un trasferimento della conoscenza, anche attraverso il supporto dei mezzi informatici.</p> |
| TESTI CONSIGLIATI | <p>Testi di riferimento:</p> <ul style="list-style-type: none"> - Architettura degli Elaboratori, M. Vanneschi, Seconda Edizione, Pisa University Press (2013) - Esercizi svolti, tracce di esercizi da svolgere, e slide delle lezioni verranno rese disponibili via web. <p>Main books:</p> <ul style="list-style-type: none"> - Architettura degli Elaboratori, M. Vanneschi, Second Edition, Pisa University Press (2013) - Exercises with solutions, tests, and slides of the lesson on all the parts of the program will be made available on the web. <p>Testi consigliati:</p> <ul style="list-style-type: none"> - Reti Logiche, M. Morris Mano, Charles R. Kime, Tom Martin, Quinta Edizione, Pearson (2019) <p>Il libro copre i concetti iniziali del corso: i fondamenti della progettazione hardware e le ultime tecniche e tecnologie per la progettazione e lo sviluppo di circuiti digitali e concetti di architettura del computer.</p> <ul style="list-style-type: none"> - Architettura dei calcolatori. Un approccio strutturale, Andrew S. Tanenbaum, Todd Austin, Sesta Edizione, Pearson (2013) <p>Questo libro copre tutti i concetti del corso e discute le piu' recenti tecnologie e gli ultimi sviluppi nell'organizzazione e nell'architettura dei computer.</p> <p>Recommended books:</p> <ul style="list-style-type: none"> - Logic & Computer Design Fundamentals, M. Morris Mano, Charles R. Kime, Tom Martin, 5th Edition, Pearson (2016) <p>The book covers the initial concepts of the course: the fundamentals of hardware design and the latest techniques and technologies for the design and development of digital circuits and computer architecture concepts.</p> <ul style="list-style-type: none"> - Structured Computer Organization, Andrew S. Tanenbaum, Todd Austin, 6th Edition, Pearson (2013) <p>This book covers all the concepts of the course and it discusses the most recent technologies and the latest developments in computer organization and architecture.</p> |

PROGRAMMA

| ORE | Lezioni |
|-----|---|
| 3 | Nozioni introduttive e fondamenti dei sistemi di elaborazione. Struttura generale e livelli di astrazione dei sistemi di elaborazione. Compilazione e interpretazione. Rappresentazione binaria e algebra di Boole. |

PROGRAMMA

| ORE | Lezioni |
|-----|---|
| 3 | Concetti di base dei circuiti logici combinatori. Definizione di reti combinatorie, loro formalizzazione e comportamento. Specifica di reti combinatorie. Procedimento di sintesi. Forme canoniche e semplificazione. |
| 3 | Definizione di reti sequenziali, loro formalizzazione e comportamento. Classificazione e tipi di reti sequenziali. |
| 3 | Progettazione e sintesi di reti sequenziali. Reti sequenziali realizzate con componenti standard e loro analisi. Componente logico memoria. Unita' di elaborazione: Modello Parte Controllo - Parte Operativa. |
| 3 | Il procedimento di progettazione delle unita' di elaborazione. Macchina di Von-Neumann. Modello di esecuzione FETCH-DECODE-EXECUTE. Introduzione alla microprogrammazione e implementazione dell'unita' di controllo della CPU. |
| 4 | Microarchitettura: microistruzioni, interfacciamento con la memoria, ottimizzazione e miglioramento delle prestazioni. |
| 3 | Livello del sistema operativo: processi cooperanti e supporto a tempo di esecuzione dei processi. |
| 4 | Introduzione alla cache, MMU con cache associativa, principi di localita' e riuso. Tempi di accesso, politiche on demand, modi di indirizzamento: diretto, associativo e associativo su insiemi. |
| 3 | Politiche di gestione dei vari livelli di memoria. Paginazione e segmentazione. Ingresso uscita: trattamento interruzioni, funzionamento del memory mapped I/O e accesso diretto alla memoria (DMA). |
| 3 | Politiche di gestione della cache. Prefetching. Politica LRU, tempo di esecuzione con fault. |
| ORE | Laboratori |
| 4 | Progettazione di moduli combinatori: codificatori-decodificatori, multiplexer-demultiplexer, reti di calcolo aritmetico-logiche. |
| 4 | Svolgimento guidato di esercizi sulla progettazione di reti combinatorie e sequenziali. |
| 4 | Svolgimento guidato di esercizi sulla progettazione e l'analisi delle unita' di elaborazione. |
| 4 | Il livello della macchina assembler: assembleri, metodi di indirizzamento. Il processore 80386: gestione della memoria, tipi di dato e istruzioni. |
| 3 | Programmazione in Assembly utilizzando il Netwide Assembler (NASM): istruzioni elementari e gestione array. Utilizzo del debugger GDB. |
| 4 | Svolgimento guidato di esercizi e programmi sul linguaggio Assembly |
| 3 | Fondamenti di elaborazione in parallelo. Parallelismo a livello di microarchitettura: memoria interlacciata. Paradigma Pipeline e Farm. Modello dei costi. |
| 3 | Parallelismo a livello di istruzioni: architettura della CPU pipeline e implementazione delle unita' della CPU pipeline. |
| 3 | Architetture multiprocessor, reti di interconnessione, classi di architetture multiprocessor, organizzazione della memoria, |