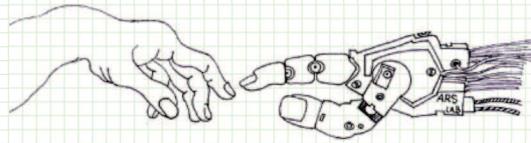


Guida rapida all'insegnamento

Sistemi dedicati

Docente: Giuseppe Scollo

Università di Catania
Dipartimento di Matematica e Informatica
Corso di Laurea Magistrale in Informatica, AA 2019-20



Indice

1. Guida rapida all'insegnamento
2. Obiettivi formativi
3. Organizzazione dell'insegnamento
4. Attività formative
5. Modalità di valutazione
6. Programma delle lezioni
7. Programma delle esercitazioni
8. Testi consigliati
9. Testi per consultazione
10. Laboratorio
11. Interazioni

Obiettivi formativi

Acquisizione e sviluppo della capacità di:

- modellazione, progettazione e realizzazione ottimale di sistemi di elaborazione dedicati a una specifica applicazione
- uso di strumenti hardware e software, quali piattaforme di sviluppo di sistemi dedicati, per progettare e realizzare sistemi dedicati ad applicazioni con alte prestazioni

Organizzazione dell'insegnamento

L'organizzazione dell'insegnamento prevede 24 ore di lezioni e 24 ore di esercitazioni di laboratorio di codesign (aula 24, lunedì e mercoledì, h. 15:00-17:00)

L'acquisizione di concetti e metodi nella disciplina è sostenuta da:

- frequenza delle lezioni ed esercitazioni
- partecipazione alle esperienze di laboratorio
- studio di testi di riferimento
- consultazione di altri testi e materiali didattici
- sperimentazione con strumenti software e piattaforme di sviluppo di sistemi dedicati
- elaborazione di soluzioni a problemi ed esercizi proposti: v. area **Esercizi**
- stesura di relazioni sulle esperienze di laboratorio: nella stessa area
- interazione con il docente, ricevimento settimanale:
G. Scollo: studio 325 (I blocco, II piano) [tel.]cit.: [095738]3007
lunedì e mercoledì, h. 13:00-15:00
- interazione collaborativa con colleghi e docente

Attività formative

Lezioni: lo studio dei testi di riferimento pone le basi metodologiche per applicare efficacemente un approccio trasversale nelle tecnologie e unitario nel risultato:

progettare in modo versatile sistemi con cifra di merito ottimale dedicati a svolgere compiti ad alto grado di specializzazione

Esercizi: a partire dalla specifica delle funzionalità astratte del sistema, il primo problema che spesso si pone è quello di selezionare l'architettura in cui tradurle, per poi procedere alla sintesi di tutti i componenti: hardware, software e interfacce di comunicazione. Gli esercizi proposti affrontano le diverse parti di questo processo.

Laboratorio di codesign: è previsto l'uso di sistemi di sviluppo per la realizzazione di applicazioni embedded, spaziando dalla configurazione su board alla sintesi di componenti con FPGA, fino alla realizzazione dell'intero sistema su un singolo chip. La stesura di relazioni sulle esperienze di laboratorio può essere prodotta da collaborazioni di gruppo.

Seminari: a titolo sperimentale, alcune lezioni (max. 1/5 del totale) sono preparate e prodotte in forma seminariale da studenti; la pianificazione dei seminari è condotta nell'ambito di una esercitazione di laboratorio.

Modalità di valutazione

Esame orale, progetto (opzionale)

- obiettivi della valutazione
- esame orale:
valutazione del conseguimento degli obiettivi formativi
- progetto opzionale:
valutazione della maturità concettuale e scientifica nella pratica della disciplina
- colloqui orali
- verifica dell'apprendimento critico della disciplina
- valutazione del contributo a relazioni sulle esperienze di laboratorio
- eventuale colloquio (opzionale) individuale sull'uso di concetti e metodi della disciplina in un progetto originale di laboratorio, concordato con il docente, che può essere realizzato in collaborazione con altri studenti
- calendario di esami

Il superamento dell'esame porta all'acquisizione di 6 crediti.

Programma delle lezioni

legenda: r = letture di riferimento, s = letture supplementari, rn.# = nota integrativa di riferimento, sn.# = nota integrativa supplementare #

- 1. Finalità e organizzazione dello studio. Introduzione al codesign di sistemi dedicati**
L01: 16/10/2019, r: S.01(1.1.4-1.4.1.6); s: VG.01(1.1-1.4), BF.01
- 2. Architetture e progettazione di sistemi dedicati**
L02: 22-23/10/2019, r: S.01(1.5.1.7), Z.01(1.1); s: VG.01(1.5-1.6)
- 3. Modelli dataflow, flusso del controllo**
L03: 28/10/2019, r: S.02; s: L5.06(6.3), M.02(2.5), sn.3, sn.4
- 4. Realizzazioni software di modelli dataflow**
L04: 04-06/11/2019, r: S.03(3.1); s: S.04
- 5. Sistemi sincroni come macchine a stati finiti con datapath (FSMD)**
L05: 11/11/2019, r: S.05(5.3-5.4.3.5.6); s: S.05(5.7)
- 6. Microprogrammazione: architetture, controllo, interpreti microprogrammati**
L06: 18/11/2019, r: S.06(6.1-6.4); s: S.06(6.6-6.8)
- 7. Progetto e analisi di programmi per sistemi dedicati**
L07: 25/11/2019, r: S.07(7.1.7.3); s: S.07(7.2.7.5), sn.7
- 8. Progettazione di System-on-Chip (SoC)**
L08: 02/12/2019, r: S.08(8.1-8.3); s: S.08(8.4), R.01
- 9. Comunicazione HW/SW, sistemi di bus on-chip**
L09: 09/12/2019, r: S.09(9.1-9.4), S.10(10.1); s: sn.11, S.10(10.2-10.4)
- 10. Interfacce di microprocessore (seminario studente)**
L10: 10/01/2020, r: S11(11.1.1-11.1.5,11.2.0,11.3.0-11.3.1,11.3.3); s: S11(11.1.6,11.2.1-11.2.2,11.3.2,11.3.4)
- 11. Interfacce hardware**
L11: 13/01/2020, r: S.12(12.1-12.3,1,12.4); s: S.12(12.3.2)
- 12. Progetto di un coprocessore multicore mappato in memoria**
L12: 20/01/2019, progetto di laboratorio

Programma delle esercitazioni

legenda: r = letture di riferimento, s = letture supplementari, rn.# = nota integrativa di riferimento, sn.# = nota integrativa supplementare #

- 1. Introduzione all'uso combinato di Gezel con un simulatore VHDL**
E01: 22/10/2019, r: S.01(1.1.1), rn.1, rn.2; s: S.A(A.1)
- 2. Introduzione al progetto di sistemi hardware con FPGA**
E02: 23-28/10/2019, r: Z.01(1.3), rn.3; s: rn.4
- 3. Linguaggi di descrizione dell'hardware: Gezel, VHDL, Verilog, SystemC**
E03: 30/10/2019, r: S.05(5.1-5.2), Z.03(3.1-3.7); s: BF.aB, Z.aB, M.2(2.7), sn.1, sn.2
- 4. Esempi di reti combinatorie in VHDL**
E04: 04/11/2019, r: Z.04(4.1-4.6); s: rn.6(App.A)
- 5. Esempi di reti sequenziali in VHDL, realizzazioni hardware di modelli dataflow**
E05: 06-11/11/2019, r: Z.06(6.1-6.5.1), S.03(3.2); s: S.03(3.3)
- 6. Esempi di FSMD in Gezel e in VHDL. Pianificazione di seminari degli studenti**
E06: 13/11/2019, r: S.05(5.4.4-5.5); s: Z.07(7.1-7.2), sn.5, sn.6
- 7. Esempio di progetto di un microprocessore in Gezel e VHDL**
E07: 22/11/2019, r: S.06(6.5); s: Z.07(7.3-7.5)
- 8. Strumenti ed esempi di analisi di programmi**
E08: 25/11/2019, r: S.07(7.4); s: S.07(7.5,7.6), sn.7, sn.8, sn.9, sn.10
- 9. Sviluppo di componenti su FPGA**
E09: 06/12/2019, r: rn.5, rn.6
- 10. Sviluppo di SoC su FPGA con profiling dell'applicazione**
E10: 13/12/2019, r: rn.7; s: sn.12
- 11. Realizzazione su FPGA di un coprocessore mappato in memoria**
E11: 17/01/2020, r: progetto di laboratorio
- 12. Realizzazione su FPGA di un coprocessore multicore mappato in memoria**
E12: 24/01/2020, progetto di laboratorio

Testi consigliati

Testi di riferimento

N.B. Le parti dei testi raccomandate sono indicate nel programma, per ciascuna lezione, nella forma abbreviata A.C(S), dove: A = iniziali degli autori del testo, C = capitolo, S = sezione/i

P.R. Schaumont: *A Practical Introduction to Hardware/Software Codesign*
2nd Edition. Springer (2012)

con correzioni al testo prodotte a partire dall'edizione 2013-2014 dell'insegnamento

M. Zwoliński: *Digital System Design With VHDL*
2nd Edition, Pearson (2004)

Note integrative di riferimento

1. *Quartus Prime Introduction Using VHDL Designs - For Quartus Prime 16.1*, Intel Corp. - FPGA University Program, November 2016
2. *Using TimeQuest Timing Analyzer - For Quartus Prime 16.1*, Intel Corp. - FPGA University Program, November 2016
3. *Quartus Prime Introduction Using Schematic Designs - For Quartus Prime 16.1*, Intel Corp. - FPGA University Program, November 2016
4. *Using Library Modules in VHDL Designs - For Quartus Prime 16.1*, Intel Corp. - FPGA University Program, November 2016
5. *Introduction to the Qsys System Integration Tool - For Quartus Prime 16.1*, Intel Corp. - FPGA University Program, November 2016
6. *Making Qsys Components - For Quartus Prime 16.1*, Intel Corp. - FPGA University Program, November 2016
7. *Performance Counter Unit Core, Ch. 36 in: Embedded Peripherals IP User Guide*, Intel Corp., UG-01085 | 2019.12.16

Testi per consultazione

Libri di testo

C. Brandolese, W. Fornaciari: *Sistemi embedded: sviluppo hardware e software per sistemi dedicati*
Pearson, Milano (2007)

E.A. Lee & S.A. Seshia: *Introduction to Embedded Systems - A Cyber-Physical Systems Approach*
2nd Ed., MIT Press (2017)

P. Marwedel: *Embedded System Design: Embedded Systems Foundations of Cyber-Physical Systems*
2nd Edition. Springer (2011)

C. Rowen: *Engineering the Complex SOC - Fast, Flexible Design with Configurable Processors*
Prentice-Hall (2004)

F. Vahid & T. Givargis: *Embedded System Design: A Unified Hardware/Software Introduction*, Wiley (2002)

M. Wolf: *Computers as components: Principles of embedded computing system design*
3rd Edition, Morgan Kaufmann (2012)

Note integrative supplementari

1. F. Vahid (2006): Digital Design, Chapter 9, Hardware Description Languages (PDF slides)
2. D.J. Smith (1996): VHDL & Verilog Compared & Contrasted
3. Ghamarian et al. (2007): Latency minimization for Synchronous Data Flow Graphs
4. Stuijk et al. (2006): Exploring trade-offs in buffer requirements and throughput constraints for Synchronous Dataflow Graphs
5. C. Brandolese, *Introduzione al linguaggio VHDL*, Politecnico di Milano
6. *Note sul VHDL*, Corso di Architettura dei Sistemi Integrati
7. *Introduction to the ARM® Processor Using Intel FPGA Toolchain - For Quartus Prime 16.1*, Intel Corp. - FPGA University Program, November 2016
8. *Introduction to the Intel Nios II Soft Processor - For Quartus Prime 16.1*, Intel Corp. - FPGA University Program, November 2016
9. *Intel FPGA Monitor Program Tutorial for ARM - For Quartus Prime 16.1*, Intel Corp. - FPGA University Program, November 2016
10. *Intel FPGA Monitor Program Tutorial for Nios II - For Quartus Prime 16.1*, Intel Corp. - FPGA University Program, November 2016
11. *Avalon® Interface Specifications, MNL-AN48USREF*, Intel Corp., 2019.10.08
12. *Profiling Nios II Systems, AN-391-3.0*, Altera Corp., July 2011

Le attività di laboratorio consistono di una serie di esperienze sui seguenti argomenti:

- codesign e cosimulazione in Gezel e VHDL
- ottimizzazione di programmi nei linguaggi C e assembly
- programmazione di FPGA in VHDL
- codesign, simulazione e valutazione delle prestazioni su sistemi di sviluppo SoC con FPGA

Forum, Moodle, Galileo: cosa va dove?

- Forum: discussioni di
 - organizzazione dell'insegnamento, avvisi, FAQ
 - problemi con l'uso di servizi on-line, strumenti software ecc.
 - proposte di idee di progetti di sistemi dedicati
- Moodle (servizi ad accesso riservato):
 - accesso a materiali didattici di supporto
 - sviluppo di problemi ed esercizi proposti
 - discussioni di tematiche pertinenti a lezioni, esperienze di laboratorio e materiali didattici
 - collaborazione di gruppo, consegna di relazioni sulle esperienze di laboratorio
 - discussioni di proposte di progetti e del loro sviluppo
 - segnalazione e discussione di errori nei materiali didattici (possono valere punti *bonus*!)
- Galileo:
 - sviluppo di progetti hardware/software con documentazione e distribuzione dei risultati nel pubblico dominio