

# Introduzione al Progetto di Sistemi Digitali

Giuliano Donzellini · Luca Oneto  
Domenico Ponta · Davide Anguita

# Introduzione al Progetto di Sistemi Digitali

Giuliano Donzellini  
Dipartimento di Ingegneria Navale, Elettrica,  
Elettronica e delle Telecomunicazioni  
(DITEN)

Università degli Studi di Genova  
Genoa  
Italy

Luca Oneto  
Dipartimento di Informatica, Bioingegneria,  
Robotica e Ingegneria dei Sistemi  
(DIBRIS)

Università degli Studi di Genova  
Genoa  
Italy

Domenico Ponta  
Università degli Studi di Genova  
Genoa  
Italy

Davide Anguita  
Dipartimento di Informatica, Bioingegneria,  
Robotica e Ingegneria dei Sistemi  
(DIBRIS)

Università degli Studi di Genova  
Genoa  
Italy

ISBN 978-88-470-3962-9      ISBN 978-88-470-3963-6 (eBook)  
<https://doi.org/10.1007/978-88-470-3963-6>

Library of Congress Control Number: 2017956313

© Springer-Verlag Italia S.r.l. 2018

Quest'opera è protetta dalla legge sul diritto d'autore e la sua riproduzione è ammessa solo ed esclusivamente nei limiti stabiliti dalla stessa. Le fotocopie per uso personale possono essere effettuate nei limiti del 15 % di ciascun volume dietro pagamento alla SIAE del compenso previsto dall'art. 68. Le riproduzioni per uso non personale e/o oltre il limite del 15 % potranno avvenire solo a seguito di specifica autorizzazione rilasciata da AIDRO, Corso di Porta Romana n.108, Milano 20122, e-mail [segreteria@aidro.org](mailto:segreteria@aidro.org) e sito web [www.aidro.org](http://www.aidro.org). Tutti i diritti, in particolare quelli relativi alla traduzione, alla ristampa, all'utilizzo di illustrazioni e tabelle, alla citazione orale, alla trasmissione radiofonica o televisiva, alla registrazione su microfilm o in database, o alla riproduzione in qualsiasi altra forma (stampata o elettronica) rimangono riservati anche nel caso di utilizzo parziale. La violazione delle norme comporta le sanzioni previste dalla legge. L'utilizzo in questa pubblicazione di denominazioni generiche, nomi commerciali, marchi registrati ecc., anche se non specificatamente identificati, non implica che tali denominazioni o marchi non siano protetti dalle relative leggi e regolamenti.

Printed on acid-free paper

This Springer imprint is published by Springer Nature  
The registered company is Springer-Verlag Italia S.r.l.  
The registered company address is: Via Decembrio 28, 20137 Milano, Italy

*A mia moglie Melina e ai miei figli Sara e Paolo,  
che mi hanno aiutato con il loro affetto in questo lavoro,  
concedendomi tutto il tempo necessario,  
spesso sottratto alle loro necessità.  
A mio Padre e mia Madre, che non ho fatto in tempo  
a ringraziare come avrei voluto.*

Giuliano Donzellini

*A mio Padre e mia Madre per il mio essere spesso lontano.  
A Federica per essermi sempre vicina.*

Luca Oneto

*A mia moglie Luisa per tutto il tempo che mi ha dedicato ed il  
suo costante sostegno ed incoraggiamento di tutta una vita.  
Alla memoria di Sandro Chiabrera, maestro e amico.*

Domenico Ponta

*Ai miei genitori e nonni che mi hanno sopportato da bambino.  
A mio marito che mi sopporta da adulto.*

Davide Anguita

---

## Prefazione del Prof. Filippo Sorbello

Con vero piacere presento il libro “Introduzione al progetto dei sistemi digitali” degli amici e Colleghi Donzellini, Oneto, Ponta e Anguita. Il testo è rivolto agli allievi di un primo corso di reti logiche e tratta i sistemi digitali partendo dalle basi teoriche, scelte ed approfondite al giusto livello, fino ad arrivare all’analisi ed alla sintesi delle reti combinatorie e delle reti sequenziali. Numerosi sono gli esempi e gli esercizi proposti per i quali viene data la soluzione.

L’evoluzione delle tecnologie elettroniche ha portato a una sempre più ampia diffusione dei sistemi digitali in ogni ambito della vita quotidiana. La velocità, la densità e la complessità degli attuali circuiti digitali sono stati resi possibili dalle metodologie automatiche di progetto e dal progresso della tecnologia.

La conoscenza delle basi teoriche delle reti logiche è comunque indispensabile per una padronanza delle architetture dei sistemi digitali di qualunque complessità ed anche per un corretto uso degli strumenti di progettazione automatica basati sui linguaggi per la descrizione dell’hardware HDL.

Gli allievi dei corsi iniziali non possiedono adeguate capacità di programmazione e di astrazione e neanche le conoscenze di fisica ed elettronica necessarie per utilizzare in modo adeguato i linguaggi HDL. Nel testo questa difficoltà viene superata avvalendosi di uno strumento di simulazione (*Deeds*), sviluppato da uno degli autori, che utilizza una interfaccia di uso immediato. *Deeds* viene usato per simulare il comportamento sia dei circuiti proposti nel testo sia dei circuiti che in autonomia, ciascuno di loro, vuole verificare dopo averli progettati. I progetti messi a punto con *Deeds* si possono poi esportare in linguaggio HDL o provare sui circuiti FPGA.

L’uso dei linguaggi per la descrizione dell’hardware unito alla conoscenza delle basi teoriche dei circuiti logici costituiscono il bagaglio di conoscenze indispensabili per entrare nel mondo digitale.

## VIII

Ritengo che il testo sia un ottimo strumento per affrontare questa sfida vista la capacità che gli Autori hanno usato nel trasferire le necessarie conoscenze teoriche e professionali in un testo chiaro nei contenuti, scorrevole nella forma, gradevole nell'aspetto.

Palermo (Italia), 2 settembre 2017

*Filippo Sorbello*

---

## Prefazione del Prof. Mauro Olivieri

Il libro di testo scritto da Giuliano Donzellini, Luca Oneto, Domenico Ponta e Davide Anguita si caratterizza per due note distintive nel vasto panorama dei testi universitari introduttivi alla progettazione digitale.

La prima nota distintiva è la sua focalizzazione su un ben definito insieme di nozioni e strumenti che costituiscono il fondamento dei sistemi digitali, ovvero la sintesi logica combinatoria e sequenziale e gli argomenti strettamente correlati ad essa. Il libro non si estende ai circuiti né ai sistemi a microprocessore, e rimanendo in tali confini si permette grande chiarezza, precisione, completezza ed autoconsistenza di sicuro giovamento per chi apprende. L'ispezione visiva dei numerosi schemi grafici e diagrammi a corredo del testo rende immediatamente al lettore questa impressione. La presenza di esercizi risolti insieme al software per la simulazione *Deeds*, inoltre, è un elemento decisivo giustamente sempre richiesto dagli studenti.

La seconda nota distintiva è l'equilibrio fra l'impianto teorico e il riferimento all'implementazione pratica, che consente un reale e solido apprendimento. Sebbene nel testo non si citi mai una volta la parola "tensione", né altre grandezze elettriche, lo studente che si serve del libro ha sempre presente che sta studiando la formalizzazione di un sistema elettronico. Allo stesso tempo, il testo non si abbandona alla facile tentazione di presentare la materia come una serie di esempi pratici di progetto senza basi teoriche. Una particolarità, questa, tipica di quella che definirei "scuola genovese" di elettronica digitale, della quale mi considero un esponente esterno.

Per le suddette motivazioni, il libro di Donzellini, Oneto, Ponta e Anguita è un testo prezioso per lo studente che si accinge a comprendere a fondo i concetti alla base del grande mondo della progettazione elettronica digitale.

---

## Prefazione degli autori

La grande e continuamente crescente complessità dei sistemi digitali di oggi pone pesanti richieste ai sistemi educativi che hanno il compito di formare le nuove generazioni di progettisti o anche solo di fornire una solida conoscenza di base sul digitale. Le istituzioni accademiche fanno fatica a tenere il passo con l'avanzamento delle tecnologie ed è compito di chi, come gli autori di questo lavoro, è incaricato della formazione a livello introduttivo ed intermedio, di affrontare il problema e compiere scelte.

È certamente ovvio che un progettista digitale deve essere addestrato all'uso di linguaggi per la descrizione dell'hardware (HDL) ed è ormai pratica corrente introdurli molto presto nei corsi, affiancando o sostituendo l'approccio tradizionale basato su componenti e schemi. La scelta di descrivere i sistemi digitali tramite HDL permette anche l'adozione di FPGA (Field Programmable Gate Arrays) per la realizzazione pratica del progetto, usando schede prototipo fornite dai produttori dei chip.

Tuttavia, a nostro parere, l'adozione di un HDL in un primo corso di reti logiche con risorse ridotte in termini di crediti (come nel nostro caso), presenta problemi. Crediamo che, sostituendo componenti logici e schemi con un HDL, non sia facile costruire, nello spazio di un corso introduttivo, una buona comprensione dei fondamenti dei circuiti digitali, in quanto sono richiesti un livello di astrazione ed una familiarità con la programmazione che gli studenti del primo anno generalmente non possiedono ancora.

Inoltre, l'impiego del software di simulazione e sintesi fornito dai produttori dei chip FPGA, anche a livello di base, pone ulteriori problemi. Tools sviluppati per il progettista di sistemi digitali non soddisfano necessariamente le esigenze della formazione: il loro uso non è immediato da parte degli studenti, che possono imparare ad usarli in modo parziale e meccanico, con il rischio che, in questo modo, tralascino importanti concetti base, nascosti sotto i tecnicismi degli HDL e degli strumenti di lavoro.



Si tratta quindi di fare in modo che gli studenti acquisiscano una solida base sulla quale costruire le loro capacità progettuali e, allo stesso tempo, adattarsi al passo veloce dell'innovazione tecnologica e quindi acquistare dimestichezza con linguaggi e strumenti di progetto.

Per queste ragioni il libro mantiene un approccio tradizionale alle reti logiche, descritte e progettate per mezzo di simboli e schemi, pur tenendo conto dello stato dell'arte di oggi nella scelta degli argomenti e, soprattutto, a livello di esercizi e progetti. Questa caratteristica ne permette un impiego ottimale nei corsi di studio che non prevedono approfondimenti nel progetto delle reti, mentre fornisce una solida base per chi andrà avanti in questa direzione.

Il libro è autosufficiente, in quanto contiene, oltre alla parte teorica, un gran numero di esempi e di esercizi, completi di soluzioni. In corsi in cui ci sia spazio per attività di laboratorio si può sfruttare, con notevoli vantaggi didattici, la simbiosi con uno strumento di simulazione, *Deeds (Digital Electronics Education and Design Suite)*, sviluppato nel corso degli anni da uno degli autori (Giuliano Donzellini), con lo scopo preciso di supportare apprendimento e attività di laboratorio per i nostri studenti dell'ingegneria dell'informazione. La stretta connessione con *Deeds* rappresenta un importante elemento di forza e di originalità del nostro lavoro, in quanto tutti gli schemi, esempi e esercizi di progetto, dal più semplice al più complesso, presentati in questo libro, sono stati creati con *Deeds* e sono disponibili online per l'immediata simulazione.

L'ambiente *Deeds* copre tutti i principali aspetti del progetto di sistemi digitali, dalle reti combinatorie e sequenziali alle macchine a stati finiti e i sistemi a microprocessore “*embedded*”, permettendo il progetto e la simulazione di reti abbastanza complesse contenenti logiche standard, macchine a stati finiti, componenti definiti dall'utente e microprocessori, inclusa la loro programmazione in *linguaggio assembly*.

*Deeds* è stato sviluppato con in mente una estrema facilità d'uso, insieme a caratteristiche quasi professionali. Le differenze principali tra *Deeds* e un simulatore professionale sono la semplicità ed immediatezza dell'interfaccia con l'utente e la disponibilità di una vasta collezione di materiale didattico e progetti. *Deeds* è un sistema “vivo” in continua evoluzione: aggiornamenti sono disponibili periodicamente per migliorare gli strumenti esistenti ed aggiungerne di nuovi. Lo stesso vale per il materiale didattico.

La transizione verso i dispositivi FPGA avviene grazie alla possibilità di esportare un intero progetto creato e simulato con *Deeds* verso uno strumento professionale e di provarlo realmente in hardware. *Deeds* consente di evitare la complessità dell'intero processo che normalmente è necessario sul software specifico professionale; permette ad uno studente alle prime armi di non dover scrivere codice HDL, che è generato automaticamente da *Deeds*. Il vasto repertorio del materiale di apprendimento di *Deeds* è quindi ridiretto verso la realizzazione con FPGA senza bisogno di sostanziali modifiche.

Tuttavia, a valle della generazione automatica del codice HDL da parte di *Deeds*, gli studenti avanzati possono interagire direttamente con gli strumenti specifici FPGA, avendo quindi la possibilità di osservare, modificare e riusare il codice HDL (VHDL nel nostro caso), compiendo così una vera graduale transizione verso le tecniche di progettazione moderne.

## Obiettivi didattici

Secondo l'esperienza degli autori l'intero contenuto del libro, accompagnato da esercitazioni di progetto e simulazione basate su *Deeds*, può essere usato in un corso introduttivo ai sistemi digitali di almeno 9 crediti.

Qui sotto riportiamo in modo sintetico i contenuti dei capitoli, indicando in corsivo gli argomenti che possono essere tralasciati senza perdere continuità con il progetto didattico, in corsi con un numero inferiore di crediti:

1. Algebra booleana e reti combinatorie
  - Trattazione classica che non richiede conoscenze preliminari.  
*Si possono tralasciare i teoremi di Shannon.*
2. Progetto di reti combinatorie
  - Sintesi e minimizzazione con le mappe di Karnaugh.
  - Reti combinatorie standard.
  - Ritardi di propagazione.  
*Le mappe con variabili riportate e le alee possono essere omesse.*
3. Aritmetica binaria
  - Trattazione classica.
  - Reti aritmetiche.  
*La trattazione dei numeri negativi in binario può essere omessa, come pure l'aritmetica BCD.*
4. Complementi sul progetto di reti combinatorie
  - Minimizzazione di espressioni con il metodo Quine-McCluskey.  
*L'intero capitolo può essere omissso.*
5. Introduzione alle reti sequenziali
  - Transizione intuitiva dalle reti combinatorie alle sequenziali.
  - Struttura e funzionamento dei principali tipi di flip-flop.
  - Caratteristiche dinamiche dei flip-flop.  
*Si possono trattare soltanto i tipi logici "D" ed "E", senza entrare in dettagli circuitali.*

## 6. Reti sincrone di flip-flop

- Introduzione alle reti sincrone di flip-flop.
- Reti sequenziali: registri e contatori.
- Tecniche di analisi temporale delle reti sincrone.

*La trattazione dei contatori e dei registri può essere ridotta, come pure l'analisi temporale delle reti sequenziali.*

## 7. Reti sequenziali come Macchine a Stati Finiti

- Progetto della MSF, realizzata tramite i diagrammi ASM.
- Esercizi risolti di diagrammi ASM.
- Sintesi della MSF con tabelle di stato e mappe.

*La sintesi della MSF può essere ridotta, omettendo le mappe a variabili riportate, o tralasciata del tutto.*

## 8. La Macchina a Stati Finiti come controllore di sistema.

- Progettazione di sistemi controllore - datapath.
- Esercizi risolti sui sistemi controllore - datapath.

*Il capitolo rappresenta l'obiettivo ultimo del corso: la trattazione ed il livello degli esercizi possono essere adattati alle esigenze e ai limiti della situazione didattica.*

## Come usare al meglio il libro

La strettissima connessione di questo libro con l'ambiente *Deeds* consiglia di usarlo in simbiosi con gli strumenti di simulazione, sia per verificare e confermare in modo attivo i concetti e le procedure esposte nel libro, sia per avere un supporto per la soluzione degli esercizi e il progetto dei sistemi.

Questa pratica “learning by doing” permette di costruire progressivamente le capacità di analisi e di progetto che costituiscono l'obiettivo da raggiungere.

---

## Materiale digitale di supporto al libro

Questo libro di testo alterna parti teoriche, esempi, esercizi e soluzioni. Tutti gli esempi e gli esercizi presentati nel libro sono stati realizzati con il simulatore *Deeds* reperibile al link:

<https://www.digitalelectronicsdeeds.com>

Nel sito sono descritte le caratteristiche del simulatore e le istruzioni su come scaricarlo e utilizzarlo (in ambiente PC Windows). L'uso del simulatore è *in locale* e non richiede il costante collegamento in rete.

Nello stesso sito abbiamo reso disponibili, come materiale addizionale al libro, gli schemi *Deeds* relativi alla quasi totalità delle figure ed esempi proposti. È inoltre disponibile tutto il materiale necessario allo svolgimento degli esercizi e alla verifica del comportamento delle soluzioni mediante *Deeds*.

Il materiale è stato ordinato seguendo la denominazione dei capitoli, delle sezioni e delle sottosezioni del libro stesso, in modo da facilitarne la fruizione. Nello stesso sito saranno resi disponibili eventuali futuri aggiornamenti, correzioni, e miglioramenti del libro.

---

# Indice

<b>Prefazione del Prof. Filippo Sorbello</b> .....	VII
<b>Prefazione del Prof. Mauro Olivieri</b> .....	IX
<b>Prefazione degli autori</b> .....	XI
<b>Materiale digitale di supporto al libro</b> .....	XV
<b>Indice</b> .....	XVII
<b>1 Algebra booleana e reti combinatorie</b> .....	1
1.1 Grandezze analogiche e logiche .....	1
1.2 Variabili booleane .....	4
1.3 Funzioni booleane .....	5
1.4 Tabelle di verità.....	5
1.5 Definizione dell'algebra booleana.....	6
1.6 Proprietà fondamentali dell'algebra booleana.....	7
1.7 Altre operazioni .....	10
1.8 Insiemi di operazioni funzionalmente completi.....	12
1.9 Teorema di espansione di Shannon.....	15
1.10 Livello di un'espressione booleana .....	18
1.11 Letterali.....	18
1.12 Prodotti fondamentali.....	18
1.13 Somme fondamentali.....	19
1.14 Implicanti .....	19
1.15 Implicanti primi.....	19
1.16 Reti combinatorie .....	20
1.16.1 Esempio: analisi di rete logica.....	20
1.16.2 Esempio: analisi di rete logica a due livelli .....	20
1.16.3 Esempio: schema circuitale di una rete logica (1) .....	21
1.16.4 Esempio: schema circuitale di una rete logica (2) .....	21

1.16.5	Esempio: definizione del comportamento di una rete logica .....	22
1.16.6	Esempio: schema circuitale da tabella di verità .....	23
1.16.7	Esempio: controllo di un impianto di riscaldamento ....	23
1.16.8	Esempio: selettore (multiplexer) a due canali .....	25
1.17	Esercizi .....	28
1.18	Soluzioni .....	30
<b>2</b>	<b>Progetto di reti combinatorie .....</b>	<b>33</b>
2.1	Mappe di Karnaugh .....	33
2.2	Sintesi AND-OR con uso delle mappe .....	36
2.2.1	Implicanti e implicanti primi nelle mappe .....	38
2.2.2	Minimizzazione con l'uso delle mappe .....	40
2.2.3	Mappe "a scacchiera" .....	41
2.2.4	Esempi di sintesi AND-OR .....	41
2.3	Sintesi OR-AND .....	43
2.3.1	Sintesi della funzione negata .....	44
2.4	Sintesi NAND-NAND .....	45
2.5	Sintesi NOR-NOR .....	46
2.6	Reti combinatorie standard .....	46
2.6.1	Decodificatore (Decoder) .....	46
2.6.2	Selettore (multiplexer) .....	50
2.6.3	Deselettore (demultiplexer) .....	52
2.6.4	Decodificatore per display a sette segmenti .....	54
2.6.5	Decodificatore BCD - 7 segmenti (con l'impiego delle indifferenze) .....	56
2.6.6	Uso di selettori per la sintesi di reti combinatorie .....	57
2.7	Mappe con variabili riportate .....	59
2.7.1	Sintesi di una mappa con variabili riportate .....	59
2.7.2	Le variabili riportate e i teoremi di espansione .....	62
2.8	Comportamento nel tempo dei circuiti logici .....	63
2.8.1	Tempi Caratteristici .....	63
2.8.2	Alee .....	65
2.8.3	Eliminazione delle alee statiche .....	67
2.8.4	Note sulla eliminazione delle alee .....	69
2.9	Esercizi .....	70
2.9.1	Mappe .....	70
2.9.2	Alee .....	72
2.10	Soluzioni .....	73
2.10.1	Mappe .....	73
2.10.2	Alee .....	76

<b>3</b>	<b>Aritmetica binaria</b>	79
3.1	Informazione binaria	79
3.2	Numerazione binaria (BIN)	80
3.2.1	Conversione da numerazione binaria a decimale	80
3.2.2	Conversione da numerazione decimale a binaria	81
3.2.3	Massimo numero rappresentabile	82
3.3	Numerazione ottale (OCT)	82
3.4	Numerazione esadecimale (HEX)	83
3.5	Generalità sui codici binari	85
3.6	Aritmetica binaria	86
3.6.1	Somma	86
3.6.2	Sottrazione	88
3.6.3	Prodotto	89
3.7	Aritmetica BCD 8421	89
3.8	Numeri razionali in binario	90
3.9	Reti aritmetiche	91
3.9.1	Semi-sommatore (half adder)	91
3.9.2	Sommatore (full adder)	91
3.9.3	Sommatore con riporto in cascata	93
3.9.4	Unità aritmetico-logica (ALU, Arithmetic Logic Unit)	93
3.10	Numeri relativi in binario	95
3.10.1	Rappresentazione in codice “modulo e segno”	95
3.10.2	Complementazione	96
3.10.3	Rappresentazione in codice “complemento a uno”	99
3.10.4	Rappresentazione in codice “complemento a due”	100
3.10.5	Estensione del segno	101
3.11	Rappresentazione di numeri reali	102
3.12	Codici alfanumerici	103
3.13	Codici a rivelazione di errore: generatore e rivelatore di parità	105
3.14	Esercizi	108
3.14.1	Numerazione Binaria	108
3.14.2	Numeri binari con segno	108
3.14.3	Numerazione Ottale e Esadecimale	109
3.15	Soluzioni	110
3.15.1	Numerazione Binaria	110
3.15.2	Numeri binari con segno	111
3.15.3	Numerazione Ottale e Esadecimale	112
<b>4</b>	<b>Complementi sul progetto di reti combinatorie</b>	115
4.1	Minimizzazione di espressioni booleane con il metodo di Quine-McCluskey	115
4.1.1	Fase di espansione	116
4.1.2	Fase di copertura	119
4.1.3	Funzioni non completamente specificate	120
4.1.4	Ottimizzazione della fase di copertura	121

4.1.5	Ottimizzazione contemporanea di più funzioni.....	126
4.2	Esercizi .....	131
4.2.1	Quine-McCluskey: sintesi di singola funzione .....	131
4.2.2	Quine-McCluskey: sintesi congiunta di più funzioni ....	134
4.3	Soluzioni .....	137
4.3.1	Quine-McCluskey: sintesi di singola funzione .....	137
4.3.2	Quine-McCluskey: sintesi congiunta di più funzioni ....	139
<b>5</b>	<b>Introduzione alle reti sequenziali.....</b>	<b>141</b>
5.1	Dalle reti combinatorie alle reti sequenziali .....	141
5.1.1	Esempio introduttivo .....	142
5.1.2	Memorizzare un bit di informazione: il flip-flop .....	143
5.1.3	Tipi logici e tipi di comando dei flip-flop .....	145
5.2	Flip-flop a comando diretto .....	145
5.2.1	Flip-flop SR (a comando diretto) .....	146
5.2.2	Flip-flop D (a comando diretto) .....	150
5.2.3	Flip-flop JK (a comando diretto) .....	150
5.3	Inizializzazione di una rete sequenziale .....	152
5.3.1	Ingressi di inizializzazione dei flip-flop .....	153
5.3.2	Generazione del segnale di inizializzazione .....	155
5.4	Flip-flop a comando abilitato a livello .....	156
5.4.1	Flip-flop SR con abilitazione a livello (SR-Latch) ....	156
5.4.2	Flip-flop D con abilitazione a livello ( <i>D-Latch</i> ) .....	158
5.4.3	Flip-flop JK con abilitazione a livello (JK-Latch) ....	159
5.5	Sincronizzazione delle reti sequenziali .....	160
5.5.1	Il segnale di sincronizzazione .....	160
5.5.2	Comando impulsivo dei flip-flop abilitati a livello ....	161
5.5.3	Il “Clock” e il “comando abilitato sul fronte” .....	163
5.5.4	La struttura “master-slave” .....	164
5.6	Flip-flop a comando abilitato sul fronte.....	166
5.6.1	Flip-flop D-PET .....	166
5.6.2	Flip-flop E-PET .....	170
5.6.3	Flip-flop JK-PET .....	172
5.6.4	Flip-flop T-PET.....	174
5.6.5	Inizializzazione sincrona dei flip-flop .....	174
5.7	Tempi caratteristici dei flip-flop .....	175
5.7.1	Relazione tra i tempi di propagazione e di mantenimento	176
5.7.2	Massima frequenza del clock di una rete con flip-flop ..	176
5.8	Flip-flop: simboli grafici e tabelle .....	178
5.8.1	Tipi logici .....	178
5.8.2	Tipi di comando .....	179
5.8.3	Tabelle di eccitazione .....	181
5.9	Esercizi .....	183
5.10	Soluzioni .....	186



<b>6</b>	<b>Reti sincrone di flip-flop</b>	189
6.1	Segnali sincroni e asincroni	191
6.1.1	Sincronizzatore	192
6.1.2	Sincronizzatore a più stadi	193
6.2	Registri	194
6.2.1	Registro parallelo	194
6.2.2	Registro a scorrimento	196
6.2.3	Registro a scorrimento con caricamento parallelo	199
6.2.4	Registro a scorrimento universale	202
6.3	Contatori	205
6.3.1	Contatore binario	205
6.3.2	Contatore con abilitazione	211
6.3.3	Contatore bidirezionale	213
6.3.4	Contatori “universali”	216
6.3.5	Contatori asincroni	219
6.4	Analisi di reti	221
6.4.1	Esempio n. 1	221
6.4.2	Esempio n. 2	224
6.4.3	Esempio n. 3	226
6.4.4	Esempio n. 4	228
6.4.5	Esempio n. 5	230
6.5	Esercizi	234
6.6	Soluzioni	254
<b>7</b>	<b>Reti sequenziali come Macchine a Stati Finiti</b>	265
7.1	Modello generale di Macchina a Stati Finiti	266
7.1.1	Macchine Sincrone e Asincrone	266
7.1.2	Macchine di Moore e di Mealy	267
7.1.3	Esempio di Macchina a Stati Finiti Sincrona	268
7.1.4	Equazioni generali dello stato successivo e delle uscite	269
7.2	Diagrammi ASM	271
7.2.1	Descrizione degli stati	271
7.2.2	Ingressi	275
7.2.3	Uscite condizionate	284
7.3	Esempi di costruzione di diagrammi ASM	288
7.3.1	Esempi introduttivi	288
7.3.2	Generatore di impulsi a rapporto pieno/vuoto regolabile	298
7.3.3	Riconoscitore di sequenza	303
7.3.4	Trasmettitore Seriale Sincrono (2 bit)	305
7.3.5	Ricevitore di comando in formato seriale sincrono	307
7.3.6	Ricevitore Seriale Sincrono (2 bit)	310
7.3.7	Gestione di pulsanti	313
7.3.8	Registro a Scorrimento (3 bit)	316
7.3.9	Rete sequenziale con uscita condizionata	319
7.3.10	Registro a scorrimento con albero di EXOR	321

7.4	Sintesi della MSF sincrona .....	323
7.4.1	Assegnazione degli stati .....	324
7.4.2	Descrizione della MSF mediante tabella degli stati .....	325
7.4.3	Sintesi dalla tabella degli stati .....	327
7.4.4	Esempi di sintesi di MSF sincrone .....	330
7.5	Comportamento nel tempo della MSF sincrona .....	344
7.6	Esercizi .....	350
7.6.1	Analisi di reti sequenziali in termini di MSF .....	350
7.6.2	Progetto di MSF a partire da specifiche testuali .....	353
7.7	Soluzioni .....	358
7.7.1	Analisi di reti sequenziali in termini di MSF .....	358
7.7.2	Progetto di MSF a partire da specifiche testuali .....	362
<b>8</b>	<b>La Macchina a Stati Finiti come controllore di sistema</b> ....	<b>371</b>
8.1	I sistemi digitali .....	371
8.2	Sistemi a controllo aperto .....	372
8.2.1	Ricevitore seriale (2 bit) .....	373
8.3	Sistemi a controllo retroazionato .....	377
8.3.1	Ricevitore e trasmettitore seriale (2 bit) .....	377
8.3.2	Generatore di impulso .....	380
8.3.3	Ricevitore seriale (8 bit) .....	385
8.3.4	Regolatore di Luminosità per Lampada .....	392
8.3.5	Serratura a combinazione .....	396
8.3.6	Distributore automatico di bevande .....	401
8.3.7	Progetto di generatore di onda quadra programmabile ..	405
8.3.8	Progetto di sistema per luminarie natalizie .....	409
8.4	Esercizi .....	415
8.4.1	Progetto di controllore, con datapath assegnato .....	415
8.4.2	Progetto di sistema completo (controllore e datapath) ..	434
8.5	Suggerimenti .....	437
8.5.1	Progetto di sistema completo (controllore e datapath) ..	437
8.6	Soluzioni .....	444
8.6.1	Progetto di controllore, con datapath assegnato .....	444
8.6.2	Progetto di sistema completo (controllore e datapath) ..	462