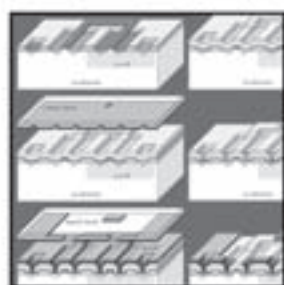


A. De Santis  
M. Cacciaglia  
C. Saggese

# sistemi2



Sistemi  
programmabili  
e analogici



CALDERINI

Proprietà letteraria riservata

© 2005 RCS Libri S.p.A., Milano

1<sup>a</sup> edizione: gennaio 2005

Fotocomposizione e riproduzioni fotolitiche: Lino 2 – Città di Castello (Pg)

Stampa: Officine Grafiche Calderini – Ozzano dell'Emilia (Bo)

Redazione: Laura Scarcella

Progetto grafico e copertina: CD&V – Firenze

ISBN 88-528-0225-8

Per i casi in cui non sia stato possibile ottenere il permesso di riproduzione del materiale illustrativo, a causa della difficoltà a rintracciare il destinatario, l'editore è a disposizione degli eventuali aventi causa.

*Fotocopie per uso personale del lettore possono essere effettuate nei limiti del 15% di ciascun volume/fascicolo di periodico dietro pagamento alla SIAE del compenso previsto dall'art. 68, comma 4, della legge 22 aprile 1941 n. 633 ovvero dall'accordo stipulato tra SIAE, AIE, SNS, e CNA, CONFARTIGIANATO, CASA, CLAAI, CONFCOMMERCIO, CONFESERCENTI il 18 dicembre 2000.*

*Le riproduzioni per uso differente da quello personale potranno avvenire, per un numero di pagine non superiore al 15% del presente volume/fascicolo, solo a seguito di specifica autorizzazione rilasciata da AIDRO, via delle Erbe, n.2, 20121 Milano, telefax 02 809506, e-mail aidro@iol.it*

Siamo presenti sul World Wide Web all'indirizzo <http://www.calderini.it>

# Prefazione

La nuova edizione del Corso di Sistemi è stata profondamente rinnovata nella struttura, nell'impianto grafico e nei contenuti. Il nuovo progetto didattico è stato redatto con il fine di far acquisire allo studente le conoscenze teoriche e le abilità di progettazione indispensabili per una formazione professionale completa, adeguata a seguire i continui e profondi cambiamenti tecnologici e rispondente alle richieste provenienti dalla società e dall'industria.

La struttura modulare dell'opera, l'ampiezza delle tematiche trattate, la presentazione dell'architettura hardware e software dei microprocessori 8088/8086, di un PLC commerciale, dei **microcontrollori PIC e ST7**, l'utilizzazione diffusa dei programmi **UltiBOARD, MultiSIM e LabVIEW**, già proposti nel primo volume, nonché l'esposizione dei fondamenti di **MATLAB e SIMULINK**, consentono al docente di organizzare una programmazione flessibile e di adeguare le strategie formative alle sue esperienze professionali, ai livelli di apprendimento delle classi e alle risorse disponibili.

Gli argomenti sono presentati in modo semplice, sono corredati da esemplificazioni, tabelle, grafici, esempi di simulazioni e proposte di verifiche sperimentali, al fine di facilitarne la comprensione e di coinvolgere gli studenti con l'immediatezza e la praticità. Gli aspetti matematici della disciplina sono contenuti e quelli più complessi sono riportati in apposite schede, senza che la scientificità e la completezza della trattazione ne risultino penalizzate.

I contenuti del Modulo 10 *Area di progetto*, in particolare, assumono una valenza molto significativa perché offrono spunti per realizzare progetti in collaborazione con i docenti delle altre discipline di indirizzo, facilitando l'integrazione tra le conoscenze teoriche e l'attività sperimentale.

In tale ambito le proposte sono dedicate alla progettazione di circuiti di condizionamento di trasduttori e per il controllo di attuatori, di schede per microcontrollori PIC, ST7, per il sistema DAQPad 6020E e per il PLC.

Gli autori ringraziano le ditte:

**ARIZONA MICROCHIP TECHNOLOGY LTD.**, produttrice dei microcontrollori PIC.

**BORLAND INTERNATIONAL**, produttrice del Turbo Pascal, Turbo-C e C++ Builder.

**BRM ITALIANA**, distributrice di ORCAD e PSPICE.

**MATH WORKS S.r.l.**, distributrice di MATLAB e SIMULINK

**MICROSOFT CORPORATION**, produttrice del software MS-DOS, Windows 98/XP, Excel e Visual Basic.

**NATIONAL INSTRUMENTS ITALY S.r.l.**, produttrice di LabVIEW

**PATRUCCO**, distributrice di MultiSIM e UltiBOARD.

**SIEMENS S.p.A.**

**SOFTEC ITALIA S.r.l.**, distributrice dei prodotti SofTec Microsystems.

**TEORES.I.**

## Modulo 1 Sintesi di reti sequenziali

<b>Unità 1 Reti sequenziali con Moore e Mealy</b> .....	8
1.1 Sintesi con automa di Moore .....	9
1.2 Sintesi con automa di Mealy .....	22
<b>Unità 2 Reti combinatorie di I/O con dispositivi programmabili</b> .....	27
2.1 Reti combinatorie realizzate con dispositivi programmabili di tipo EPROM .....	27
<b>Unità 3 Dispositivi logici programmabili</b> .....	32
3.1 Dispositivi PLA, PAL e GAL .....	32
3.2 Dispositivi PLD, CPLD e FPGA .....	34
3.3 Programmazione dei dispositivi .....	34
Esercizi proposti .....	36

## Modulo 2 Trasduttori ed attuatori

<b>Unità 1 Trasduttori</b> .....	38
1.1 Linearizzazione caratteristica dei trasduttori .....	39
1.2 Trasduttori di temperatura .....	41
1.2.1 Termoresistenza RTD .....	41
1.2.2 Termistori .....	44
1.3 Trasduttore di luminosità .....	48
1.3.1 Il fotoresistore NORP-12 .....	48
1.4 Termocoppia .....	50
<b>Unità 2 Attuatori</b> .....	52
2.1 Solenoide .....	52
2.2 Amplificatori di potenza lineari .....	53
2.3 Tiristori .....	55
2.4 Relè .....	58
2.5 Motore passo-passo .....	59
Esercizi proposti .....	61

## Modulo 3 Dispositivi di memoria e programmabili

<b>Unità 1 Dispositivi di memoria</b> .....	64
1.1 Caratteristiche delle memorie .....	64
1.2 Classificazione sulle memorie .....	67
1.3 Memorie RAM per il PC .....	68
1.4 Tipi di moduli .....	69
1.5 Approfondimento sulle memorie .....	70
<b>Unità 2 Architettura di un microprocessore</b> .....	74
2.1 Architettura del microprocessore .....	75
2.2 Ricerca ed esecuzione dei codici operativi .....	77
2.3 L'evoluzione dei processori .....	79
2.3.1 Architettura RISC dei nuovi processori ..	81
2.3.2 Sistemi di memoria cache .....	83
2.4 Alcune caratteristiche dei processori ..	84
2.5 Quadro storico dei processori .....	85

<b>Unità 3 Interfacciamento delle periferiche con la CPU</b> .....	89
3.1 Collegamento delle periferiche ai bus della CPU .....	89
3.2 Interfaccia di periferiche di I/O con la CPU .....	91
3.2.1 Interfaccia di periferiche di input .....	91
3.2.2 Interfaccia di periferiche di output .....	93
3.2.3 Interfaccia di periferiche di input e output .....	94
3.3 Circuiti di decodifica .....	96
3.4 Tecniche di colloquio tra CPU e periferiche .....	97
<b>Unità 4 Architettura di un microcontroller</b> .....	100
4.1 Architettura di un microcontrollore ..	101
4.2 Periferiche di I/O di un microcontroller .....	102
<b>Unità 5 Strumenti di supporto per lo sviluppo di programmi in linguaggio assembly</b> .....	104
5.1 Linguaggio macchina .....	104
5.2 Linguaggio assembly .....	105
5.3 Fase di editing .....	106
5.4 Direttive dell'assemblatore .....	107
5.5 Fase di assemblaggio .....	108
5.6 Macro assemblatori .....	109
5.7 Fase di linking .....	109
5.8 Il debugger .....	110
Esercizi proposti .....	111

## Modulo 4 Microcontrollori PIC

<b>Unità 1 Caratteristiche di base dei microcontrollori PIC</b> .....	114
1.1 I microcontrollori della Microchip Technology .....	114
<b>Unità 2 Struttura interna dei microcontrollori PIC</b> .....	117
2.1 Il microcontroller PIC16F84A .....	118
2.1.1 Organizzazione della memoria .....	119
2.1.2 Nozioni di programmazione in assembler del PIC16F84A .....	120
2.1.3 Alcuni dei file register. Funzioni speciali .....	120
2.2 Introduzione alla simulazione di un programma .....	124
2.3 Analisi generale dei file register .....	125
<b>Unità 3 Il software dei microcontrollori</b> .....	131
3.1 La programmazione del microcontroller PIC16F84A .....	131
3.2 Esempi di programmazione .....	135
3.3 Programmi di I/O .....	136
<b>Unità 4 L'ambiente integrato di MPLAB</b> .....	143
4.1 Introduzione .....	143
4.2 Creazione di un progetto .....	144
4.3 Simulazione di un programma .....	146
4.4 Debug di un programma .....	149

4.4.1	Programmazione del microcontroller ..	151
<b>Unità 5</b>	<b>Programmazione in C dei microcontrollori</b>	154
5.1	Scrittura del file sorgente .....	154
5.2	Creazione di un progetto .....	155
5.3	Programmi di esempio .....	157
	Esercizi proposti .....	163

## Modulo 5 Microcontrollori ST7

<b>Unità 1</b>	<b>Microcontrollori ST7FLITE2</b>	166
1.1	CPU .....	167
1.2	Memoria .....	168
1.3	Pin out .....	169
<b>Unità 2</b>	<b>Programmazione in Assembler</b>	172
2.1	Istruzioni di caricamento .....	173
2.2	Istruzioni aritmetiche .....	174
2.3	Istruzioni logiche .....	175
2.4	Istruzioni di salto, chiamata e ritorno ...	176
2.5	Istruzioni varie e di controllo .....	177
2.6	Modi di indirizzamento dei micro ST7 .....	178
<b>Unità 3</b>	<b>PK-ST7FLITE2</b>	190
3.1	Progetto di un programma .....	191
3.2	ST7 Debugger .....	195
<b>Unità 4</b>	<b>Periferiche di I/O</b>	198
4.1	Porte di I/O .....	198
4.2	Watchdog .....	199
4.3	Autoreload Timer .....	200
4.4	Esercizi svolti .....	202
	Esercizi proposti .....	209

## Modulo 6 La famiglia dei processori 8088/86

<b>Unità 1</b>	<b>Processori x86</b>	212
1.1	Architettura interna dell'8086 .....	212
1.2	Principio di funzionamento della CPU 8086 .....	214
1.3	Descrizione dei segnali della CPU 8086 ..	216
1.4	Processori della famiglia x86 successivi all'8086 .....	217
<b>Unità 2</b>	<b>Programmazione dei processori x86</b>	219
2.1	Segmentazione degli indirizzi .....	219
2.2	Registri delle CPU 8086/88 .....	223
2.2.1	Registri per uso generale .....	224
2.2.2	Registri segmento .....	225
2.2.3	Registri indice, puntatore e base .....	227
2.2.4	Registro dei flag .....	228
2.3	Principi di programmazione .....	229
2.4	Esercizi svolti .....	231
2.4.1	Esercitazioni sulla segmentazione degli indirizzi .....	231
2.4.2	Esercizi sull'uso dei registri di transito .....	233
2.4.3	Esercizi con i registri indice e di segmento .....	234
2.5	Programmi proposti .....	235
2.5.1	Programmi più semplici .....	235
2.5.2	Programmi più complessi .....	236
<b>Unità 3</b>	<b>Programmazione in Assembler dell'8088/86</b>	238
3.1	Stesura di un programma in Assembler .....	238

3.2	Direttive dell'Assembler .....	239
3.3	Procedure .....	245
3.3.1	Procedure near e far .....	246
3.4	Inizializzazione dello stack .....	247
3.5	Tecniche di indirizzamento .....	249
3.6	Registro dei flag .....	251
3.7	Set di istruzioni dei processori 8088/86 .....	251
3.7.1	Istruzioni di trasferimento dei dati ....	252
3.7.2	Istruzioni aritmetiche e logiche .....	253
3.7.3	Istruzioni per il trattamento delle stringhe .....	258
3.7.4	Istruzioni di controllo del programma .....	261
3.7.5	Istruzioni sui flag .....	262
3.8	Interruzioni software .....	263

## Modulo 7 CAD

<b>Unità 1</b>	<b>Simulazione di MultiSIM</b>	272
1.1	Bode Plotter .....	272
1.2	Funzione di trasferimento .....	276
1.3	Fourier .....	277
1.4	Analizzatore logico .....	278
1.5	Analizzatore di spettro .....	279
1.6	Esercizi svolti .....	280
<b>Unità 2</b>	<b>Ambiente MultiSIM per UltiBOARD</b>	287
2.1	Nuovi componenti di MultiSIM .....	287
2.2	Modifica dei componenti in MultiSIM .....	288
2.3	Da MultiSIM ad UltiBOARD .....	288
2.4	Lo sbroglio di UltiBOARD .....	289
<b>Unità 3</b>	<b>MATLAB</b>	294
3.1	Tool Symbolic Math .....	294
3.2	Vettori matrici .....	295
3.3	Grafici .....	296
3.4	Control System Toolbox .....	296
3.5	Editor/Debugger .....	297
3.6	Esercizi svolti .....	298
<b>Unità 4</b>	<b>SIMULINK</b>	306
4.1	Libreria <i>Sources</i> .....	306
4.2	Libreria <i>Sinks</i> .....	308
4.3	Libreria <i>Linear</i> .....	311
4.4	Libreria <i>Connections</i> .....	312
4.5	Menu SIMULATION .....	313
4.6	Regole per la costruzione degli schemi a blocchi .....	313
<b>Unità 5</b>	<b>LabVIEW</b>	316
5.1	Registri .....	316
5.2	Foglio elettronico e I/O su file .....	318
5.3	Funzioni matematiche .....	319
5.4	Counter .....	322
5.4.1	Pulse Train .....	323
5.4.2	Pulse Period .....	324
5.4.3	Frequency .....	326
	Esercizi proposti .....	328

## Modulo 8 Sistemi lineari

<b>Unità 1</b>	<b>Sistemi lineari</b>	332
1.1	Sistemi del secondo ordine con $\zeta = 1$ .....	332
1.2	Sistemi del secondo ordine con $0 \leq \zeta < 1$ .....	334

1.3	Parametri della risposta al gradino ...	336
1.4	Risposta forzata e risposta libera di un sistema del primo ordine .....	339
1.5	Risposta forzata e risposta libera di un sistema del secondo ordine .....	341
1.6	Esercizi svolti .....	346
<b>Unità 2</b>	<b>Risposta in frequenza .....</b>	<b>360</b>
2.1	Diagrammi di Bode .....	360
2.1.1	Funzione $G(s) = k$ con $k > 1$ .....	362
2.1.2	Funzione $G(s) = 1 + s \cdot \tau$ .....	363
2.1.3	Funzione $G(s) = \frac{1}{1 + s \cdot \tau}$ .....	365
2.1.4	Funzione $G(s) = s$ .....	366
2.1.5	Funzione $G(s) = \frac{1}{s}$ .....	367
2.2	I diagrammi di Bode con MATLAB .....	368
2.3	Esercizi svolti .....	373
	Esercizi proposti .....	388

## Modulo 9 I PLC

<b>Unità 1</b>	<b>Caratteristiche di base dei PLC .....</b>	<b>392</b>
1.1	Principio di funzionamento di un PLC ..	393
1.2	Schemi a contatti .....	395
<b>Unità 2</b>	<b>Il sistema S7-200 .....</b>	<b>397</b>
2.1	Caratteristiche hardware del sistema ..	397
2.2	Principi di programmazione .....	399
2.3	Scrittura di un programma .....	401
2.4	Alcuni programmi .....	403
2.5	Analisi di alcuni comandi in KOP .....	405
	Esercizi proposti .....	418

## Modulo 10 Area di progetto

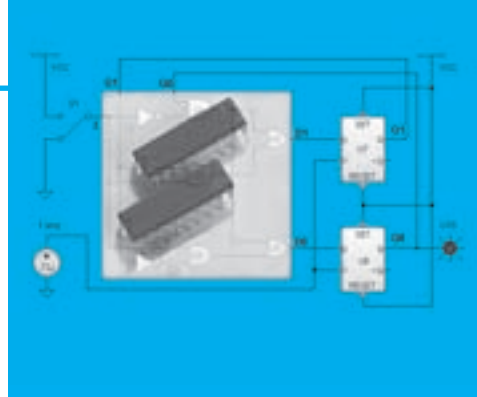
<b>Unità 1</b>	<b>Controllo di motore passo-passo in LabVIEW .....</b>	<b>420</b>
<b>Unità 2</b>	<b>Misura di grandezze fisiche in LabVIEW .....</b>	<b>427</b>
2.1	Measurement & Automation .....	430
2.2	Misura di temperatura .....	432

<b>Unità 3</b>	<b>Sistema di controllo ON/OFF .....</b>	<b>434</b>
3.1	Controllo di temperatura con LabVIEW .....	435
3.2	Controllo di temperatura con ST7 .....	436
<b>Unità 4</b>	<b>Applicazioni dei microcontrollori PIC .....</b>	<b>440</b>
4.1	Presentazione della scheda .....	440
4.2	Applicazioni software .....	443
<b>Unità 5</b>	<b>Controllo del livello di un liquido con PLC .....</b>	<b>446</b>
5.1	Descrizione del sistema .....	446
5.2	Circuiti di condizionamento .....	446
5.2.1	Dimensionamento del condizionatore per i sensori del liquido .....	447
5.2.2	Dimensionamento del condizionatore per il trasduttore di temperatura .....	448
5.3	Collegamenti al PLC dei trasduttori e delle elettrovalvole .....	449
5.4	Programma di gestione del sistema ..	449
<b>Unità 6</b>	<b>Misure della temperatura .....</b>	<b>450</b>
6.1	Trasduttore PT100 .....	450
6.2	Termocoppia .....	451
<b>Unità 7</b>	<b>Automi con dispositivi programmabili .....</b>	<b>453</b>
7.1	Realizzazione di un automa .....	453
7.2	Rete combinatoria emulata con un PIC16F84A .....	454
7.3	Realizzazione di un automa con dispositivo programmabile a stati .....	458

<b>Appendice: il linguaggio C .....</b>	<b>464</b>
A.1.1 L'ambiente a linea di comando del C++ Builder .....	464
A.1.2 Tipi di dati .....	465
A.1.3 Variabili e costanti .....	466
A.1.4 Schematizzazione dei programmi in C .....	468
A.1.5 Operatori .....	469
A.1.6 Le funzioni printf( ) e scanf( ) .....	473
A.1.7 Altre funzioni di input e output .....	475
A.1.8 Le funzioni del linguaggio C .....	476
A.1.9 Le strutture di controllo in C .....	480
A.1.10 Array .....	487
A.1.11 I puntatori .....	489
A.1.12 Funzioni aritmetiche .....	491

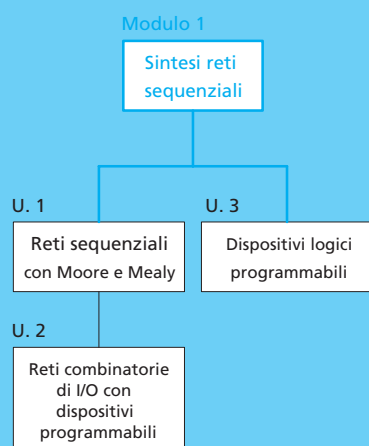
<b>Indice analitico .....</b>	<b>493</b>
-------------------------------	------------

# Modulo 1. Sintesi di reti sequenziali



- ▶ UNITÀ 1. Reti sequenziali con Moore e Mealy
- ▶ UNITÀ 2. Reti combinatorie di I/O con dispositivi programmabili
- ▶ UNITÀ 3. Dispositivi logici programmabili

UCATION SPA



## PREREQUISITI

- ▶ Conoscenza dei sistemi di numerazione binario ed esadecimale
- ▶ Possesso delle conoscenze di base di elettronica e dei dispositivi logici con particolare riferimento ai flip-flop
- ▶ Conoscenza delle modalità di sintesi delle reti logiche combinatorie
- ▶ Possesso delle conoscenze di base degli argomenti esposti nel Modulo 5 del Volume 1

## OBIETTIVI

- ▶ Effettuare la sintesi di una rete sequenziale a partire da un grafo di flusso utilizzando automi di Moore e di Mealy
- ▶ Saper eseguire la simulazione di una rete logica utilizzando un programma specifico
- ▶ Acquisire gli elementi di base sui dispositivi logici programmabili

# 1. Reti sequenziali con Moore e Mealy

## unità

Per descrivere i passi da seguire per effettuare la sintesi di una rete sequenziale con i modelli di Moore o di Mealy, debbono tenersi ben presenti i modelli delle reti raffigurati nella [figura 1.1](#).

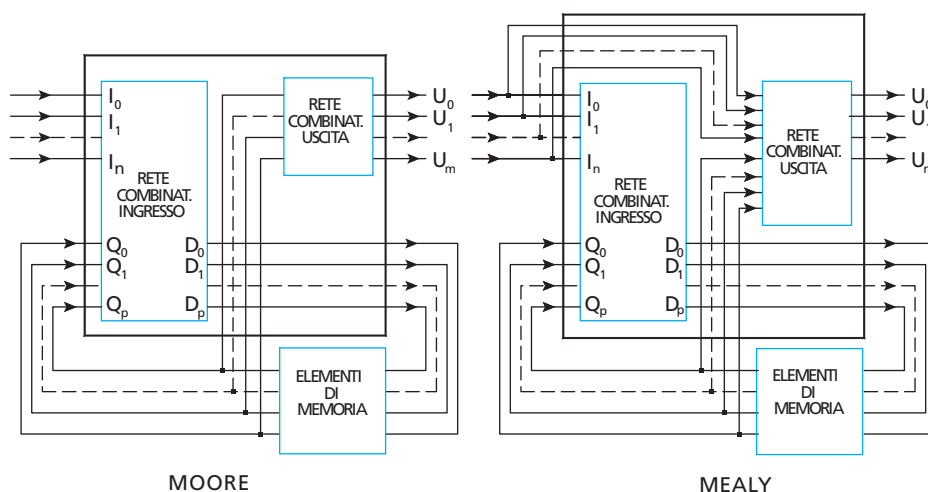


Fig. 1.1

Una volta tracciato il *diagramma degli stati* (→ Volume 1, Modulo 5, Unità 2), il procedimento consiste essenzialmente nel realizzare:

- la sintesi della *rete combinatoria d'ingresso* scegliendo il tipo di elementi di memoria da utilizzare (flip-flop). Le uscite della rete combinatoria d'ingresso diventano gli ingressi dei flip-flop;
- la sintesi della *rete combinatoria d'uscita*. Gli ingressi della rete combinatoria d'uscita sono costituiti dalle uscite dei flip-flop (Moore) o dalle uscite dei flip-flop e dagli ingressi esterni (Mealy).

Per effettuare la sintesi di reti sequenziali sincrone si deve quindi procedere seguendo i passi elencati.

1. Dalla descrizione del problema si traccia il *diagramma degli stati* del sistema. Normalmente è il passo più difficile da realizzare in quanto è richiesta un'attenta analisi del problema.
2. Si associano ai singoli stati, arbitrariamente, le *variabili di stato*. Il numero delle variabili di stato è legato al numero degli stati presenti nel sistema. Essendo  $N$  il numero degli stati del sistema, per la sintesi della rete occorrono  $n$  variabili di stato in modo che sia  $N \leq 2^n$ .
3. Si associano ai singoli stati le configurazioni possibili delle uscite tenendo conto dei valori indicati per esse nel diagramma degli stati (*tabella d'uscita*).
4. Si pone il diagramma degli stati sotto forma di tabella (*tabella degli stati* o *tabella delle transizioni*) tenendo conto dello stato da cui si parte e quello in cui si porta il sistema in base al livello assunto dall'ingresso o dagli ingressi.
5. Si esegue, ove possibile, un'eventuale minimizzazione della tabella degli stati con la fusione di stati equivalenti.
6. Si stabilisce il tipo di elemento di memoria (flip-flop) da impiegare per la sintesi.
7. Si costruisce la *tabella di eccitazione* per la sintesi della rete combinatoria d'ingresso tenendo conto delle *tabelle di transizione* del flip-flop scelto.



8. Dalla tabella di eccitazione per mezzo delle *mappe di Karnaugh* si formulano le *equazioni di eccitazione* per mezzo delle quali si procede alla sintesi della rete combinatoria d'ingresso.
9. Dalla tabella di uscita per mezzo delle mappe di Karnaugh si formulano le *equazioni d'uscita* per mezzo delle quali si procede alla sintesi della rete combinatoria d'uscita.
10. Si procede alla sintesi della rete *combinatoria d'ingresso* e di quella *d'uscita*.
11. Si fa una eventuale simulazione logica della rete complessiva sintetizzata.

Alcuni esempi chiariranno il modo di procedere.

## 1.1 ● Sintesi con automa di Moore

### → Esempio 1

Si voglia realizzare la rete logica descritta con il diagramma degli stati analizzato nella [figura 1.2](#). Il diagramma descrive un sistema (con un solo ingresso  $I$  e una sola uscita  $U$ ) che deve cambiare stato ogni volta che l'interruttore fornisce un livello che passa dallo stato basso a quello alto.

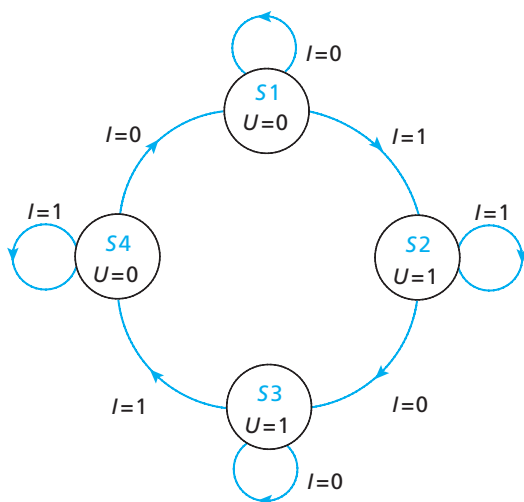


Fig. 1.2

Essendo presenti nel sistema 4 stati denominati  $S1$ ,  $S2$ ,  $S3$  e  $S4$ , occorrono 2 variabili di stato ( $Q_1$  e  $Q_0$ ) che possono essere associate arbitrariamente agli stati come nella [tabella 1.1](#). In tale tabella sono riportati anche i livelli che assume l'uscita nei vari stati.

Come prima cosa conviene tradurre il diagramma degli stati nella corrispondente tabella di transizione degli stati ([tab. 1.2](#)). Nella tabella 1.2, nella prima colonna, si è indicato lo stato da cui si parte e nelle altre colonne quello in cui si porta il sistema in base al livello assunto dall'ingresso. In questa tabella sono riportati, accanto agli stati, anche i valori assunti dalle variabili di stato (tra parentesi quadre). La stessa tabella può essere riscritta in base ai valori assunti dalla variabile di stato in corrispondenza delle diverse combinazioni dell'ingresso.

TAB. 1.1

Stato	$Q_1$	$Q_0$	$U$
S1	0	0	0
S2	0	1	1
S3	1	1	1
S4	1	0	0

TAB. 1.2

Stato di partenza	$I$	
	0	1
S1 [0 0] $\Rightarrow$	S1 [0 0]	S2 [0 1]
S2 [0 1] $\Rightarrow$	S3 [1 1]	S2 [0 1]
S3 [1 1] $\Rightarrow$	S3 [1 1]	S4 [1 0]
S4 [1 0] $\Rightarrow$	S1 [0 0]	S4 [1 0]

Dalla tabella 1.2, quindi, prelevando i valori posti tra parentesi, si traccia immediatamente la [tabella 1.3](#), dove le variabili sulle prime colonne sono relative a  $Q_1(t)$  e quelle delle seconde colonne a  $Q_0(t)$ . La tabella 1.3 indicherà quindi il valore assunto dalla variabile di stato nello stato di partenza  $Q(t)$  e quello a cui si deve portare  $Q(t+1)$  in base al valore assunto dall'ingresso, quando il sistema passerà nel nuovo stato o permarrà in quello in cui si trova.

TAB. 1.3

$Q_1(t)$	$Q_0(t)$		$I = 0$		$I = 1$	
			$Q_1(t+1)$	$Q_0(t+1)$	$Q_1(t+1)$	$Q_0(t+1)$
0	0	$\Rightarrow$	0	0	0	1
0	1	$\Rightarrow$	1	1	0	1
1	1	$\Rightarrow$	1	1	1	0
1	0	$\Rightarrow$	0	0	1	0

Per la sintesi occorre utilizzare due flip-flop essendo quattro gli stati e quindi due le variabili di stato. Se si utilizzano **flip-flop di tipo D**, coincidendo la tabella di transizione con quella di eccitazione, è possibile tracciare immediatamente le *mappe di Karnaugh* relative a  $D_1$  e  $D_0$ . Nella [tabella 1.3](#), si prendono separatamente, nella parte relativa a  $I = 0$  e  $I = 1$ , le variabili delle prime colonne (per  $D_1$ ) e quelle delle seconde (per  $D_0$ ). Si ottengono così le mappe riportate di seguito.

$Q_1 Q_0 \backslash I$	0	1
0 0	0	0
0 1	1	0
1 1	1	1
1 0	0	1

$$D_1 = Q_0 \bar{I} + Q_1 I$$

$Q_1 Q_0 \backslash I$	0	1
0 0	0	1
0 1	1	1
1 1	1	0
1 0	0	0

$$D_0 = Q_0 \bar{I} + \bar{Q}_1 I$$

TAB. 1.4

Tipo T			
$Q(t)$	$\Rightarrow$	$Q(t+1)$	T
0	$\Rightarrow$	0	0
0	$\Rightarrow$	1	1
1	$\Rightarrow$	0	1
1	$\Rightarrow$	1	0

Se si volessero usare per la sintesi dei **flip-flop di tipo T** dovrebbe essere presa in considerazione la tabella di transizione di questo tipo di dispositivo ([tab. 1.4](#)).

In questo caso la [tabella 1.3](#) si trasforma nella [tabella 1.5](#). Si ricordi che ogni volta che dallo stato di partenza a quello di arrivo si passa da 0 a 1 o da 1 a 0 si ha commutazione e si deve porre nella casella corrispondente un uno, mentre se non c'è commutazione ( $0 \rightarrow 0$  o  $1 \rightarrow 1$ ), uno zero.

TAB. 1.5

			$I = 0$		$I = 1$	
$Q_1(t)$	$Q_0(t)$		$Q_1(t+1)$	$Q_0(t+1)$	$Q_1(t+1)$	$Q_0(t+1)$
0	0	$\Rightarrow$	0	0	0	1
0	1	$\Rightarrow$	1	0	0	0
1	1	$\Rightarrow$	0	0	0	1
1	0	$\Rightarrow$	1	0	0	0

Procedendo alla sintesi con il flip flop di tipo T si hanno le mappe seguenti.

$Q_1 Q_0 \backslash I$	0	1
0 0	0	0
0 1	1	0
1 1	0	0
1 0	1	0

$Q_1 Q_0 \backslash I$	0	1
0 0	0	1
0 1	0	0
1 1	0	1
1 0	0	0

Come si può vedere dalle mappe, non sono possibili raggruppamenti e pertanto si ha per  $T_1$  e per  $T_0$ :

$$T_1 = \bar{Q}_1 \cdot Q_0 \cdot \bar{I} + Q_1 \cdot \bar{Q}_0 \cdot \bar{I}$$

$$T_0 = \bar{Q}_1 \cdot \bar{Q}_0 \cdot I + Q_1 \cdot Q_0 \cdot I$$

Usando **flip-flop di tipo J-K** – osservando la tabella di transizione per questo flip-flop ([tab. 1.6](#)) e la [tabella 1.3](#) di transizione degli stati (riportata di nuovo nella pagina accanto per comodità e scissa in due parti per eseguire più semplicemente la costruzione della [tab. 1.7](#)) –, si ricava la *tabella di eccitazione* ([tab. 1.7](#)) anch'essa divisa in due parti per semplicità.

TAB. 1.6

Tabella transizione J-K				
$Q(t)$		$Q(t+1)$	$J$	$K$
0	$\Rightarrow$	0	0	X
0	$\Rightarrow$	1	1	X
1	$\Rightarrow$	0	X	1
1	$\Rightarrow$	1	X	0

TAB. 1.3

		$I = 0$	$I = 1$
$Q_1(t)$		$Q_1(t+1)$	$Q_1(t+1)$
0	$\Rightarrow$	0	0
0	$\Rightarrow$	1	0
1	$\Rightarrow$	1	1
1	$\Rightarrow$	0	1

		$I = 0$	$I = 1$
$Q_0(t)$		$Q_0(t+1)$	$Q_0(t+1)$
0	$\Rightarrow$	0	1
1	$\Rightarrow$	1	1
1	$\Rightarrow$	1	0
0	$\Rightarrow$	0	0

TAB. 1.7

Stato di partenza		$I = 0$		$I = 1$	
$Q_1(t)$		$Q_1(t+1)$		$Q_1(t+1)$	
		$J_1$	$K_1$	$J_1$	$K_1$
0	$\Rightarrow$	0	X	0	X
0	$\Rightarrow$	1	X	0	X
1	$\Rightarrow$	X	0	X	0
1	$\Rightarrow$	X	1	X	0

Stato di partenza		$I = 0$		$I = 1$	
$Q_0(t)$		$Q_0(t+1)$		$Q_0(t+1)$	
		$J_0$	$K_0$	$J_0$	$K_0$
0	$\Rightarrow$	0	X	1	X
1	$\Rightarrow$	X	0	X	0
1	$\Rightarrow$	X	0	X	1
0	$\Rightarrow$	0	X	0	X

Dalla tabella 1.7 si tracciano le mappe per  $J_1$ ,  $K_1$  e  $J_0$ ,  $K_0$ .

$I$		0	1
$Q_1 Q_0$	0 0	0	0
	0 1	1	0
	1 1	X	X
	1 0	X	X

$$J_1 = \bar{I} \cdot Q_0$$

$I$		0	1
$Q_1 Q_0$	0 0	X	X
	0 1	X	X
	1 1	0	0
	1 0	1	0

$$K_1 = \bar{I} \cdot \bar{Q}_0$$

$I$		0	1
$Q_1 Q_0$	0 0	0	1
	0 1	X	X
	1 1	X	X
	1 0	0	0

$$J_0 = I \cdot \bar{Q}_1$$

$I$		0	1
$Q_1 Q_0$	0 0	X	X
	0 1	0	0
	1 1	0	1
	1 0	X	X

$$K_0 = I \cdot Q_1$$

Per realizzare la rete occorrono due flip-flop J-K, una NOT (per invertire l'ingresso  $I$ ) e quattro AND.

## Realizzazione della rete progettata con 2 flip-flop di tipo D

È necessario effettuare, servendosi della tabella 1.5, anche la sintesi della rete combinatoria di uscita. Si può osservare dalla tabella che l'uscita  $U$ , per tutti e quattro gli stati, coincide con la variabile di stato  $Q_0$ , per questo si può scrivere:

$$U = Q_0$$

Si realizza quindi la rete progettata usando 2 flip-flop di tipo D. Nello schema di [figura 1.3](#) (realizzato con MultiSIM), utilizzato per eseguire la simulazione del sistema, gli ingressi dei flip-flop di *clear* e *preset* sono disattivati tenendoli allo stato alto. Una sonda che segnala lo stato dell'uscita  $U$  è collegata a  $Q_0$  del flip-flop. Il clock è stato fissato con una frequenza di 1 kHz. Per eseguire la simulazione del circuito si attivi di volta in volta l'interruttore S1 con la barra spaziatrice, portandolo a  $V_{CC}$  e poi di nuovo a GND; la sonda posta in uscita si accende e si spegne ogni volta che si ha una transizione da livello basso ad alto di S1 in accordo con le specifiche del progetto.

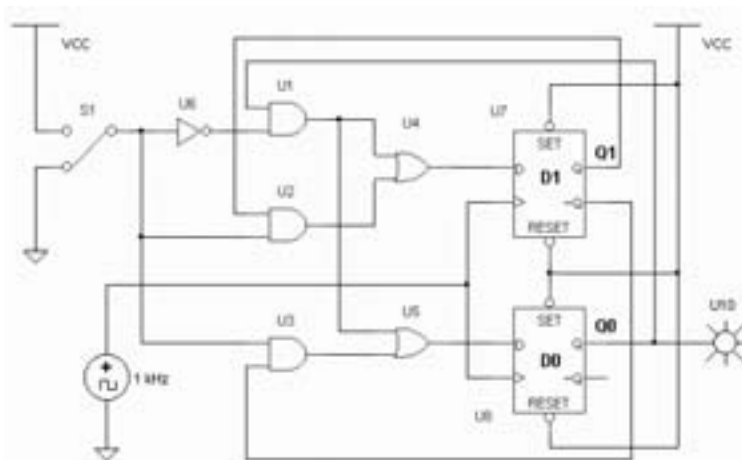


Fig. 1.3

## → Esempio 2

Si realizzi un sistema digitale che effettui il controllo di un dispositivo elettro-meccanico costituito da un asse (vite senza fine), posto in rotazione da un motore; su esso si muove una chiocciola che può variare il senso di marcia facendo ruotare il motore nel verso opposto a quello precedente. In prossimità delle due estremità dell'asse sono posti due switch di fine corsa azionati dalla chiocciola. I fine corsa forniscono in condizione di riposo un livello basso e, quando attivati, un livello alto. Il sistema di controllo deve essere in grado di variare il verso di marcia della chiocciola, invertendo il senso di rotazione del motore ogni volta che essa giunge su una delle estremità dell'asse ed aziona il corrispondente fine corsa (SW1 o SW2).

Per il diagramma degli stati e la trattazione completa del sistema → Volume 1, Modulo 5, Unità 2, esempio 4. Per invertire il senso di marcia del motore, che si suppone sia in corrente continua, alimentato a 12 V, si deve invertire la polarità di alimentazione dell'armatura utilizzando per esempio un relè a doppio scambio. Per motori con tensioni e correnti più elevate può essere utilizzato un teleruttore. In [figura 1.4](#) è riportato lo schema per invertire la polarità di alimentazione del motore.

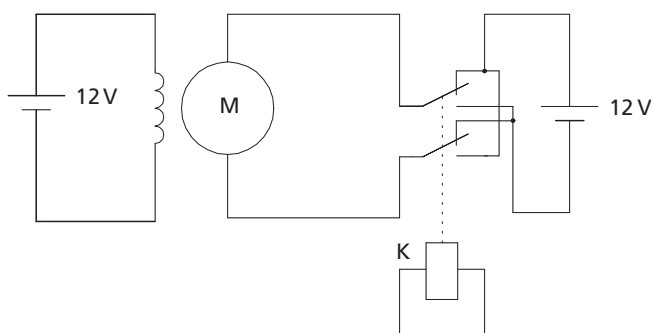


Fig. 1.4

La rete logica da realizzare deve quindi avere un'uscita ( $U$ ) che alimenta la bobina  $K$  del relè e due ingressi ( $I_1$  e  $I_2$ ) collegati agli switch.

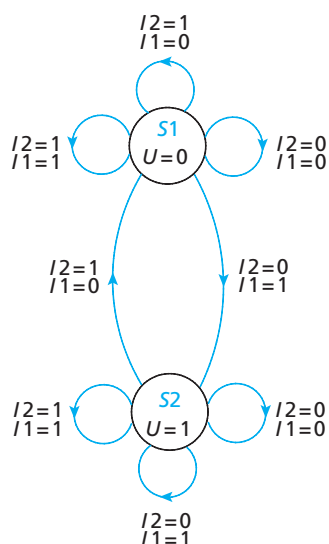


Fig. 1.5

In figura 1.5 è riportata la rappresentazione di Moore del diagramma degli stati del sistema.

Il sistema con due stati, il primo con uscita  $U = 0$  ed il secondo con  $U = 1$  si porta dallo stato S1 allo stato S2 solamente quando  $I_1 = 1$  e  $I_2 = 0$  (è attivo SW1) e dallo stato S2 ad S1 con  $I_1 = 0$  e  $I_2 = 1$  (attivo SW2). Per tutte le altre combinazioni degli ingressi il sistema rimane negli stati in cui si trova. È stata presa in considerazione anche la combinazione degli ingressi  $I_1 = 1$  e  $I_2 = 1$  (che non si può avere) ponendo anche per tale condizione la permanenza del sistema nello stato attuale.

TAB. 1.8

Stato	Q	U
S1	1	1
S2	0	0

Essendo presenti nel sistema solamente due stati si avrà una sola *variabile di stato*, che sarà chiamata Q. Si associ in modo arbitrario il valore assunto dalla variabile di stato con i singoli stati, come illustrato nella tabella 1.8.

Nella tabella è stato inserito anche il valore che assume l'uscita in corrispondenza dei singoli stati. Come è possibile osservare, non è necessario effettuare la sintesi della rete combinatoria d'uscita in quanto si ha  $U = Q$ . Per la realizzazione del sistema occorrerà utilizzare un solo flip-flop. La variabile di stato rappresenta l'uscita del flip-flop.

Il diagramma degli stati deve essere tradotto in una tabella (*tabella di transizione degli stati*) che descrive le transizioni presenti nel diagramma in base alle configurazioni che hanno gli ingressi.

Nella tabella 1.9, nella prima colonna si è indicato lo stato da cui si parte e nelle altre colonne quello in cui si porta il sistema in base alla combinazione che hanno gli ingressi. Accanto allo stato si è posto tra parentesi anche il valore assegnato alla variabile di stato Q che può desumersi dalla tabella 1.8.

La tabella 1.9 può essere riscritta in base ai valori assunti dalla variabile di stato in corrispondenza delle diverse combinazioni degli ingressi. La tabella 1.10 indicherà quindi il valore assunto dalla variabile di stato nello stato di partenza  $Q(t)$  e quello a cui si deve portare  $Q(t+1)$  in base al valore assunto dagli ingressi, quando il sistema passerà nel nuovo stato o permarrà in quello in cui si trova.

TAB. 1.9

Stato di partenza	$I_2 I_1$			
	0 0	0 1	1 1	1 0
S1 [0] $\Rightarrow$	S1 [0]	S2 [1]	S1 [0]	S1 [0]
S2 [1] $\Rightarrow$	S2 [1]	S2 [1]	S2 [1]	S1 [0]

TAB. 1.10

$Q(t)$	$I_2 I_1$			
	0 0	0 1	1 1	1 0
0 $\Rightarrow$	0	1	0	0
1 $\Rightarrow$	1	1	1	0

Si deve ora effettuare la scelta del tipo di flip-flop con cui realizzare la rete. Questa scelta può essere fatta tra flip-flop di tipo D, T o J-K. Prescelto in modo arbitrario il tipo di flip-flop, si dovrà prendere in considerazione per esso la relativa tabella di transizione ( $\rightarrow$  Volume 1, Modulo 5, Unità 1).

## Realizzazione della rete con un flip-flop di tipo D

TAB. 1.11

Tipo D			
$Q(t)$		$Q(t+1)$	D
0	$\Rightarrow$	0	0
0	$\Rightarrow$	1	1
1	$\Rightarrow$	0	0
1	$\Rightarrow$	1	1

Per effettuare la sintesi si scelga un flip-flop di tipo D.

Per questo flip-flop la tabella di transizione è riportata nella tabella 1.11.

Si ricordi che per i flip-flop di tipo D, le tabelle di transizione e di eccitazione sono uguali in quanto il valore dell'uscita  $Q(t+1)$  coincide con il valore che deve avere l'ingresso D all'istante t.

Quindi, utilizzando per la sintesi il flip-flop di tipo *D*, può essere utilizzata direttamente la tabella 1.8 per costruire la rete combinatoria d'ingresso. La tabella può essere posta sotto forma di mappa di Karnaugh.

$I_2 I_1$	0 0	0 1	1 1	1 0
$Q$				
0	0	1	0	0
1	1	1	1	0

$$D = Q \cdot \bar{I}_2 + \bar{I}_2 \cdot I_1 + Q \cdot I_1$$

Tenendo conto che in questo caso i livelli assunti dall'uscita della rete coincidono con quelli assegnati alla variabile di stato  $Q$  ( $U \equiv Q$ ), in base alla tabella 1.8, non si rende necessaria la realizzazione della *rete combinatoria di uscita*. Si può quindi procedere alla sintesi dell'intera rete in base allo schema generale di [figura 1.6](#). Il circuito è realizzato con MultiSIM. All'ingresso della rete sono collegati i segnali d'ingresso esterni  $I_1$  e  $I_2$  provenienti dai fine corsa (*SW1* e *SW2*) e la variabile di stato  $Q$  che è l'uscita dell'elemento di memoria. Un BJT collegato all'uscita del flip-flop pilota le bobine *K1* e *K2* di due relè posti in parallelo (sostituiscono un singolo relè a doppio scambio). Nel circuito è presente anche uno switch collegato con il clear del D-LATCH per avviare il sistema con  $U = 0$ . La frequenza del clock è fissata a 100 Hz. Sull'asse del motore, per controllare il verso di rotazione, sono stati inseriti due led. Quello acceso indica in quale direzione si sta muovendo la chiocciola e quindi quale dei due switch *SW1* o *SW2* deve essere azionato.

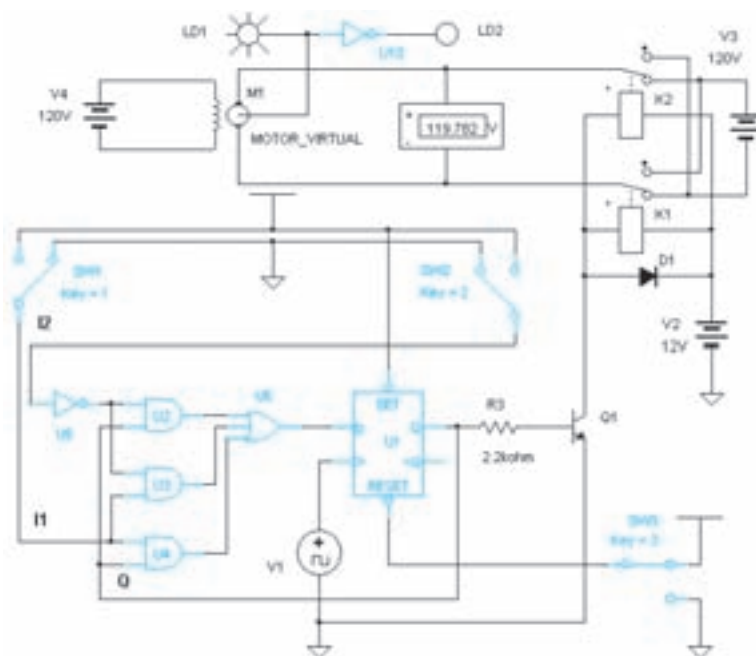


Fig. 1.6

### Esecuzione della simulazione

1. Gli switch *SW1* e *SW2*, attivabili con i tasti numerici 1 e 2, debbono trovarsi come in [figura 1.6](#) (livello basso).
2. Avviata la simulazione, portare *SW3* a massa e poi a livello alto con il tasto numerico 3.
3. Se è acceso *LD1* con il tasto numerico 1 portare per qualche istante *SW1* a livello alto e poi riportarlo a massa (viene simulata la pressione del carrello sul fine corsa). Dopo alcuni istanti, i relè *K1* e *K2* commutano e si accende *LD2* (il motore ha invertito il senso di rotazione e il carrello si muove nella direzione opposta).
4. Con il tasto numerico 2 portare per qualche istante *SW2* a livello alto e poi riportarlo a massa. Dopo alcuni istanti, i relè *K1* e *K2* commutano di nuovo e si accende *LD1*.

### Note

- I relè adoperati nel circuito (menu BASIC  $\Rightarrow$  RELAY  $\Rightarrow$  RELAY1C) sono polarizzati e quindi devono essere inseriti come in [figura](#). Per i relè sono stati adoperati i simboli DIN da attivare con OPTIONS  $\Rightarrow$  PREFERENCES  $\Rightarrow$  COMPONENT BIN, per gli altri componenti i simboli ANSI.

- Il diodo posto in parallelo alle bobine dei relè è di protezione per il BJT e nel circuito di simulazione può anche non essere inserito.
- Il voltmetro in parallelo al motore segnala l'inversione di polarità quando avviene la commutazione dei relè.
- Nel circuito è stato usato un motore DC alimentato a 120 V che è quello disponibile in MultiSIM (menu Misc  $\Rightarrow$  MOTOR).

### Esecuzione della simulazione in modalità automatica

Inserendo al posto degli switch SW1 e SW2 un generatore di parole, la simulazione può essere eseguita automaticamente e in modo ciclico. Lo schema elettrico del sistema è riprodotto in [figura 1.7](#). Nel circuito è stata introdotta una rete RC per il reset automatico del flip-flop all'avvio del sistema: inizialmente il condensatore è scarico e quindi sul pin di reset c'è livello basso; poi il condensatore si carica e il pin si porta, a regime, a livello alto.

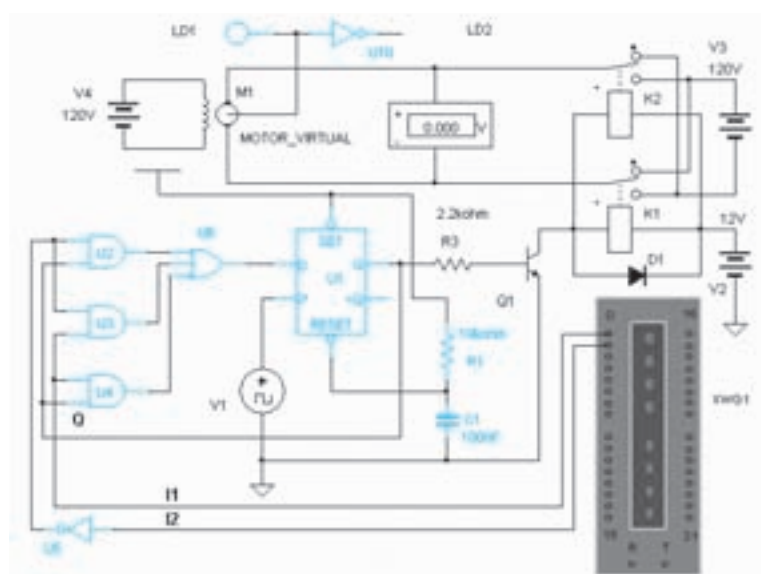


Fig. 1.7



Fig. 1.8

La programmazione del generatore è riportata in [figura 1.8](#).

- Impostare il clock a 100 Hz. La scelta della frequenza potrebbe essere modificata in relazione a quella del computer utilizzato.
- Programmare le parole da 0000h fino a 000Eh come in figura.
- Selezionare Cycle.
- Avviata la simulazione, il generatore di parole commuta automaticamente gli ingressi della rete combinatoria, invertendo ogni volta il verso di rotazione del motore.



### → Esempio 3

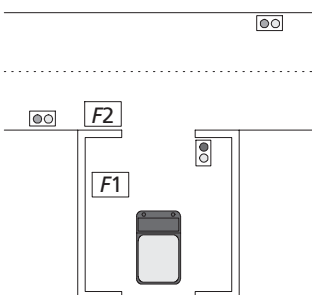


Fig. 1.9

In un garage dei vigili del fuoco che si affaccia su una strada con traffico di autoveicoli, è presente un automezzo. Quando questo deve uscire, dirigendosi verso l'uscita, attiva la fotocellula F1. Due semafori con luce verde e rossa sono posti sulla strada e un altro all'uscita del garage. Con l'attivazione della fotocellula F1 i semafori sulla strada passano dal verde al rosso e quello sull'uscita del garage dal rosso al verde. Una seconda fotocellula, posta sull'uscita del garage, intercetta l'autoveicolo e segnala quando esso è completamente uscito in modo da ripristinare lo stato iniziale dei semafori ([fig. 1.9](#)).

Per il diagramma degli stati e la trattazione completa del sistema → Volume 1, Modulo 5, Unità 2, esempio 5.

Si può rappresentare il sistema con due ingressi ( $F_1$  e  $F_2$ ) e un'uscita  $U$  che controlla tutti i semafori.

Nella [tabella 1.12](#) sono descritti gli stati del sistema in base alle condizioni delle due fotocelle.

TAB. 1.12

$F_2$	$F_1$	Stato del sistema
0	0	veicolo fermo o completamente uscito
0	1	veicolo in movimento
1	1	veicolo in uscita
1	0	veicolo in uscita

TAB. 1.13

$U$	Semafori stradali	Semaforo interno
0	verde	rosso
1	rosso	verde

Nella [figura 1.10](#) è riportato il diagramma degli stati del sistema. Il sistema logico combinatorio dell'uscita che pilota i semafori è realizzato seguendo le indicazioni della [tabella 1.13](#).

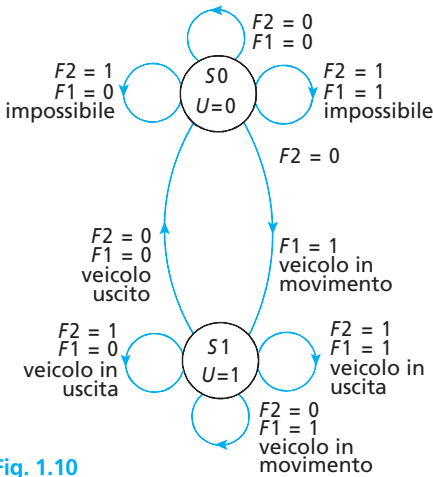


Fig. 1.10

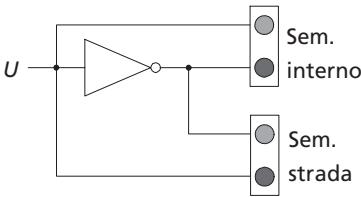


Fig. 1.11

Essendo presenti nel sistema solamente due stati si avrà la **variabile di stato  $Q$**  che verrà associata in modo arbitrario con i singoli stati come illustrato nella [tabella 1.14](#).

TAB. 1.14

Stato	$Q$	$U$
$S_0$	1	1
$S_1$	0	0

Partendo dal diagramma degli stati si ricava la [tabella di transizione degli stati](#) ([tab. 1.15](#)). Accanto agli stati, tra parentesi quadre, è stato posto anche il livello che assume la variabile di stato. La seconda parte della tabella, che può anche non essere scritta, riporta i soli valori della variabile di stato.

TAB. 1.15

Stato di partenza	$F_2 F_1$			
	0 0	0 1	1 1	1 0
$S_0 [0] \Rightarrow$	$S_0 [0]$	$S_1 [1]$	$S_0 [0]$	$S_0 [0]$
$S_1 [1] \Rightarrow$	$S_0 [0]$	$S_1 [1]$	$S_1 [1]$	$S_1 [1]$

$Q(t)$	$F_2 F_1$			
	0 0	0 1	1 1	1 0
0 $\Rightarrow$	0	1	0	0
1 $\Rightarrow$	0	1	1	1



## Realizzazione della rete con un flip-flop di tipo J-K

Tenendo presente la tabella di transizione del J-K e la seconda parte della tabella 1.15, si realizza la [tabella 1.17](#) e quindi si tracciano le mappe per J e K.

TAB. 1.16

Tabella transizione J-K				
Q(t)	⇒	Q(t + 1)	J	K
0	⇒	0	0	X
0	⇒	1	1	X
1	⇒	0	X	1
1	⇒	1	X	0

TAB. 1.17

Stato di partenza		F <sub>2</sub> F <sub>1</sub>							
		0 0		0 1		1 1		1 0	
Q(t)		Q(t + 1)		Q(t + 1)		Q(t + 1)		Q(t + 1)	
		J	K	J	K	J	K	J	K
0	⇒	0	X	1	X	0	X	0	X
1	⇒	X	1	X	0	X	0	X	0

F <sub>2</sub> F <sub>1</sub>	0 0	0 1	1 1	1 0
Q				
0	0	1	0	0
1	X	X	X	X

$$J = \bar{F}_2 \cdot F_1$$

F <sub>2</sub> F <sub>1</sub>	0 0	0 1	1 1	1 0
Q				
0	X	X	X	X
1	1	0	0	0

$$K = \bar{F}_2 \cdot \bar{F}_1$$

La rete dell'intero sistema è riportata nella [figura 1.12](#). Nello schema sono introdotti anche due segnalatori per controllare lo stato delle fotocellule.

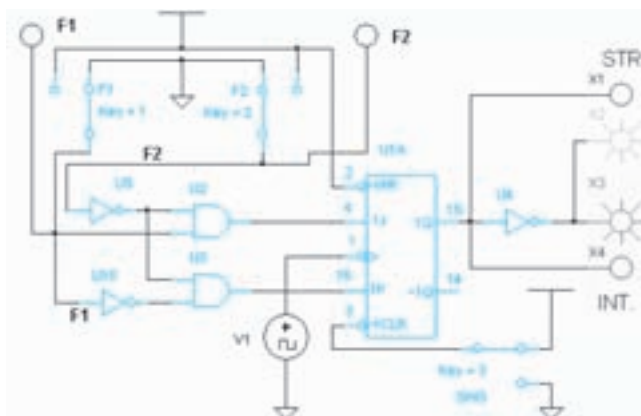


Fig. 1.12

## Esecuzione della simulazione

Una volta avviata la simulazione si devono eseguire i passi elencati di seguito.

1. Porre F1 ed F2 a livello basso (stato iniziale S0, veicolo fermo); portare con il tasto numerico 3 SW3 a livello basso e poi alto (reset del flip-flop). Il semaforo stradale è verde.
2. Portare F1 a livello alto con il tasto numerico 1 (il veicolo in movimento intercetta la fotocellula F1). Si passa allo stato S1. Il semaforo stradale diventa rosso.

- Portare  $F2$  a livello alto con il tasto numerico 2 (il veicolo in movimento intercetta la fotocellula  $F2$ ).
- Portare  $F1$  a livello basso (il veicolo in movimento libera la fotocellula  $F1$ ).
- Portare  $F2$  a livello basso (il veicolo in movimento libera la fotocellula  $F2$ ). Il veicolo è completamente uscito. Si ritorna allo stato iniziale  $S0$  e il semaforo stradale ritorna verde.



#### → Esempio 4

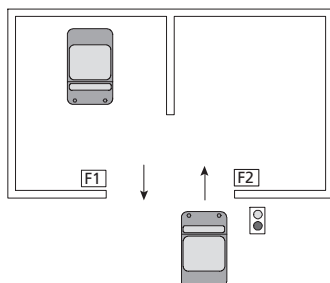


Fig. 1.13

Un'autorimessa dispone di due posti per autovetture. Il controllo delle autovetture in ingresso è fatto con una fotocellula  $F2$  e quello in uscita con la fotocellula  $F1$  secondo le specifiche della [tabella 1.18](#). L'ingresso delle autovetture nell'autorimessa è controllato da un semaforo con luce verde (posti disponibili) e luce rossa (posti esauriti). Inizialmente nel garage non sono presenti vetture. Si deve realizzare un sistema in grado di rilevare le vetture che entrano e che escono e che accenda il semaforo verde se sono disponibili posti e rosso se non lo sono.

Per la trattazione completa del sistema → Volume 1, Modulo 5, Unità 2, esempio 6. Si fa l'ipotesi che non possano essere intercettate contemporaneamente le fotocellule  $F1$  e  $F2$ .

TAB. 1.18

$F2$	$F1$	Stato del sistema
0	0	Nessuno veicolo in entrata o uscita
0	1	Veicolo in entrata
1	1	Veicolo in entrata e veicolo in uscita
1	0	Veicolo in uscita

TAB. 1.19

$U$	Semaforo	
0	Verde	Posti disponibili
1	Rosso	Posti non disponibili

### Realizzazione della rete con due flip-flop di tipo J-K

Il diagramma degli stati del sistema è rappresentato in [figura 1.14](#).

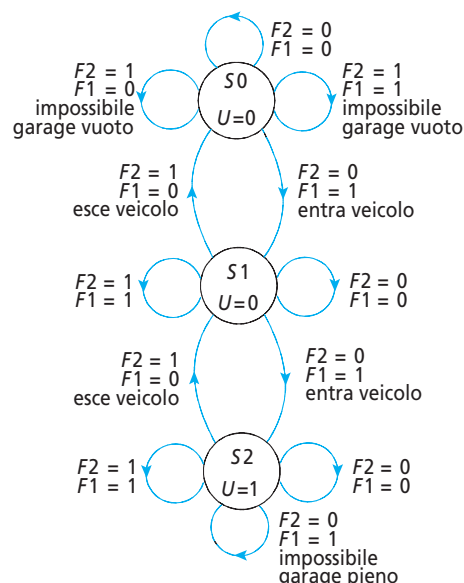


Fig. 1.14

Si deve osservare che il diagramma degli stati proposto soddisfa le specifiche del progetto solamente se il segnale fornito dalle fotocellule viene sincronizzato con il clock del sistema e rimane attivo per un periodo di tempo non superiore a quello del clock. Se questo non si verifica, il sistema si porta subito allo stato successivo.

Per evitare questo è necessario dotare ciascuna fotocellula di un circuito di sincronizzazione o elaborare un diagramma con nuovi stati che permettano l'evoluzione del sistema verso il nuovo stato solamente quando si disattiva il segnale fornito dalla fotocellula. In questa maniera il diagramma diventa sempre più complesso all'aumentare dei posti macchina di cui è dotato il garage. Procedendo come gli altri esempi, partendo dal diagramma si ricava la [tabella di transizione degli stati](#) ([tab. 1.20](#)) e poi, effettuando la sintesi con due flip-flop J-K (permettono di realizzare una rete combinatoria più semplice), si compilano le altre tabelle riportate.

Infine si realizzano le quattro mappe relative a  $J_1$ ,  $K_1$ ,  $J_0$  e  $K_0$ . L'uscita  $U$ , che attiva il semaforo rosso quando viene raggiunto lo stato  $S_2$ , è data da:

$$U = Q_0 Q_1$$

TAB. 1.20

Stato di partenza		$F_2 F_1$			
		0 0	0 1	1 1	1 0
S0[00]	⇒	S0[00]	S1[01]	S0[00]	S0[00]
S1[01]	⇒	S1[01]	S2[11]	S1[01]	S0[00]
S2[11]	⇒	S2[11]	S2[11]	S2[11]	S1[01]
S3[10]	⇒	S0[00]	S0[00]	S0[00]	S0[00]

TAB. 1.21

$Q_1(t)$		$F_2 F_1$				$Q_0(t)$		$F_2 F_1$			
		0 0	0 1	1 1	1 0			0 0	0 1	1 1	1 0
0	⇒	0	0	0	0	0	⇒	0	1	0	0
0	⇒	0	1	0	0	1	⇒	1	1	1	0
1	⇒	1	1	1	0	1	⇒	1	1	1	1
1	⇒	0	0	0	0	0	⇒	0	0	0	0

TAB. 1.22

Tabella transizione J-K				
$Q(t)$	⇒	$Q(t+1)$	J	K
0	⇒	0	0	X
0	⇒	1	1	X
1	⇒	0	X	1
1	⇒	1	X	0

TAB. 1.23

Stato di partenza		$F_2 F_1$							
		0 0		0 1		1 1		1 0	
$Q_1(t)$		$Q_1(t+1)$		$Q_1(t+1)$		$Q_1(t+1)$		$Q_1(t+1)$	
		J1	K1	J1	K1	J1	K1	J1	K1
0	⇒	0	X	0	X	0	X	0	X
0	⇒	0	X	1	X	0	X	0	X
1	⇒	X	0	X	0	X	0	X	1
1	⇒	X	1	X	1	X	1	X	1

TAB. 1.24

Stato di partenza		$F_2 F_1$							
		0 0		0 1		1 1		1 0	
$Q_0(t)$		$Q_0(t+1)$		$Q_0(t+1)$		$Q_0(t+1)$		$Q_0(t+1)$	
		J0	K0	J0	K0	J0	K0	J0	K0
0	⇒	0	X	1	X	0	X	0	X
1	⇒	X	0	X	0	X	0	X	1
1	⇒	X	0	X	0	X	0	X	0
0	⇒	0	X	0	X	0	X	0	X

$J_1$				
$F_2 F_1$ $Q_1 Q_0$	0 0	0 1	1 1	1 0
0 0	0	0	0	0
0 1	0	1	0	0
1 1	X	X	X	X
1 0	X	X	X	X

$$J_1 = \bar{F}_2 \cdot F_1 \cdot Q_0$$

$K_1$				
$F_2 F_1$ $Q_1 Q_0$	0 0	0 1	1 1	1 0
0 0	X	X	X	X
0 1	X	X	X	X
1 1	0	0	0	1
1 0	1	1	1	1

$$K_1 = F_2 \cdot \bar{F}_1 + \bar{Q}_0$$

$J_0$				
$F_2 F_1$ $Q_1 Q_0$	0 0	0 1	1 1	1 0
0 0	0	1	0	0
0 1	X	X	X	X
1 1	X	X	X	X
1 0	0	0	0	0

$$J_0 = \bar{F}_2 \cdot F_1 \cdot \bar{Q}_1$$

$K_0$				
$F_2 F_1$ $Q_1 Q_0$	0 0	0 1	1 1	1 0
0 0	X	X	X	X
0 1	0	0	0	1
1 1	0	0	0	0
1 0	X	X	X	X

$$K_0 = F_2 \cdot \bar{F}_1 \cdot \bar{Q}_1$$

Lo schema elettrico del sistema è rappresentato in [figura 1.15](#).

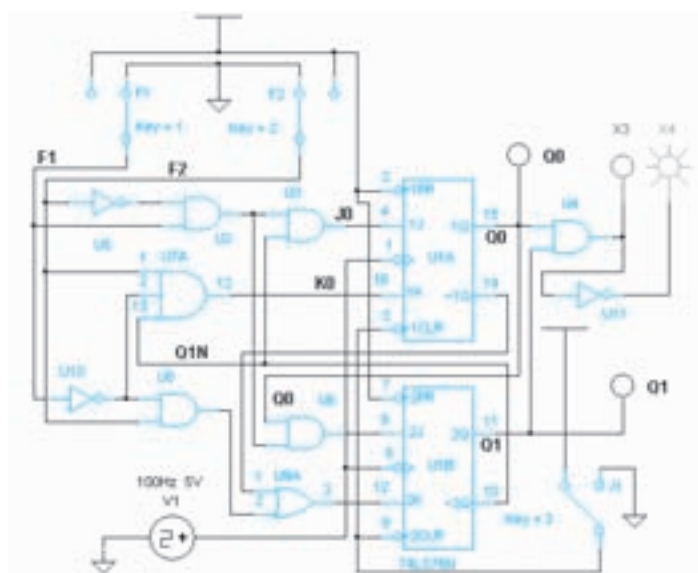


Fig. 1.15

Il verde del semaforo è attivo quando è spento il rosso. Nello schema sono posti anche due indicatori su  $Q_0$  e  $Q_1$  che permettono di controllare lo stato del sistema. Con il tasto numerico 3 si può portare il sistema allo stato iniziale.

### Esecuzione della simulazione

Il clock di sistema è stato posto a 100 Hz in modo da avere il tempo per disattivare i segnali delle fotocellule.

Se i livelli di  $F_1$  o  $F_2$  (attivati con i tasti numerici 1 e 2) sono tenuti troppo a lungo allo stato alto, il sistema cambia stato due volte portandosi allo stato successivo.

I passi da eseguire per la simulazione sono elencati di seguito.

1. Portare allo stato iniziale il sistema con il tasto numerico 3.
2. Portare a livello alto  $F_1$  finché non si raggiunge lo stato  $S_1$  ( $Q_1$  spento,  $Q_0$  acceso); è entrato un veicolo; il semaforo è verde. [Appena raggiunto lo stato  \$S\_1\$  riportare a livello basso  \$F\_1\$ .](#)

3. Portare a livello alto  $F_1$  finché non si raggiunge lo stato  $S_2$  ( $Q_1$  acceso,  $Q_0$  acceso); è entrato un secondo veicolo; il semaforo diviene rosso.
4. Portare a livello alto  $F_1$  finché non si raggiunge lo stato  $S_2$  ( $Q_1$  acceso,  $Q_0$  acceso); è entrato un secondo veicolo; il semaforo diviene rosso. Riportare a livello basso  $F_1$ .
5. Portare a livello alto  $F_2$  finché non si raggiunge di nuovo lo stato  $S_1$  ( $Q_1$  spento,  $Q_0$  acceso); è uscito un veicolo; il semaforo diviene verde. **Appena raggiunto lo stato  $S_1$  riportare a livello basso  $F_2$ .**
6. Portare a livello alto  $F_2$  finché non si raggiunge lo stato  $S_0$  ( $Q_1$  spento,  $Q_0$  spento); è uscito un altro veicolo, il garage è vuoto; il semaforo è verde.

### Circuito di sincronizzazione delle fotocellule

Per sincronizzare le fotocellule può essere ideato un sistema realizzato secondo il diagramma degli stati della [figura 1.16](#). Il sistema, sintetizzato secondo le procedure precedenti, utilizzando due flip-flop  $J$ - $K$ , a partire dalla [tabella](#) delle transizioni [1.25](#), porta alle seguenti relazioni per  $J$ ,  $K$  e  $U$ :

$$\begin{aligned} J_0 &= \bar{F}_2 \cdot F_1 \cdot \bar{Q}_1 & K_0 &= F_2 \cdot \bar{F}_1 \cdot \bar{Q}_1 \\ J_1 &= \bar{F}_2 \cdot F_1 \cdot Q_0 & K_1 &= F_2 \cdot \bar{F}_1 + \bar{Q}_0 \\ U_0 &= Q_0 \cdot \bar{Q}_1 \end{aligned}$$

Il clock del sistema deve essere fissato a 1 kHz. Dallo stato aggiunto  $S_3$  si è deciso di riportare il sistema a  $S_0$ .

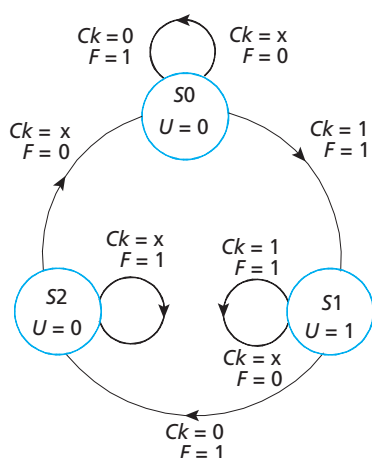


Fig. 1.16

TAB. 1.25

Stato di partenza		Ck F			
		0 0	0 1	1 1	1 0
S0[00]	⇒	S0[00]	S1[00]	S0[01]	S0[00]
S1[01]	⇒	S1[01]	S2[11]	S1[01]	S0[01]
S2[11]	⇒	S2[00]	S2[11]	S2[11]	S1[00]
S3[10]	⇒	S0[11]	S0[11]	S0[11]	S0[11]

Nella [figura 1.17](#) è visualizzato il circuito di sincronismo. Sono visibili i punti di I/O (PLACE ⇒ PLACE INPUT/OUTPUT) attraverso i quali verrà collegato alla precedente rete di figura 1.15 ad ognuna delle fotocellule ed al segnale di clock. Per non rendere troppo complessa la rappresentazione del sistema i due circuiti di sincronismo, ognuno per ciascuna fotocellula, sono inseriti come *subcircuit*.

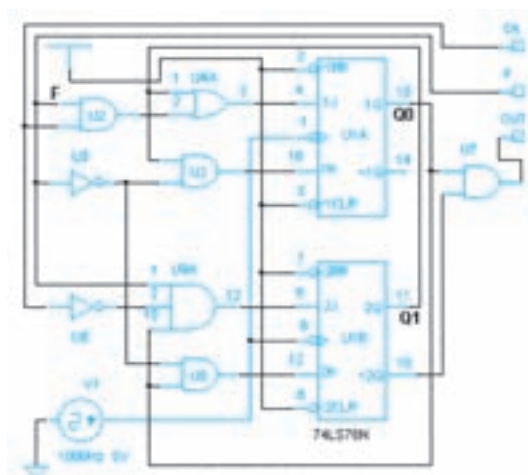


Fig. 1.17

Si deve procedere nel seguente modo:

1. realizzare e salvare il circuito di sincronismo di figura 1.17;
2. selezionarlo tutto e copiarlo (EDIT  $\Rightarrow$  SELECT ALL e poi EDIT  $\Rightarrow$  COPY);
3. aprire il circuito di figura 1.15;
4. posizionare il cursore in posizione esterna al circuito rappresentato e usare il menu PLACE  $\Rightarrow$  PLACE AS SUBCIRCUIT. Ripetere l'operazione per posizionare il secondo circuito di sincronismo;
5. connettere ai circuiti il clock e le due fotocellule  $F1$  e  $F2$  (rappresentate con degli switch) eliminando il precedente collegamento.

Nella figura 1.18 è rappresentato l'intero circuito.

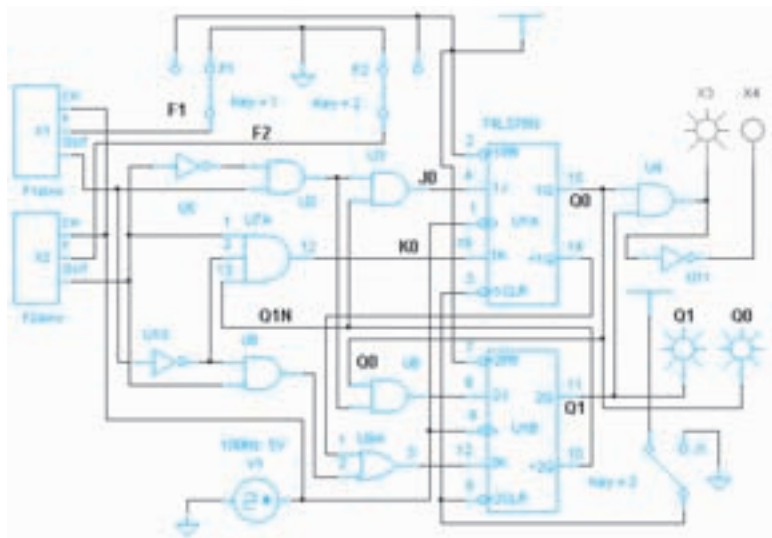


Fig. 1.18

### Esecuzione della simulazione

Nell'eseguire la simulazione si porti innanzitutto con il tasto numerico 3 il sistema allo stato iniziale ( $Q_1$  e  $Q_0$  spenti), riportando poi lo switch a livello alto.

Ogni volta che si attiva  $F1$  o  $F2$  attendere che il sistema cambi stato controllando  $Q_1$  e  $Q_0$  prima di riportare lo switch in posizione di riposo (livello basso).

## 1.2 ● Sintesi con automa di Mealy

### → Esempio - Riconoscitore di sequenze con automa di Mealy

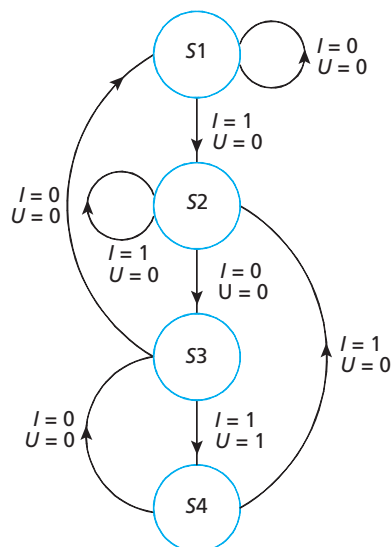


Fig. 1.19

Si propone ora la sintesi del riconoscitore della sequenza 101 realizzato con l'automa di Mealy e con flip-flop di tipo D. Sono sequenze considerate utili 101 101101 e 10101. Sequenza non utile 00. Il diagramma di stato per questo esercizio riportato in figura 1.19.

Come è possibile osservare dal diagramma, il sistema con quattro stati utili ha un ingresso  $I$  da cui entrano i bit della sequenza e un'uscita  $U$  che diviene alta quando è riconosciuta la sequenza utile.

Dal diagramma degli stati è poi possibile ricavare la tabella di transizione degli stati descritta nella tabella 1.26.

Come è possibile osservare dalla tabella 1.26, gli stati  $S_2$  e  $S_4$  (evidenziati) sono equivalenti ( $S_4 \equiv S_2$ ) e quindi possono essere fusi eliminando la riga dello stato  $S_4$  e sostituendo  $S_4$  (in colore) presente nello stato  $S_3$ , con  $S_2$ . Dopo le trasformazioni la tabella 1.26 si trasforma nella [tabella 1.27](#).

**TAB. 1.26**

Stato di partenza	Input $I$	
	0	1
$S_1 \Rightarrow$	$S_1/0$	$S_2/0$
$S_2 \Rightarrow$	$S_3/0$	$S_2/0$
$S_3 \Rightarrow$	$S_1/0$	$S_4/1$
$S_4 \Rightarrow$	$S_3/0$	$S_2/0$

**TAB. 1.27**

Stato di partenza	Input $I$	
	0	1
$S_1 \Rightarrow$	$S_1/0$	$S_2/0$
$S_2 \Rightarrow$	$S_3/0$	$S_2/0$
$S_3 \Rightarrow$	$S_1/0$	$S_2/1$

Si deve aggiungere poi di nuovo uno stato  $S_4'$ , per evitare lo stato di blocco del sistema, assumendo arbitrariamente che il sistema si porti da  $S_4'$  nello stato iniziale sia con  $I = 0$  sia con  $I = 1$ . Si ottiene allora la nuova [tabella 1.28B](#) dove in colore sono evidenziate le uscite. Quindi, in questo caso pur essendo avvenuta la fusione di due stati, non c'è stata un'effettiva riduzione di essi. Il sistema con quattro stati ha due variabili di stato,  $Q_1$  e  $Q_0$ , che possono essere associate arbitrariamente ai singoli stati, come riportato nella [tabella 1.28A](#).

**TAB. 1.28A**

Stato		
	$Q_1$	$Q_0$
$S_1$	0	0
$S_2$	0	1
$S_3$	1	1
$S_4'$	1	0

**TAB. 1.28B**

Stato di partenza	Input $I$	
	0	1
$S_1 \Rightarrow$	$S_1/0$	$S_2/0$
$S_2 \Rightarrow$	$S_3/0$	$S_2/0$
$S_3 \Rightarrow$	$S_1/0$	$S_2/1$
$S_4' \Rightarrow$	$S_1/0$	$S_1/0$

Sostituendo ai simboli degli stati ( $S_1, S_2, \dots$ ) i valori assunti dalle variabili di stato si ottiene la [tabella 1.29](#).

**TAB. 1.29**

Stato	$Q_1$	$Q_0$	Stato di partenza	Input $I$	
				0	1
$S_1$	0	0	$S_1 \Rightarrow$	00/0	01/0
$S_2$	0	1	$S_2 \Rightarrow$	11/0	01/0
$S_3$	1	1	$S_3 \Rightarrow$	00/0	01/1
$S_4'$	1	0	$S_4 \Rightarrow$	00/0	00/0

Scindendo poi per gli ingressi (colonne in grigio) i valori di  $Q_1(t+1)$  e  $Q_0(t+1)$  si ottiene la [tabella 1.30](#).

**TAB. 1.30**

Stato di partenza			Input $I$			
			0		1	
$Q_1(t)$	$Q_0(t)$		$Q_1(t+1)$	$Q_0(t+1)$	$Q_1(t+1)$	$Q_0(t+1)$
0	0	$\Rightarrow$	0	0	0	1
0	1	$\Rightarrow$	1	1	0	1
1	1	$\Rightarrow$	0	0	0	1
1	0	$\Rightarrow$	0	0	0	0

Per la rete d'uscita dalla tabella 1.29, prendendo in considerazione le sole uscite (colonne azzurre) si ha la [tabella 1.31](#).

Per la sintesi con i [flip-flop di tipo D](#) può essere usata direttamente la tabella 1.30 ricavando da essa le mappe di Karnaugh.

**TAB. 1.31**

			<i>U</i>	
$Q_1(t)$	$Q_0(t)$		$Q_1(t+1)$	$Q_0(t+1)$
0	0	$\Rightarrow$	0	0
0	1	$\Rightarrow$	0	0
1	1	$\Rightarrow$	0	1
1	0	$\Rightarrow$	0	0

	$I$	0	1
$Q_1 Q_0$			
S1	0 0	0	0
S2	0 1	1	0
S3	1 1	0	0
S4'	1 0	0	0

Da cui si ricava:

$$D_1 = \bar{Q}_1 Q_0 \bar{I}$$

	$I$	0	1
$Q_1 Q_0$			
S1	0 0	0	1
S2	0 1	1	1
S3	1 1	0	1
S4'	1 0	0	0

Da cui si ricava:

$$D_0 = \bar{Q}_1 Q_0 + I Q_0 + \bar{Q}_1 I$$

Per l'uscita *U* si ha la tabella:

	$I$	0	1
$Q_1 Q_0$			
S1	0 0	0	0
S2	0 1	0	0
S3	1 1	0	1
S4'	1 0	0	0

Da cui si ricava:

$$U = Q_1 Q_0 I$$

(si ricordi che nell'automa di Mealy le uscite dipendono anche dal valore assunto dagli ingressi).



# Realizzazione della rete progettata con due flip-flop di tipo D

Procedendo alla sintesi con i flip-flop di tipo D si ha la rete di [figura 1.20](#).

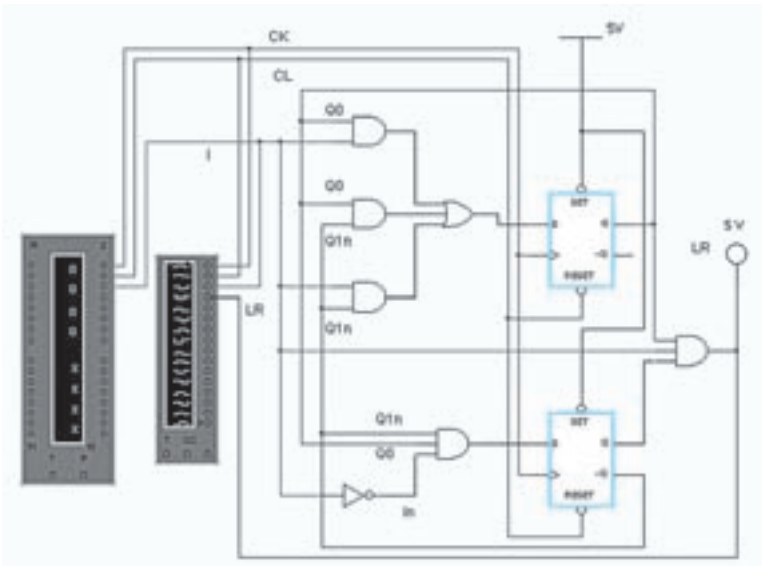


Fig. 1.20

Nella [tabella 1.32](#) sono elencate le sequenze di bit da impostare sul generatore di parole per eseguire la simulazione e nella [figura 1.21](#) è raffigurato lo strumento aperto in cui sono visibili le impostazioni da effettuare.

TAB. 1.32

N.	Pattern	I	CL	CK
00h	0000h	0	0	0
01h	0001h	0	0	1
02h	0002h	0	1	0
03h	0007h	1	1	1
04h	0006h	1	1	0
05h	0003h	0	1	1
06h	0006h	1	1	0
07h	0007h	1	1	1
08h	0002h	0	1	0
09h	0003h	0	1	1
0Ah	0006h	1	1	0
0Bh	0007h	1	1	1

N.	Pattern	I	CL	CK
0Ch	0002h	0	1	0
0Dh	0003h	0	1	1
0Eh	0002h	0	1	0
0Fh	0003h	0	1	1
10h	0006h	1	1	0
11h	0007h	1	1	1
12h	0006h	1	1	0
13h	0007h	1	1	1
14h	0006h	1	1	0
15h	0007h	1	1	1
16h	0006h	1	1	0
17h	0003h	0	1	1

N.	Pattern	I	CL	CK
18h	0002h	0	1	0
19h	0007h	1	1	1
1Ah	0006h	1	1	0
1Bh	0003h	0	1	1
1Ch	0002h	0	1	0
1Dh	0007h	1	1	1
1Eh	0006h	1	1	0
1Fh	0003h	0	1	1
20h	0002h	0	1	0
1h	0003h	0	1	1
22h	0002h	0	1	0
23h	0003h	0	1	1

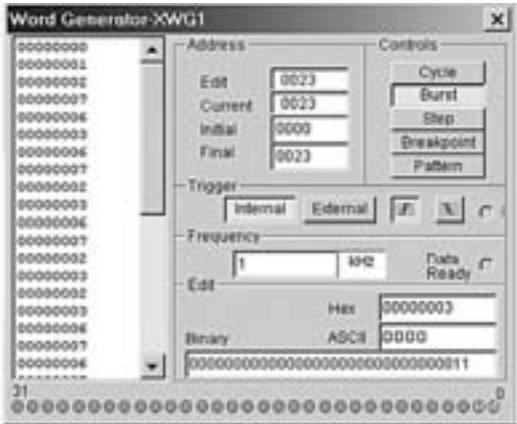


Fig. 1.21

Si tenga presente che nell'automa di Mealy, a differenza di quello di Moore, lo stato degli ingressi esterni influenza le uscite prima del fronte di salita del clock. Quindi, nella tabella 1.32 vanno presi in considerazione i livelli dell'ingresso prima che il clock passi dal livello basso a quello alto.

Con la simulazione si ottiene il diagramma di [figura 1.22](#). Per prima è riconosciuta la sequenza **101** e poi la sequenza concatenata **10101**.

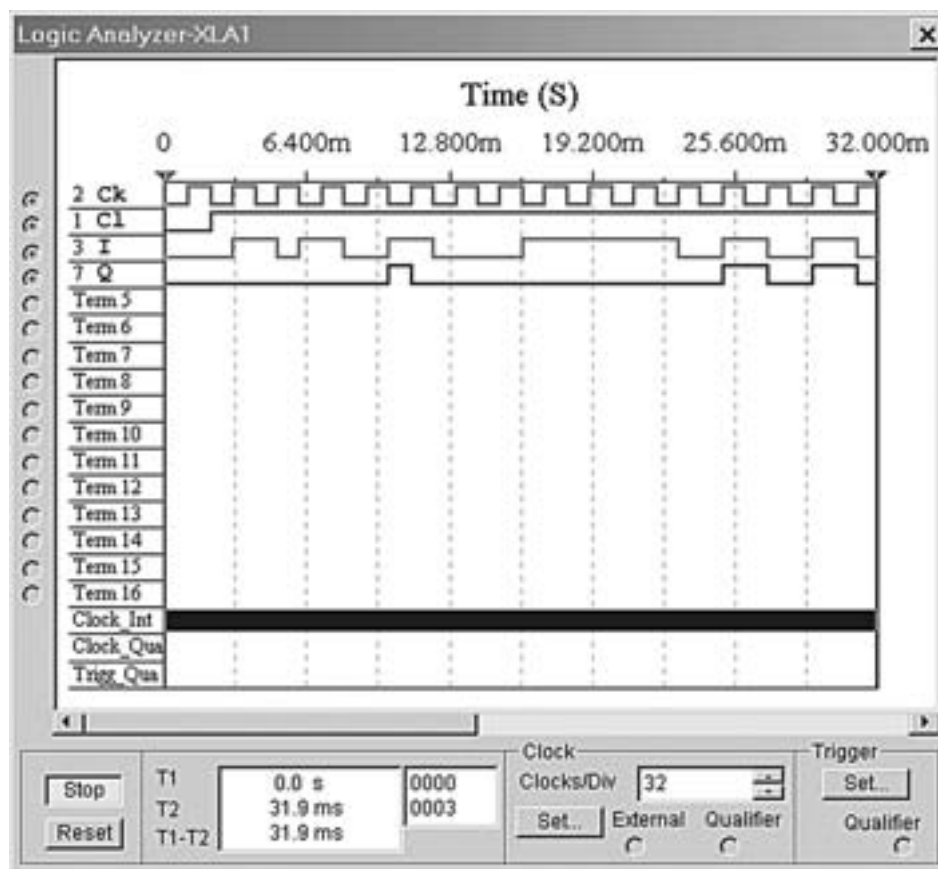


Fig. 1.22

Nota: i nomi dei segnali non sono visibili nella simulazione.

È possibile osservare che, trattandosi di un automa di Mealy, l'accensione del LED ROSSO avviene dopo il riconoscimento della sequenza (**101**), subito dopo la commutazione di  $I_1$  da livello basso a livello alto, prima che arrivi il fronte di salita del clock.



## 2. Reti combinatorie di I/O con dispositivi programmabili

### unità

Le reti combinatorie di ingresso e di uscita per la realizzazione dei sistemi sequenziali, ottenute con l'automa di Moore o di Mealy, possono essere realizzate anche utilizzando dispositivi programmabili come memorie EPROM o EEPROM, con dispositivi del tipo PAL (*Programmable Array Logic*), GAL (*Generic Array Logic*) o PLD (*Programmable Logic Device*) e simili (→ Modulo 1, Unità 3) o con altri dispositivi programmabili particolari.

### 2.1 ● Reti combinatorie realizzate con dispositivi programmabili di tipo EPROM

La EPROM è una memoria programmabile dall'utente (con apposito programmatore) e cancellabile con l'esposizione ai raggi ultravioletti (→ Modulo 3, Unità 1).

Servendosi di una EPROM, opportunamente programmata, si può sintetizzare qualsiasi funzione logica combinatoria.

Si faccia riferimento allo schema di una generica EPROM riportato in [figura 2.1](#).

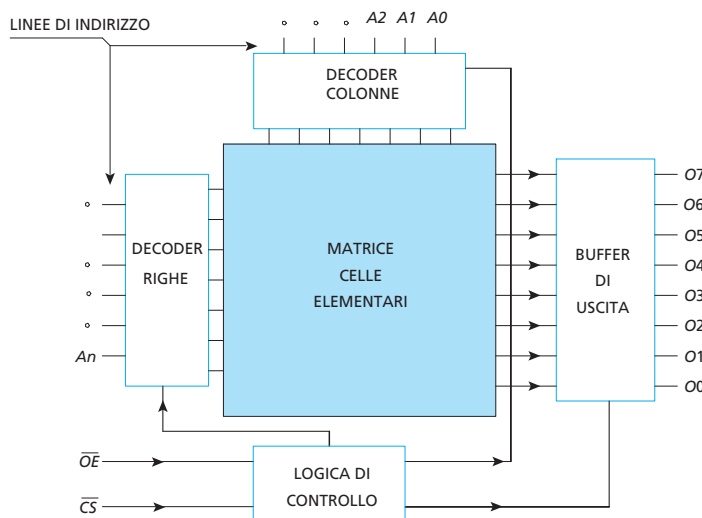


Fig. 2.1

Sulle linee d'indirizzo ( $A_n, \dots, A_2, A_1, A_0$ ) vanno collegati gli ingressi della rete combinatoria da realizzare.

Le uscite  $O_7, \dots, O_0$  della EPROM sono anche le uscite della rete combinatoria.

La programmazione del dispositivo dev'essere fatta realizzando una tabella della verità, in cui si associano alle linee d'indirizzo della memoria, gli ingressi della rete.

I valori che debbono avere le uscite della memoria, in corrispondenza ad ogni combinazione delle linee d'indirizzo, si ricavano direttamente dalle mappe di Karnaugh costruite durante il procedimento di sintesi.

Un esempio chiarirà quanto detto. Se la EPROM avrà un numero maggiore di linee d'indirizzo e/o dei dati, quelle eccedenti non verranno utilizzate.

## → Esempio 1

Si voglia realizzare la rete combinatoria d'ingresso dell'esempio 1 dell'Unità 1 di questo Modulo nel caso di sintesi con flip-flop di tipo D servendosi di una EPROM.

La rete combinatoria d'ingresso del sistema ha come ingressi le variabili di stato  $Q_1$ ,  $Q_0$  e l'ingresso esterno  $I$ .

Le uscite della rete  $D_1$  e  $D_0$  sono collegate agli elementi di memoria. Nella figura 2.2 è riportato lo schema della rete precedentemente realizzato (fig. 1.3) mettendo in evidenza gli ingressi e le uscite della rete combinatoria.

Lo schema è stato modificato, inserendo la NOT  $U_9$  (per invertire  $Q_1$ ), senza utilizzare il segnale già disponibile all'uscita del flip-flop  $U_7$ .

Nella figura 2.2 tutta la parte in grigio verrà sostituita da una EPROM programmata, come mostrato nella figura 2.3.

Nello schema non sono riportate le connessioni per l'alimentazione della memoria. I pin  $\overline{OE}$  e  $\overline{CS}$  sono tenuti a livello basso per abilitare l'uscita dei dati sulle linee  $D_1$  e  $D_0$ .

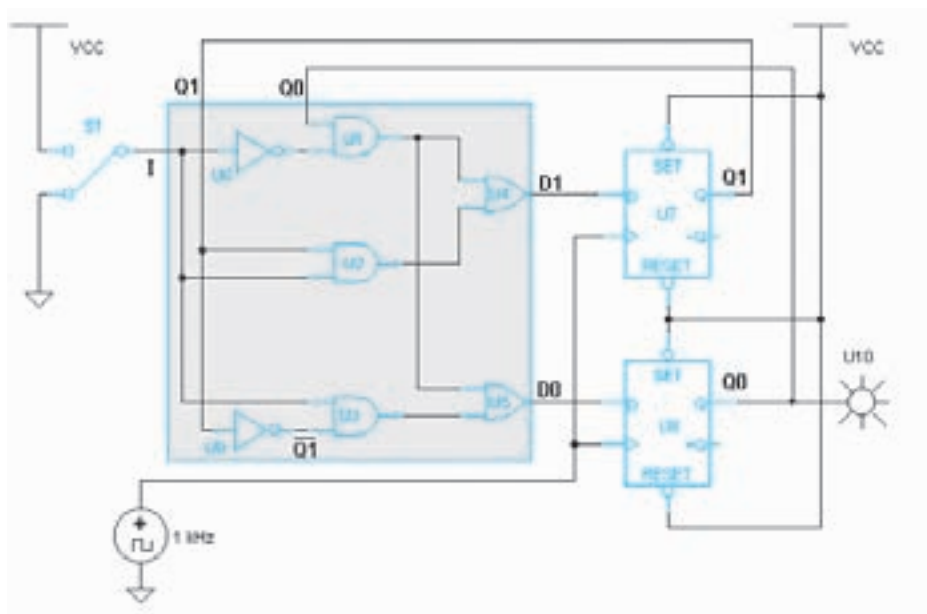


Fig. 2.2

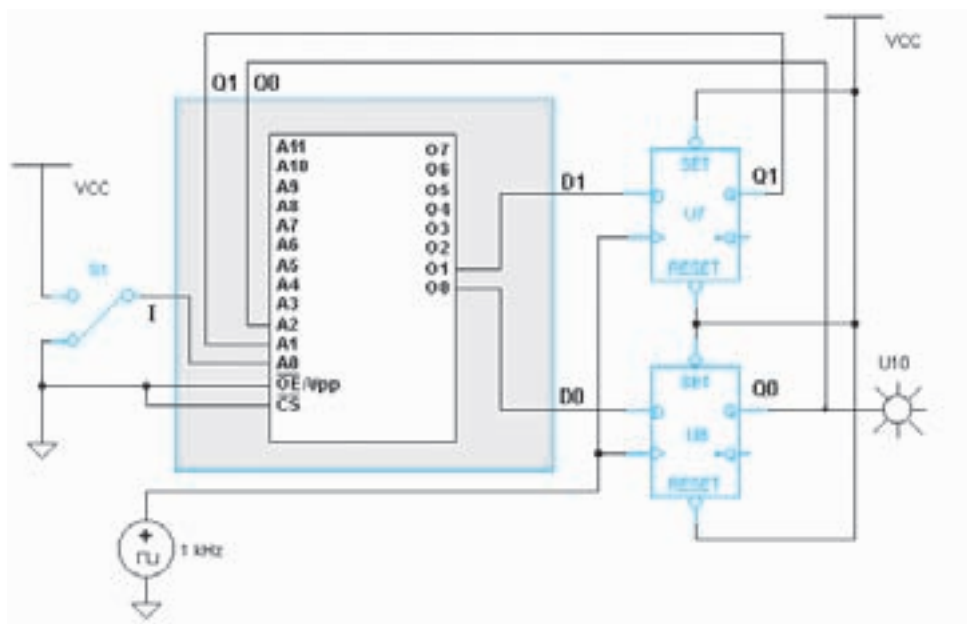


Fig. 2.3

## Programmazione della EPROM

Dalle mappe dell'esempio 1 dell'Unità 1, qui ripetute per comodità, si ricava la tabella della verità di seguito riportata.

$D_1$			$D_0$		
$Q_1 \backslash Q_0 \backslash I$	0	1	$Q_1 \backslash Q_0 \backslash I$	0	1
0 0	0	0	0 0	0	1
0 1	1	0	0 1	1	1
1 1	1	1	1 1	1	0
1 0	0	1	1 0	0	0

TAB. 2.1

$Q_1$	$Q_0$	$I$	$D_1$	$D_0$
$A_2$	$A_1$	$A_0$	$O_1$	$O_0$
0	0	0	0	0
0	0	1	0	1
0	1	0	1	1
0	1	1	0	1
1	0	0	0	0
1	0	1	1	0
1	1	0	1	1
1	1	1	1	0

Nella seconda riga della tabella sono elencati gli accoppiamenti tra gli ingressi della rete ( $Q_1$ ,  $Q_0$ ,  $I$ ) e le linee d'indirizzo della EPROM ( $A_2$ ,  $A_1$ ,  $A_0$ ) e gli accoppiamenti tra le uscite della rete ( $D_1$ ,  $D_0$ ) e le uscite della EPROM ( $O_1$ ,  $O_0$ ).

La programmazione del dispositivo deve essere fatta con opportuno programmatore, inviando sulle linee d'indirizzo i valori presenti sulla tabella e ponendo, per ogni indirizzo selezionato, sulle linee dati  $O_1$ ,  $O_0$  i codici corrispondenti presenti.

## Simulazione dell'automa con una EPROM

Per controllare il funzionamento dell'automa con rete combinatoria realizzata per mezzo di un dispositivo programmabile, è stata costruita con MultiSIM una rete logica che simula il funzionamento di una memoria programmata. La memoria è realizzata con il demultiplexer 74LS138 e con 16 buffer three-state presenti in quattro integrati 74LS125. La 74LS138, per ogni combinazione presente sulle linee  $A$ ,  $B$  e  $C$ , porta bassa una delle uscite  $Y_0 \dots Y_7$ , ciascuna delle quali a sua volta attiva una coppia dei buffer three-state fornendo il codice per  $D_1$  e  $D_0$ . Gli ingressi delle coppie di buffer vanno collegate a sorgenti di tensione con livello alto ( $V_{cc}$ ) o basso ( $GND$ ) in base alla tabella della verità. Per esempio, con gli ingressi  $A$ ,  $B$  e  $C$  tutti a livello alto (ultima riga della tabella della verità) diviene basso  $Y_7$  (tutte le altre uscite sono a livello alto) e i due buffer con il controllo collegato a questa uscita forniranno per  $D_1$  livello alto (l'ingresso del buffer è connesso a  $V_{cc}$ ) e per  $D_0$  livello basso (ingresso del buffer connesso a  $GND$ ).

Nello schema di figura 2.4 sono stati aggiunti tre switch e tre indicatori luminosi, collegati con le linee d'indirizzo, e due indicatori posti sulle uscite, per verificare il funzionamento del circuito. Facendo assumere alle linee d'indirizzo tutte le combinazioni della tabella della verità, le uscite dovranno seguire i livelli assunti da  $O_1$  e  $O_0$ .

Cambiando i collegamenti sugli ingressi dei buffer è possibile variare le uscite della tabella della verità, per la sintesi di un automa diverso, purché la rete combinatoria non abbia più di tre ingressi e due uscite. È possibile aumentare il numero delle uscite aggiungendo coppie di 74LS125. Usando il 74154 (o il 74HCT154), demultiplexer 4 a 16, è possibile implementare una memoria con quattro ingressi.

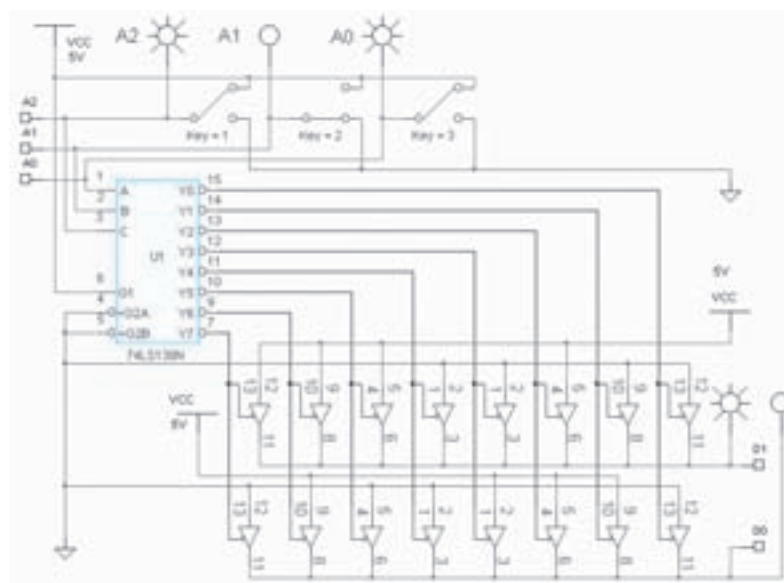


Fig. 2.4

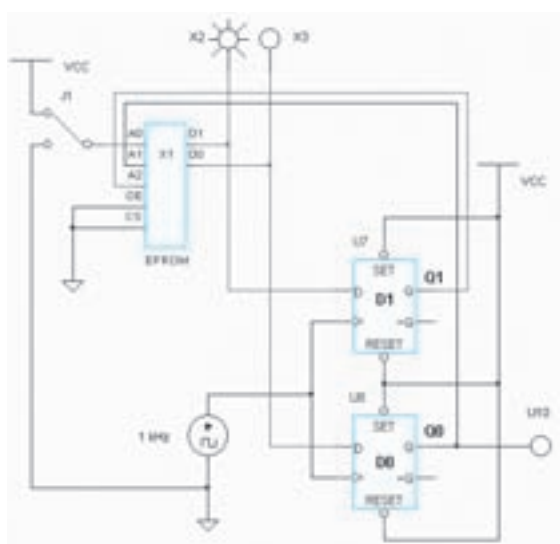


Fig. 2.5

Nella costruzione dell'automata, usando la memoria realizzata, bisogna eliminare dallo schema di figura 2.4 i tre deviatori e i cinque indicatori luminosi lasciando i connettori di I/O  $A_2$ ,  $A_1$ ,  $A_0$  e  $D_1$ ,  $D_0$ .

Nella figura 2.5 è rappresentato l'intero circuito dell'automata con la memoria salvata come Subcircuit (→ Circuito di sincronizzazione delle fotocellule nell'esempio 4 dell'Unità 1).

Per controllare il funzionamento del circuito sono stati aggiunti i due indicatori luminosi  $X_2$  e  $X_3$  che visualizzano lo stato delle linee di uscita della memoria. Avviando la simulazione il led  $U_{10}$  deve accendersi (se è spento) o spegnersi (se è acceso) ogni volta che il deviatore fornisce un livello che passa dallo stato basso a quello alto.



## → Esempio 2

Si riprenda l'automata dell'esempio 3 dell'Unità 1, e lo si realizzi con un flip-flop di tipo D (invece che J-K).

Dalla tabella 2.2 (corrispondente alla tabella 1.15 dell'Unità 1), ricordando che per un flip-flop di tipo D la tabella di transizione e quella di eccitazione coincidono, si ricava direttamente la mappa di Karnaugh e da essa la tabella della verità (tab. 2.3).

TAB. 2.2

Stato di partenza	$F_2 F_1$			
	0 0	0 1	1 1	1 0
S0 [0] ⇒	S0 [0]	S1 [1]	S0 [0]	S0 [0]
S1 [1] ⇒	S0 [0]	S1 [1]	S1 [1]	S1 [1]

$Q(t)$	$F_2 F_1$			
	0 0	0 1	1 1	1 0
0 ⇒	0	1	0	0
1 ⇒	0	1	1	1

Si programma di nuovo la EPROM in base alla tabella 2.3 e si collega all'elemento di memoria (flip-flop di tipo D).  
Si ottiene lo schema di figura 2.6.

D <sub>0</sub>				
F <sub>2</sub> F <sub>1</sub>	0 0	0 1	1 1	1 0
Q				
0	0	1	0	0
1	0	1	1	1

TAB. 2.3

Q	F <sub>2</sub>	F <sub>1</sub>	D <sub>0</sub>
A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	O <sub>0</sub>
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

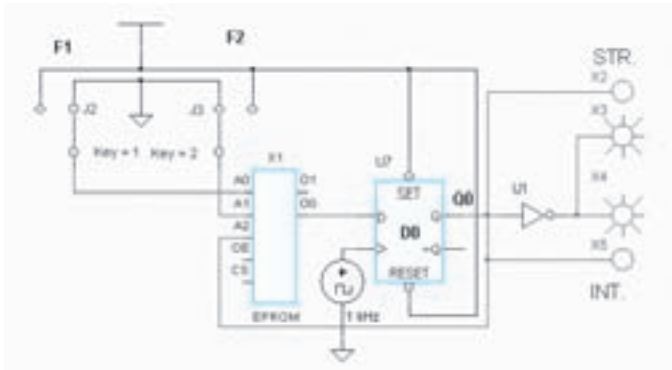


Fig. 2.6

Come è possibile dedurre dagli esempi precedenti, gli automi a stati finiti realizzati possono anche essere considerati sistemi programmabili che possono svolgere compiti diversi modificando solamente la programmazione della memoria.

Nel primo esempio fatto, con due flip-flop, modificando i dati presenti nella EPROM, si possono realizzare reti sequenziali diverse con un massimo di quattro stati. Il numero di ingressi massimi (esterni più interni) è invece legato al numero di linee d'indirizzo della EPROM. Nel secondo esempio, essendo presente un solo flip-flop, cambiando i dati presenti nella memoria, è possibile realizzare reti sequenziali diverse con due stati. Si può anche affermare in linea generale che gli ingressi esterni forniscano un codice formato da un certo numero di bit (tanti quanti sono le linee d'ingresso esterne) e il sistema esegue delle azioni, all'arrivo del segnale di clock, portandosi da uno stato all'altro, legate sia al codice ricevuto che ai livelli presenti sugli ingressi interni, prima dell'arrivo del segnale di clock. Negli esempi degli automi 2, 3 e 4 dell'Unità 1, sono presenti due ingressi esterni e quindi il codice fornito al sistema è a due bit. Con questo codice a due bit possono essere implementati quattro diversi comandi da inviare all'automa. Aumentando il numero d'ingressi e il numero di stati (inserendo più flip-flop) si ottengono sistemi programmabili che possono accettare in ingresso molti più codici e che possono svolgere più azioni diverse passando da uno stato all'altro. La memoria programmata (EPROM o altro), contenente i dati, è l'elemento che, in base ai codici ricevuti e allo stato attuale del sistema, porta l'automa in uno stato o nell'altro. In linea del tutto generale si può dire che si sono realizzati i primi passi per costruire un dispositivo programmabile in grado di eseguire funzioni diverse in base ai codici ricevuti.

# 3. Dispositivi logici programmabili

## unità

I **dispositivi logici programmabili** sono dei circuiti integrati in cui può essere definita la funzione logica che determina i valori dei segnali di uscita. Tale funzione può essere combinatoria o sequenziale e più o meno complessa a seconda del tipo e del modello di circuito programmabile utilizzato.

I dispositivi logici programmabili sono stati ideati con l'obiettivo di sostituire i circuiti logici standard perché con un unico chip è possibile realizzare la funzione svolta da numerosi dispositivi standard. Questi dispositivi ad elevata scala di integrazione vengono utilizzati per realizzare prototipi di circuiti *custom*<sup>1</sup> (grazie alla programmabilità è possibile riconfigurare facilmente il circuito per correggere errori di programmazione e di progetto). Inoltre, quando si lavora con bassi volumi di produzione, i circuiti programmabili possono sostituire convenientemente i circuiti custom stessi.

## 3.1 Dispositivi PLA, PAL e GAL

Un **dispositivo PLA** è costituito da una matrice di porte AND d'ingresso ed una matrice di porte OR d'uscita entrambe programmabili (fig. 3.1).

Il piano delle porte AND (AND array) definisce per ciascuna porta AND quali sono i segnali d'ingresso collegati. Il piano delle porte OR, invece, definisce i termini di prodotto che devono essere collegati a ciascuna porta OR.

Programmando le connessioni, indicate con il pallino, è possibile, in teoria, realizzare per ciascuna uscita qualsiasi funzione sintetizzabile come somma di prodotti (fig. 3.2).

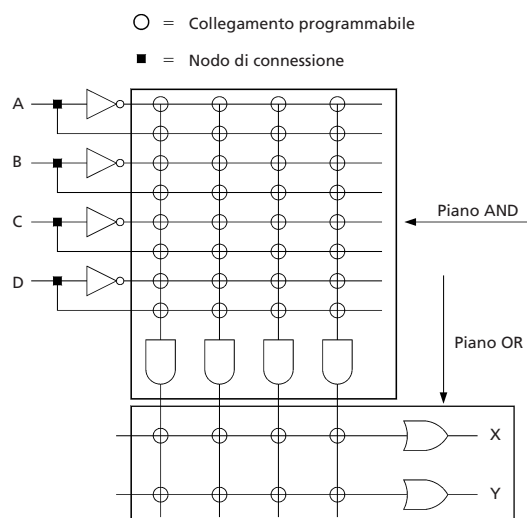


Fig. 3.1

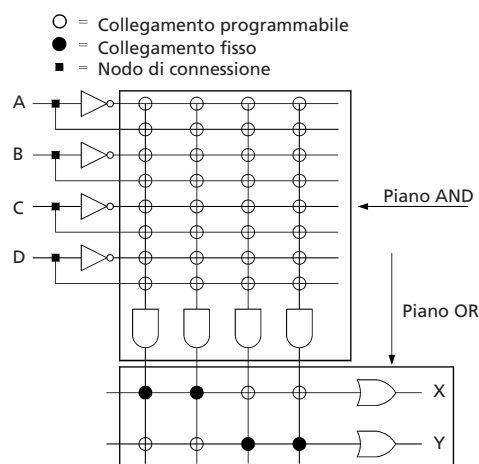


Fig. 3.2

<sup>1</sup> Un **circuito custom** è un circuito integrato progettato appositamente per realizzare una determinata funzione.



In realtà il tipo di funzione è limitato dalle risorse del dispositivo PLA utilizzato: il numero di ingressi, quello delle uscite e le dimensioni del piano delle AND.

Un piano con quattro porte AND, ad esempio, consente di realizzare solo funzioni che richiedono un numero massimo di quattro termini di prodotto.

I **dispositivi PAL** (*Programmable Array Logic*) sono sicuramente i circuiti programmabili più utilizzati. La struttura dei dispositivi PAL, derivata da quella delle PLA, è basata sul fatto che qualsiasi funzione logica può essere descritta tramite somme di prodotti logici. Un dispositivo PAL presenta una matrice d'ingresso costituita da porta AND collegate ad una serie di OR le cui uscite sono indirizzate sui pin d'uscita del dispositivo. La matrice delle AND è programmabile e, pertanto, consente di effettuare il prodotto logico con qualsiasi combinazione di ingressi primari. La struttura delle OR, invece, è fissa e non consente di indirizzare l'uscita di una AND su una qualsiasi OR, ma è necessario rispettare una certa configurazione. Per ciascuna porta OR, infatti, sono associate un certo numero di porte AND (fig. 3.3).

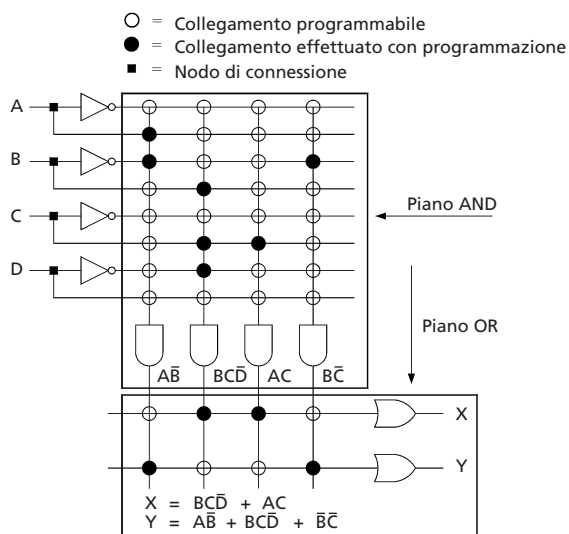


Fig. 3.3

Questo implica che per ciascuna porta OR sono disponibili un numero più ristretto di termini di prodotto rispetto al numero totale di porte AND. Considerato, inoltre, che le porte OR hanno associate porte AND distinte, è necessario duplicare i termini OR quando sono richiesti termini di prodotto identici su porte OR distinte. Tale scelta è motivata principalmente dal fatto che nella maggior parte delle situazioni sono sufficienti un numero di termini di prodotto limitati, per cui l'architettura delle PAL risulta essere più compatta e veloce.

Le PAL esistenti attualmente in commercio hanno un'architettura più complessa di quella indicata in figura 3.3 perché sono presenti porte three-state per controllare lo stato di alta impedenza, flip-flop per realizzare funzioni sequenziali e linee di reazione che riportano le uscite in ingresso consentendo, in tal modo, di realizzare funzioni con numerosi termini di prodotto.

I dispositivi PAL di vecchia concezione che consentivano di implementare funzioni combinatorie con uscite attive basse ( $L = Low$ ) o alte ( $H = High$ ) oppure sequenziali con flip-flop sull'uscita ( $R = Registered$ ) sono state soppiantate da **dispositivi GAL** (*Generic Array Logic*) nei quali è presente, oltre ai piani AND ed OR, anche la logica necessaria per gestire le varie configurazioni. Il blocco logico associato a ciascuna uscita costituisce una **macrocella**.

I dispositivi GAL possono essere considerati circuiti PAL universali in quanto consentono di configurare le singole uscite in modalità combinatoria oppure sequenziale, sono compatibili con i vecchi dispositivi PAL ed hanno una maggiore flessibilità rispetto ad essi. Utilizzando un unico dispositivo GAL, infatti, è possibile realizzare funzioni logiche di media complessità.

La sigla di identificazione di un dispositivo PAL riporta il numero degli ingressi, quello delle uscite ed il tipo di dispositivo. La sigla PAL22V10, ad esempio, indica che il dispositivo ha 22 ingressi, 10 uscite ed è di tipo V (*Versatile*).

Il numero di ingressi e quello delle uscite rappresentano il numero massimo di pin disponibili, in quanto ogni singolo pin del chip può essere configurato come ingresso, uscita o bidi-

reazionale. Se 20 pin nel dispositivo PAL22V10 sono configurati come ingressi, solo i restanti 2 possono essere configurati come uscite. Generalmente il dispositivo che dispone del numero di risorse sufficienti per realizzare la funzione descritta dal progettista è individuato in modo automatico da un programma CAD dedicato.

Si tenga presente che l'architettura delle PAL di tipo V è molto simile a quella delle GAL.

## 3.2 ● Dispositivi PLD, CPLD e FPGA

Poiché la dimensione dei dispositivi PAL cresce con il numero di ingressi sul piano AND, degradando la velocità di risposta del circuito, risulta evidente l'impossibilità di realizzare circuiti PAL di grande complessità.

Questo limite viene aggirato dai **dispositivi PLD** (*Programmable Logic Device*) i quali sono costituiti da tante piccole PAL, dette **blocchi logici**, aventi il piano delle AND di dimensioni ridotte e collegate tra di loro da un bus globale. Un dispositivo PLD è costituito, quindi, da macrocelle locali che dispongono solo di collegamenti interni all'interno dello stesso blocco logico e da macrocelle globali che hanno accesso ai collegamenti sul bus globale che collega i vari blocchi.

I **dispositivi CPLD** (*Complex Programmable Logic Device*), simili in linea di principio ai PLD, contengono un numero elevato di blocchi logici e di macrocelle e necessitano di strutture più complesse e flessibili per l'interconnessione dei singoli blocchi logici. Di conseguenza, al posto del bus globale vi è un blocco di interconnessione programmabile.

Gli **FPGA** (*Field Programmable Gate Array*) presentano un'architettura interna costituita da tanti piccoli blocchi logici disposti a matrice. Questi blocchi logici sono configurabili e le connessioni tra un blocco e l'altro vengono effettuate tramite dei canali configurabili e dei blocchi di scambio. In corrispondenza di ciascun pin vi sono dei blocchi speciali che gestiscono la funzione del pin stesso (input, output, bidirezionale).

Si tenga presente che è difficile definire un'architettura generale dei dispositivi FPGA perché ciascun costruttore adotta soluzioni che risultano più efficienti in alcune applicazioni piuttosto che in altre.

I dispositivi FPGA possono essere considerati dei gate array programmabili perché la loro struttura è particolarmente adatta all'implementazione di dispositivi complessi.

Si ricorda che i gate array sono dei circuiti di tipo semi-custom che presentano al loro interno una struttura predefinita di transistor CMOS.

Il progettista, tramite software CAD, definisce i collegamenti necessari per realizzare funzioni sequenziali e combinatorie. Essenzialmente i gate array sono dei circuiti semi-lavorati dove la struttura CMOS è prefabbricata. Il progettista definisce le connessioni (per realizzare la funzione desiderata) che serviranno per completare il processo di fabbricazione del circuito.

Considerato che per la descrizione ad alto livello dei circuiti FPGA si usano gli stessi linguaggi di descrizione dell'hardware utilizzati per i gate array e per i circuiti custom in generale, risulta particolarmente agevole convertire un dispositivo FPGA in un circuito custom. Per tale motivo le applicazioni tipiche di questi dispositivi sono la realizzazione di prototipi di chip custom o il loro utilizzo come ASIC (*Application Specific Integrated Circuit*) per bassi volumi di produzione.

## 3.3 ● Programmazione dei dispositivi

La programmazione dei dispositivi PLA, PAL, PLD, CPLD e FPGA può avvenire in diversi modi a seconda della tecnologia utilizzata per la loro realizzazione.

1. **Dispositivi fuse ed antifuse:** la programmazione si realizza applicando delle tensioni adatte a fondere i **microfusibili** presenti all'interno del circuito. In questo modo vengono effettuati (*antifuse*) o interrotti (*fuse*) dei collegamenti all'interno del circuito e non è possibile cambiare la funzione programmata perché il processo di programmazione è distruttivo.

2. **Dispositivi EPROM ed EEPROM:** la programmazione si realizza programmando le memorie contenute all'interno del dispositivo. Ciascun bit della memoria controlla lo stato di un transistor che, funzionando come interruttore, stabilisce o meno dei collegamenti all'interno del circuito. Questo tipo di tecnica consente di riprogrammare la funzione del circuito garantendo una maggiore flessibilità di utilizzo.
3. **Dispositivi RAM:** si utilizza una memoria volatile per configurare il dispositivo. In questo caso si ha una maggiore flessibilità poiché la programmazione può essere riconfigurata durante il funzionamento. Tuttavia, la perdita del contenuto della memoria allo spegnimento del sistema implica la presenza di un metodo di inizializzazione che carichi il contenuto della RAM all'accensione.

Il progettista definisce la funzione logica da realizzare con programmi CAD che consentono di definire e simulare il funzionamento del circuito.

In genere la descrizione della funzione da realizzare può avvenire:

1. tramite descrizione grafica della funzionalità (*schematic entry*): si disegna uno schema elettrico utilizzando delle primitive (AND, OR, flip-flop, contatori, ecc.) che collegate tra di loro realizzano la funzione desiderata;
2. tramite linguaggi HDL (*Hardware Description Language*) che descrivono in maniera astratta il funzionamento del circuito.

In entrambi i casi il programma CAD realizza la sintesi e la mappatura sul dispositivo da programmare dalla descrizione del circuito, e ricava la configurazione dei fusibili o della memoria. La programmazione effettiva dei dispositivi avviene tramite appositi circuiti programmatori simili ai programmatori di memorie EPROM.

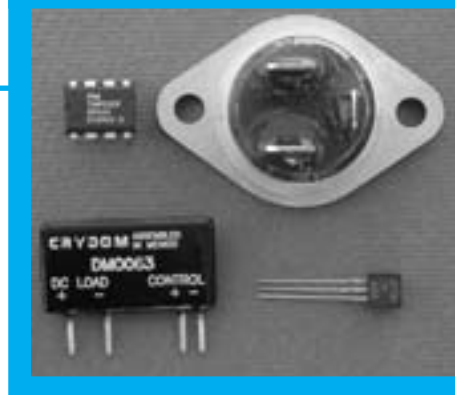
## Esercizi proposti

### Unità 1

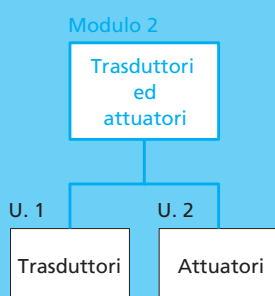
- 1** Utilizzando l'automa di Moore, si realizzi la rete logica del 1° esempio, usando flip-flop di tipo  $T$  (sono già state trovate le funzioni logiche di  $T_0$  e  $T_1$ ). Se ne esegua la simulazione usando il generatore di parole e l'analizzatore di stati logici di MultiSIM.
- 2** Utilizzando l'automa di Moore, si realizzi la rete logica del 1° esempio usando flip-flop di tipo  $J-K$  (sono già state trovate le funzioni logiche di  $J_0$ ,  $K_0$  e  $J_1$ ,  $K_1$ ) e se ne esegua la simulazione.
- 3** Utilizzando l'automa di Moore, si realizzi la rete logica del 2° esempio usando flip-flop di tipo  $J-K$  e se ne esegua la simulazione.
- 4** Utilizzando l'automa di Moore, si realizzi la rete logica del 3° esempio usando flip-flop di tipo  $D$  e se ne esegua la simulazione.
- 5** Si tracci il diagramma di stato secondo Moore per l'autorimessa del 4° esempio (si utilizzino comunque due fotocellule distinte per l'ingresso e l'uscita). Si deve evitare, senza ricorrere a particolari restrizioni del clock o alla sincronizzazione delle fotocellule, che il sistema non cambi stato, anche se il segnale emesso dalle fotocellule rimane attivo, fin quando esso si porta di nuovo a livello basso. Si realizzi poi l'automa usando flip-flop di tipo  $D$  e se ne esegua la simulazione.
- 6** Si trasformi il diagramma di figura 1.2 in un diagramma di Mealy. Si realizzi poi la rete logica relativa al diagramma con flip-flop di tipo  $D$  e se ne esegua la simulazione.
- 7** Si trasformi il diagramma di figura 1.14 in un diagramma di Mealy. Si realizzi poi la rete logica relativa al diagramma con flip-flop di tipo  $D$  e se ne esegua la simulazione.
- 8** Si realizzi la rete logica sequenziale descritta dal diagramma degli stati di figura 1.19 (riconoscitore di sequenze) con l'automa di Moore utilizzando flip-flop di tipo  $D$ . Se ne esegua la simulazione usando il generatore di parole e l'analizzatore di stati logici di MultiSIM.

# Modulo 2. Trasduttori ed attuatori

- ▶ UNITÀ 1. Trasduttori
- ▶ UNITÀ 2. Attuatori



UCATION SPA



## PREREQUISITI

- ▶ Conoscenza delle unità di misura
- ▶ Conoscenza delle regole di risoluzione delle reti elettriche lineari e non lineari
- ▶ Conoscenza delle caratteristiche funzionali dei diodi, dei transistor e degli amplificatori operazionali

## OBIETTIVI

- ▶ Conoscere le caratteristiche di alcuni trasduttori
- ▶ Saper scegliere il trasduttore adatto per il rilievo della grandezza fisica
- ▶ Saper schematizzare a blocchi un sistema per il rilievo di una specifica grandezza fisica
- ▶ Saper dimensionare i circuiti di linearizzazione
- ▶ Saper dimensionare il circuito per il rilievo di una specifica grandezza fisica
- ▶ Conoscere gli azionamenti per uno specifico attuatore
- ▶ Saper strutturare un circuito pilota per un attuatore

# 1. Trasduttori

## unità

Un buon trasduttore deve avere una caratteristica  $I/U$  lineare e quindi la funzione  $U = f(I)$  che la rappresenta deve essere quella di una retta affinché il rilievo della grandezza fisica abbia una buona accuratezza.

Considerato che molti trasduttori non presentano una caratteristica Ingresso/Uscita lineare, è necessario provvedere alla linearizzazione della caratteristica.

Una fotoresistenza alimentata da un generatore di corrente costante  $I$  (fig. 1.1), ad esempio, ha una caratteristica esponenziale decrescente fortemente non lineare, in quanto la tensione d'utilizzazione  $V = R(Lux) \cdot I$  diminuisce in modo esponenziale all'aumentare della luminosità.

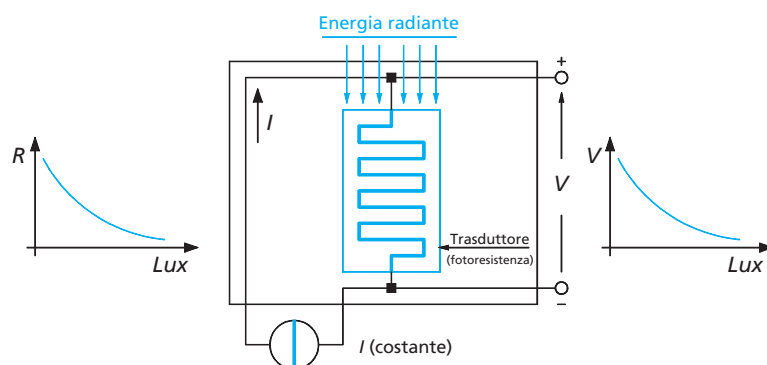


Fig. 1.1

Poiché la non linearità della caratteristica *Tensione/Lux* (fig. 1.2) non dà proporzionalità tra la luminosità (grandezza fisica controllata) e la tensione (grandezza elettrica d'uscita) è necessario, come detto, **linearizzare** la caratteristica, ossia correggerla in modo da renderla lineare affinché il sistema di misura sia ad elevata precisione.

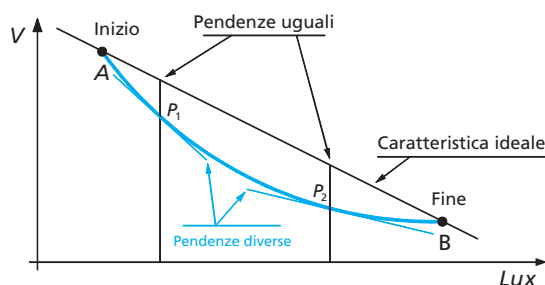


Fig. 1.2

Il processo di linearizzazione può essere fatto utilizzando tecniche diverse e più o meno sofisticate, sia in via software sia con circuiti hardware.

- Nel primo caso si opera con appositi programmi oppure correggendo i valori già acquisiti, confrontandoli con altri preventivamente registrati in una tabella posta in memoria.
- Nel secondo caso si utilizzano componenti quali amplificatori particolari o circuiti che hanno una curva di risposta complementare a quella del trasduttore.

## 1.1 ● Linearizzazione caratteristica dei trasduttori

Nella figura 1.2 è riportata la caratteristica di un trasduttore non lineare.

Se si uniscono il punto iniziale e quello finale della caratteristica, si ottiene una retta con pendenza in ogni punto costante (fig. 1.3).

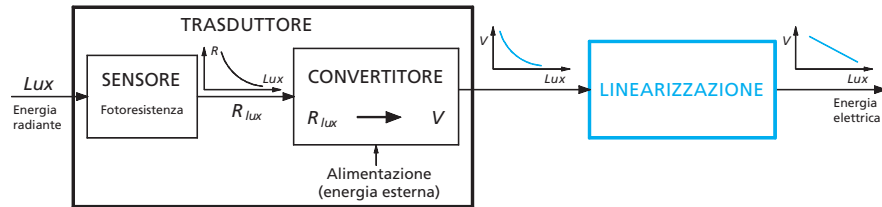


Fig. 1.3

Per linearizzare la caratteristica esponenziale di un trasduttore di temperatura a variazione di resistenza (fig. 1.4), si utilizza una resistenza di linearizzazione  $R_L$  collegata in serie alla resistenza  $R_T$  del trasduttore (fig. 1.5).

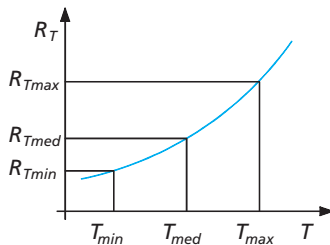


Fig. 1.4

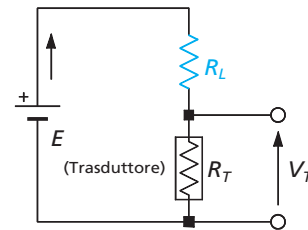


Fig. 1.5

Lo schema di figura 1.5 converte la variazione della resistenza  $R_T$ , provocata dalla variazione di temperatura  $T$ , in una variazione di tensione  $V_T$  utilizzabile nei circuiti elettronici. Si osservi che il circuito proposto fornisce una nuova caratteristica tra la tensione  $V_T$  e la temperatura  $T$ .

Per calcolare la resistenza di linearizzazione  $R_L$  è necessario fissare il range di temperatura  $T$  e calcolare il valore medio della temperatura.

Indicando con  $T_{min}$  la temperatura minima e con  $T_{max}$  quella massima si ha la  $T_{med}$ :

$$T_{med} = \frac{T_{max} + T_{min}}{2}$$

Il circuito di linearizzazione proposto modifica la caratteristica  $T/V_T$  se è verificata la seguente uguaglianza (fig. 1.6):

$\Delta V_{T2} = \Delta V_{T1}$	$\Delta V_{T2}$ = variazione di tensione nel range di temperatura $T_{med} \div T_{max}$
	$\Delta V_{T1}$ = variazione di tensione nel range di temperatura $T_{min} \div T_{med}$

Detti  $R_{Tmin}$ ,  $R_{Tmed}$  e  $R_{Tmax}$  le resistenze del trasduttore rispettivamente alle temperature  $T_{min}$ ,  $T_{med}$  e  $T_{max}$  (fig. 1.6), dal circuito di figura 1.5 si ricavano i valori delle tensioni  $V_{Tmin}$ ,  $V_{Tmed}$  e  $V_{Tmax}$ .

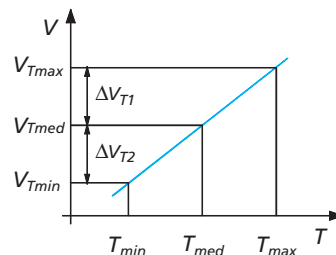


Fig. 1.6

$$V_{Tmin} = \frac{R_{Tmin}}{R_L + R_{Tmin}} \cdot E \quad V_{Tmed} = \frac{R_{Tmed}}{R_L + R_{Tmed}} \cdot E \quad V_{Tmax} = \frac{R_{Tmax}}{R_L + R_{Tmax}} \cdot E$$

Le variazioni di tensioni  $\Delta V_{T1}$  e  $\Delta V_{T2}$ , in funzione delle rispettive variazioni di temperatura, sono:

$$\Delta V_{T1} = V_{Tmed} - V_{Tmin} = \left( \frac{R_{Tmed}}{R_L + R_{Tmed}} - \frac{R_{Tmin}}{R_L + R_{Tmin}} \right) \cdot E \quad [1.1]$$

$$\Delta V_{T2} = V_{Tmax} - V_{Tmed} = \left( \frac{R_{Tmax}}{R_L + R_{Tmax}} - \frac{R_{Tmed}}{R_L + R_{Tmed}} \right) \cdot E \quad [1.2]$$

Uguagliando la [1.1] e la [1.2] e semplificando, si ricava la resistenza di linearizzazione  $R_L$ :

$$\frac{R_{Tmed} \cdot (R_L + R_{Tmin}) - R_{Tmin} \cdot (R_L + R_{Tmed})}{(R_L + R_{Tmin})} = \frac{R_{Tmax} \cdot (R_L + R_{Tmed}) - R_{Tmed} \cdot (R_L + R_{Tmax})}{R_L + R_{Tmax}}$$

$$R_L = \frac{R_{Tmed} \cdot (R_{Tmin} + R_{Tmax}) - 2 \cdot R_{Tmin} \cdot R_{Tmax}}{R_{Tmin} + R_{Tmax} - 2 \cdot R_{Tmed}} \quad [1.3]$$

Poiché la resistenza  $R_L$  dipende solo dai valori delle resistenze  $R_{Tmin}$ ,  $R_{Tmed}$  e  $R_{Tmax}$ , la [1.3] può essere utilizzata per tutti i trasduttori a variazione di resistenza con caratteristiche non lineari, indipendentemente dal tipo di grandezza fisica rilevata.

La linearizzazione della caratteristica del trasduttore può essere fatta sia con un collegamento serie sia con un collegamento parallelo, poiché quest'ultimo deriva dal primo applicando il teorema di Norton (fig. 1.7).

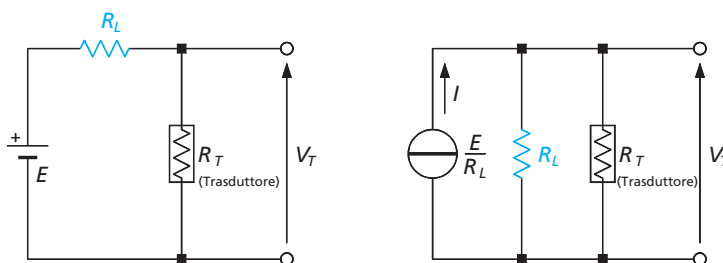


Fig. 1.7

### → Esempio

Si linearizzi la caratteristica della fotoresistenza (fig. 1.8) nel range d'illuminamento  $10 \div 1000$  lx.

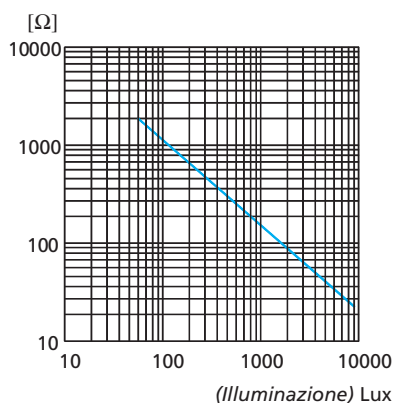


Fig. 1.8

Caratteristiche elettriche	Valori	Unità di misura
Potenza massima	100	mW
Tensione massima	150	V
Resistenza al buio	10	MΩ
Resistenza a 1000 lux	75 - 300	Ω

Dal grafico della figura 1.8 si ricavano i valori delle resistenze per 10 lx, 505 lx e 1000 lx.

$$\begin{aligned} L_{min} &= 10 \text{ lx} & \Rightarrow & R_{Lmin} = 8 \text{ k}\Omega \\ L_{med} &= 505 \text{ lx} & \Rightarrow & R_{Lmed} = 400 \Omega \\ L_{max} &= 1000 \text{ lx} & \Rightarrow & R_{Lmax} = 180 \Omega \end{aligned}$$



Dalla [1.3] si ha:

$$R_L = \frac{R_{I\text{med}} \cdot (R_{I\text{min}} + R_{I\text{max}}) - 2 \cdot R_{I\text{min}} \cdot R_{I\text{max}}}{R_{I\text{min}} + R_{I\text{max}} - 2 \cdot R_{I\text{med}}} = \frac{400 \cdot (8000 + 180) - 2 \cdot 8000 \cdot 180}{180 + 8000 - 2 \cdot 400} = 53,11 \Omega$$

Utilizzando la configurazione parallelo tra la resistenza del trasduttore  $R_T$  e quella di linearizzazione  $R_L$ , si ha:

$$R_{eq-I\text{min}} = \frac{R_{I\text{min}} \cdot R_L}{R_{I\text{min}} + R_L} = \frac{180 \cdot 53,11}{180 + 53,11} = 41,00 \Omega$$

$$R_{eq-I\text{med}} = \frac{400 \cdot 53,11}{400 + 53,11} = 46,88 \Omega$$

$$R_{eq-I\text{max}} = \frac{8000 \cdot 53,11}{8000 + 53,11} = 53,11 \Omega$$

Poiché le variazioni di resistenza in più ed in meno rispetto alla  $R_{eq-I\text{med}}$  sono circa uguali a 6  $\Omega$ , si conclude che il circuito proposto linearizza abbastanza bene la caratteristica del trasduttore.

La **figura 1.9** mostra qualitativamente il processo di linearizzazione che subisce la resistenza del trasduttore di luminosità. La resistenza equivalente  $R_{eqT}$ , parallelo tra la resistenza del termistore  $R_{Lux}$  e quella di linearizzazione  $R_L$ , presenta una lieve non linearità che si riduce sempre di più limitando il range di lavoro.

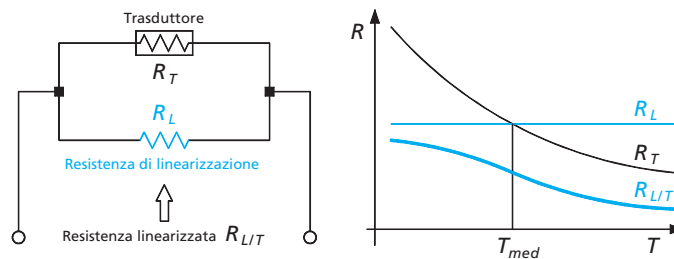


Fig. 1.9

## 1.2 ● Trasduttori di temperatura

Le tecniche di rilevamento dei trasduttori di temperatura sono basate su diversi principi fisici come la dilatazione dei corpi (termometro a mercurio), l'effetto termico (lamina bimetallica), l'effetto termoelettrico (termocoppie). In commercio esistono molti trasduttori a variazione di resistenza costruiti con metalli speciali (termoresistenze) o con semiconduttori.

### 1.2.1 ● Termoresistenza RTD

La termoresistenza, o RTD (*Resistance Temperature Detector*, rilevatore di temperatura a resistenza), è un trasduttore di temperatura a variazione di resistenza (**fig. 1.10**). È costruito con materiali metallici come platino, nichel, tungsteno, ecc. ed è molto utilizzato in applicazione industriale.

La termoresistenza PT100 è un trasduttore di precisione costruito in pellicola di platino ed è un dispositivo robusto, economico con dimensioni contenute.

Ha una caratteristica  $R/T$  lineare in un vasto range ed è tarata con raggio laser (**fig. 1.11**).



Fig. 1.10

**Fig. 1.11** Termoresistenza al platino PT100.

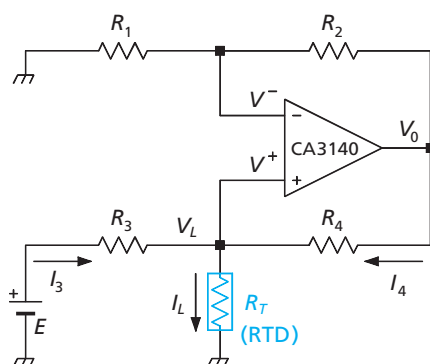


La resistenza  $R_T$  è una funzione lineare della temperatura  $T$  [1.4].

$$R_T = R_0 \cdot (1 + \alpha \cdot T) \quad [1.4]$$

dove:

- $R_T$  è il valore della resistenza alla temperatura generica  $T$ ;
- $R_0 = 100 \, \Omega$  è il valore nominale della resistenza alla temperatura  $T = 0 \, ^\circ\text{C}$ ;
- $\alpha = 3,85 \cdot 10^{-3} \, ^\circ\text{C}^{-1}$  è una costante dimensionale;
- $T$  è la temperatura generica in gradi Celsius.



**Fig. 1.12**

Per convertire le variazioni di resistenza  $R_T$  in variazioni di tensione  $V_u$  si possono utilizzare semplici circuiti, come il ponte di Wheatstone, o circuiti complessi che alimentano la termoresistenza con intensità di corrente costante  $I$ . In quest'ultimo caso si utilizzano circuiti con amplificatori operazionali o integrati dedicati.

In figura 1.12 è riportato lo schema di un generatore di corrente costante  $I_L$  realizzato con l'amplificatore operazionale CA3140. La resistenza di carico  $R_T$  è la resistenza del trasduttore PT100 ed ha un terminale collegato a massa.

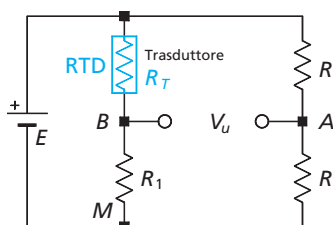
Se è  $\frac{R_4}{R_3} = \frac{R_2}{R_1}$  e l'amplificatore operazione è ideale, l'intensità di corrente  $I_L$  che scorre nella

termoresistenza  $R_T$  è costante ed è  $I_L = \frac{E}{R_3}$ .

Per modificare il valore della  $I_L$  è necessario modificare i valori di  $R_3$  e di  $R_1$  perché deve essere sempre rispettata la condizione  $\frac{R_4}{R_3} = \frac{R_2}{R_1}$ .

### → Esempio

Si calcoli la tensione  $V_u$  in funzione della temperatura  $T$  prodotta da una termoresistenza PT100 nell'ipotesi che il range di temperatura sia compreso tra  $0 \, ^\circ\text{C}$  e  $100 \, ^\circ\text{C}$  (fig. 1.13).



**Fig. 1.13**

Si bilancia il ponte di Wheatstone a temperatura  $T = 0 \, ^\circ\text{C}$  ponendo  $R_1 = 100 \, \Omega$ . Dalla figura 1.13 si ricava:

$$V_u = V_{AM} - V_{BM}$$

$$V_u = \frac{R}{2 \cdot R} \cdot E - \frac{R_1}{R_1 + R_T} \cdot E = \left( \frac{1}{2} - \frac{R_1}{R_1 + R_T} \right) \cdot E \quad [1.5]$$

Per  $T = 0 \, ^\circ\text{C}$  la [1.5] diviene:

$$V_u = \left( \frac{1}{2} - \frac{100}{100 + 100} \right) \cdot E = 0 \, \text{V}$$

Un aumento della temperatura  $T$  provoca un aumento del valore della  $R_T$  e quindi lo sbilanciamento del ponte.

Per  $T \neq 0^\circ\text{C}$ , dalla [1.5] si ha:

$$V_u = \frac{R_T - R_1}{2 \cdot (R_1 + R_T)} \cdot E \quad [1.6]$$

La tensione d'uscita  $V_u = f(T)$  non è una funzione lineare della temperatura perché la [1.6] è una funzione razionale fratta della variabile dipendente  $R_T$ . La legge di variazione della tensione  $V_u$  non può essere rappresentata con l'equazione di una retta. In pratica, il circuito utilizzato per la conversione  $T/V_u$  introduce una non linearità anche se ha il pregio di fornire una tensione d'uscita  $V_u = 0\text{ V}$  per una qualsiasi temperatura minima (fattore di *offset*).

Per verificare la non linearità introdotta dal sistema di conversione proposto, si utilizza la [1.4] per calcolare i valori della resistenza  $R_T$  del trasduttore e la [1.6] per calcolare i valori della tensione di uscita  $V_u$  (tab. 1.1).

Limitando a 2 mA l'intensità di corrente  $I$  che attraversa la termoresistenza al fine di trascurare l'effetto Joule, si ricava il valore della f.e.m.  $E$  per  $T = 0^\circ\text{C}$  (condizione peggiore):

$$E = (R_T + R_1) \cdot 2 \cdot 10^{-3} = (100 + 100) \cdot 2 \cdot 10^{-3} = 0,4\text{ V}$$

Incrementando il valore della temperatura di  $10^\circ\text{C}$ , dalla [1.6] si ricavano i valori di tensione riportati nella tabella 1.1.

TAB. 1.1

$T [^\circ\text{C}]$	0	10	20	30	40	50	60	70	80	90	100
$R_T [\Omega]$	100	103,85	107,70	111,55	115,40	119,25	123,10	126,95	130,80	134,65	138,50
$V_u [\text{mV}]$	0	3,77	7,41	10,91	14,29	17,55	20,70	23,74	26,68	29,53	32,28
$\Delta V [\text{mV}]$	-	3,77	3,64	3,50	3,38	3,26	3,15	3,04	2,94	2,85	2,75

L'analisi dei dati della tabella 1.1 conferma che ad uguali incrementi della temperatura  $T$  non corrispondono aumenti proporzionali della tensione d'uscita  $V_u$ .

Considerato che la tensione  $V_u$  è dell'ordine dei mV, è necessario aumentarne il valore con amplificatori ad elevato guadagno.

Nella figura 1.14 è proposto un circuito d'amplificazione con l'integrato INA111 avente le seguenti caratteristiche:

- bassa tensione di offset;
- elevatissima impedenza d'ingresso ( $10^{12}\ \Omega$ ) per lo stadio d'ingresso a FET;
- alimentazione duale  $\pm 6\text{ V} \div \pm 18\text{ V}$ ;
- guadagni  $G$  compresi tra 1 e 1000 definiti mediante il resistore  $R_G$ ;
- $V_0 = G \cdot (V_{IN}^+ - V_{IN}^-)$  con  $G = 1 + \frac{50\text{ k}\Omega}{R_G}$ ;
- costo molto contenuto.

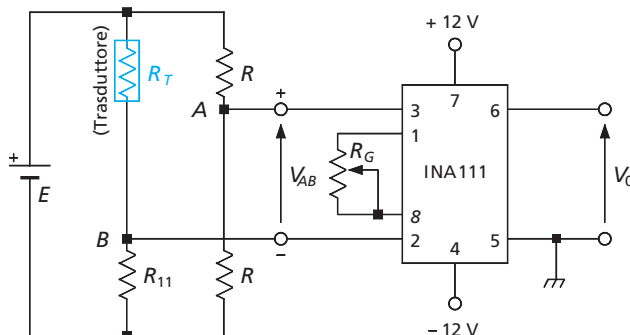


Fig. 1.14

Per una tensione d'uscita  $V_0 = 8 \text{ V}$ , minore di quella d'alimentazione dell'integrato, si ha:

$$G = \frac{V_0}{V_{AB}} = \frac{8}{32,28 \cdot 10^{-3}} = 247,83 \quad R_G = \frac{50 \cdot 10^3}{G-1} = \frac{50\,000}{29,26} = 1708,8$$



## 1.2.2 ● Termistori

I termistori (fig. 1.15) sono trasduttori di temperatura a variazione di resistenza realizzati con materiali aventi un coefficiente di temperatura elevato e si distinguono in PTC e NTC. I primi hanno un coefficiente di temperatura positivo e sono caratterizzati da una caratteristica  $T/R$  esponenziale crescente all'aumentare della temperatura; mentre i secondi hanno un coefficiente di temperatura negativo e la caratteristica  $T/R$  ha un andamento esponenziale decrescente all'aumentare della temperatura.

La legge di variazione della resistenza  $R_T$  con la temperatura  $T$  di un termistore NTC è:

$$R_T = A \cdot e^{\frac{B}{T}} \quad [1.7]$$

dove:

- $R_T$  è la resistenza del trasduttore alla temperatura generica  $T$ ;
- $A$  e  $B$  sono costanti dipendenti dal materiale con  $2000 \text{ K} < B < 5500 \text{ K}$  (fig. 1.16);
- $T$  è la temperatura generica espressa in gradi Kelvin a differenza delle termoresistenze.

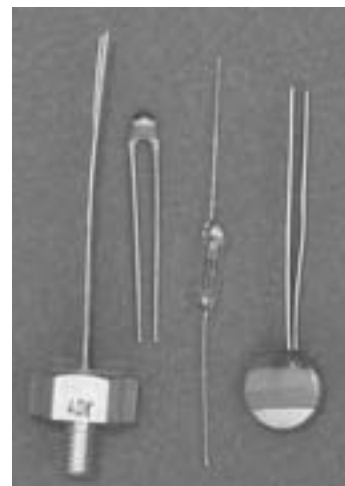
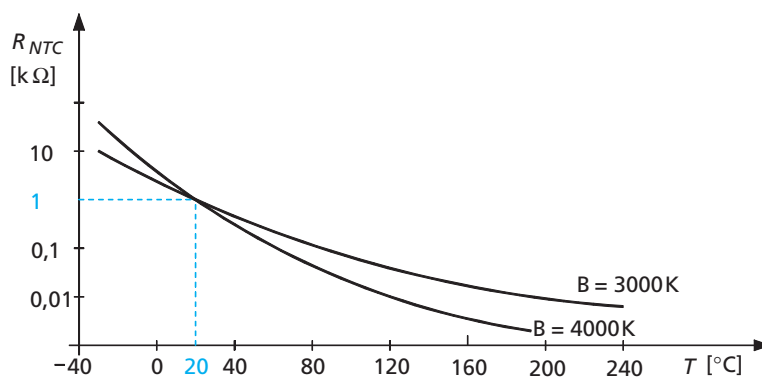


Fig. 1.15

Fig. 1.16 Caratteristiche qualitative NTC.



Dette  $R_{T1}$  e  $R_{T2}$  la resistenza rispettivamente alla temperatura  $T_1$  e alla temperatura  $T_2$ , dalla [1.7] si ricava:

$$R_{T2} = A \cdot e^{\frac{B}{T_2}} \quad [1.8]$$

$$R_{T1} = A \cdot e^{\frac{B}{T_1}} \quad [1.9]$$

Dividendo membro a membro la [1.8] e la [1.9] si ha:

$$\frac{R_{T2}}{R_{T1}} = \frac{e^{\frac{B}{T_2}}}{e^{\frac{B}{T_1}}} = e^{B \cdot \left( \frac{1}{T_2} - \frac{1}{T_1} \right)} \quad R_{T2} = R_{T1} \cdot e^{B \cdot \left( \frac{1}{T_2} - \frac{1}{T_1} \right)}$$

Se  $T_1$  è riferita alla temperatura ambiente  $T_a = 293 \text{ K}$  ( $T_a = 20^\circ \text{C}$ )<sup>1</sup>, e  $T_2$  indica una temperatura generica  $T$ , si ha:

$$R_T = R_{Ta} \cdot e^{B \cdot \left( \frac{1}{T} - \frac{1}{Ta} \right)} = R_{Ta} \cdot e^{B \cdot \left( \frac{T_a - T}{Ta \cdot T} \right)} \quad [1.10]$$

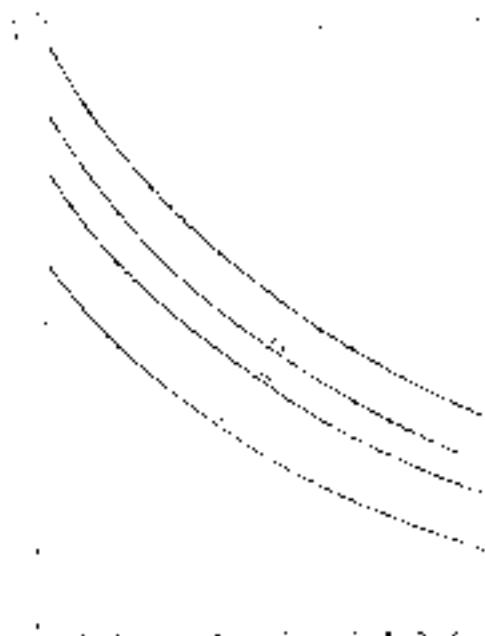
La caratteristica dei termistori NTC può essere linearizzata utilizzando sia la [1.3] sia la [1.11], quest'ultima suggerita dalle case costruttrici perché è indipendente dal range di funzionamento:

$$R_L = R_{Tmed} \cdot \frac{B - 2 \cdot T_{med}}{B + 2 \cdot T_{med}} \quad [1.11]$$

dove:

- $T_{med}$  è il valore medio della temperatura entro il campo di lavoro;
- $R_{Tmed}$  è la resistenza della NTC corrispondente alla temperatura media  $T_{med}$ ;
- $B$  è la costante dimensionale del termistore.

Nella **figura 1.17** sono riportate le caratteristiche di una famiglia di NTC della serie K11 Siemens.



**Fig. 1.17**

### → Esempio 1

Si linearizzi la caratteristica  $R/T$  del termistore NTC-K 25 della Siemens nel range di temperatura (**tab. 1.2**).

**TAB. 1.2**

Tipo	$R \text{ } [\Omega]$	Tolleranza	$B \text{ } [\text{K}]$	Colore	Codice
K 25 – 1k	1000	20 %	3530	Violetto	Q 63 025 – K102 – M

Si calcolano i valori delle resistenze  $R_{Tmin}$ ,  $R_{Tmed}$  e  $R_{Tmax}$ :

$$T_{min} = (0^\circ \text{C} \Rightarrow 273 \text{ K}) \Rightarrow R_{Tmin} = 2417 \text{ } \Omega \quad [\Delta R\% = -641\%]$$

$$T_{med} = (50^\circ \text{C} \Rightarrow 323 \text{ K}) \Rightarrow R_{Tmed} = 326 \text{ } \Omega$$

$$T_{max} = (100^\circ \text{C} \Rightarrow 373 \text{ K}) \Rightarrow R_{Tmax} = 75 \text{ } \Omega \quad [\Delta R\% = +76,9\%]$$

Dalla [1.3], si ha:

$$R_L = \frac{R_{Tmed} \cdot (R_{Tmin} + R_{Tmax}) - 2 \cdot R_{Tmin} \cdot R_{Tmax}}{R_{Tmin} + R_{Tmax} - 2 \cdot R_{Tmed}} = \frac{326 \cdot (75 + 2417) - 2 \cdot 75 \cdot 2417}{75 + 2417 - 2 \cdot 326} = 244 \text{ } \Omega$$

<sup>1</sup> La temperatura di riferimento può essere diversa. In tal caso è fornita dal costruttore.

La linearizzazione della caratteristica può essere verificata mediante un collegamento parallelo (fig. 1.18) tra la resistenza del trasduttore  $R_T$  e quella di linearizzazione  $R_L$  e calcolando i valori delle resistenze equivalenti alle diverse temperature:

$$R_{eqTmin}(0\text{ }^{\circ}\text{C} \Rightarrow 273\text{ K}) = \frac{R_{Tmin} \cdot R_L}{R_{Tmin} + R_L} = \frac{2417 \cdot 244}{2417 + 244} = 221,62\text{ }\Omega$$

$$R_{eqTmed}(50\text{ }^{\circ}\text{C} \Rightarrow 323\text{ K}) = \frac{326 \cdot 244}{326 + 244} = 139,50\text{ }\Omega$$

$$R_{eqTmax}(100\text{ }^{\circ}\text{C} \Rightarrow 373\text{ K}) = \frac{75 \cdot 244}{75 + 244} = 57,35\text{ }\Omega$$

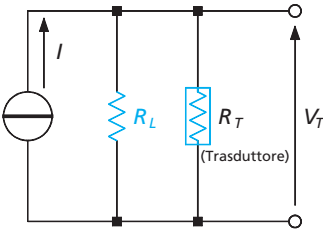


Fig. 1.18

I risultati ottenuti sono riportati in tabella 1.3 e sono rappresentati in figura 1.19.

TAB. 1.3

$T\text{ }[^{\circ}\text{C}]$	0	50	100
$T\text{ [K]}$	273	323	373
$R_T\text{ }[\Omega]$	2417	326	75
$R_{LT}\text{ }[\Omega]$	221,6	139,5	57,3
$\Delta R\text{ }[\Omega]$		82,1	82,2

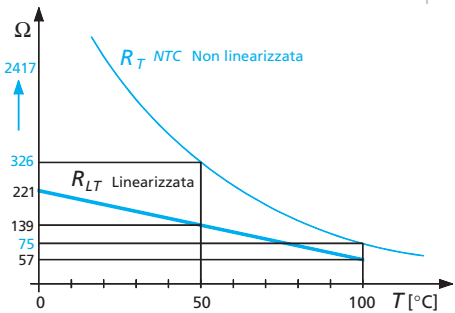


Fig. 1.19

### → Esempio 2

Si utilizzi un termistore NTC K25-1k per progettare un circuito che fornisca una tensione d'uscita  $0\text{ V} < V_0 < 8\text{ V}$  per un range di temperatura  $20\text{ }^{\circ}\text{C} < T < 80\text{ }^{\circ}\text{C}$ .

Le caratteristiche elettriche della NTC sono:

- potenza a  $60\text{ }^{\circ}\text{C}$   $P = 400\text{ mW}$ ;
- temperatura di riferimento  $T_0 = 20\text{ }^{\circ}\text{C}$ ;
- campo di temperatura  $-25\text{ }^{\circ}\text{C} \div +100\text{ }^{\circ}\text{C}$ .

Tipo	$R\text{ }[\Omega]$	Tolleranza	$B\text{ [K]}$	Colore	Codice
K 25 – 1k	1000	20 %	3530	Violetto	Q 63 025 – K102 – M

Utilizzando la [1.10] si possono tabulare i valori assunti dal trasduttore NTC nel campo di funzionamento per incrementi di temperatura  $\Delta T$  costanti.

Ad esempio, per  $\Delta T = 10^\circ\text{C}$  si trovano i valori riportati nella [tabella 1.4](#).

**TAB. 1.4**

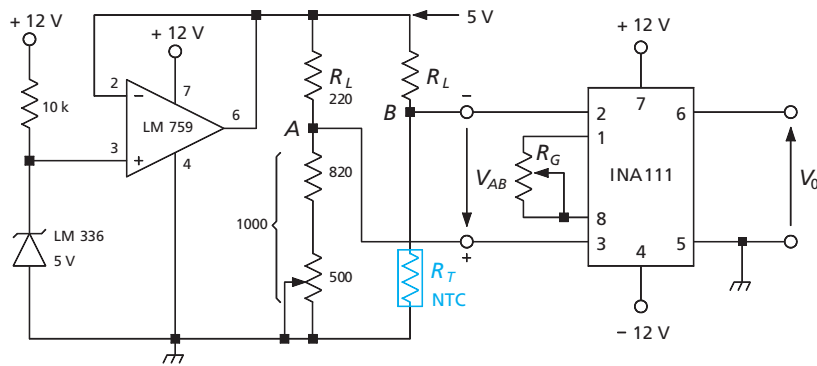
$T [^\circ\text{C}]$	20	30	40	50	60	0	80
$R [\Omega]$	1000	671	463	326	235	172	129

La resistenza di linearizzazione  $R_L$ , calcolata applicando la [1.3], è:

$$R_L = \frac{R_{T_{med}} \cdot (R_{T_{min}} + R_{T_{max}}) - 2 \cdot R_{T_{min}} \cdot R_{T_{max}}}{R_{T_{min}} + R_{T_{max}} - 2 \cdot R_{T_{med}}} = \frac{326 \cdot (129 + 10^3) - 2 \cdot 129 \cdot 10^3}{129 + 10^3 - 2 \cdot 326} = 230,72 \Omega$$

Per la conversione  $T/V$  si utilizza il ponte di Wheatstone che permette l'offset, ( $V_{AB} = 0\text{ V}$ ) per la temperatura  $T$  diversa da  $0^\circ\text{C}$  ([fig. 1.20](#)):

- l'integrato LM336-5V fornisce la tensione di riferimento  $V_{REF} = 5\text{ V}$ ;
- l'amplificatore operazionale di potenza LM759 alimenta il ponte di Wheatstone ed eroga l'opportuna intensità di corrente assorbita dal ponte;
- l'amplificatore per strumentazione INA111 è utilizzato per definire il guadagno.



**Fig. 1.20**

Poiché la massima intensità di corrente assorbita dalla NTC si ha quando è  $T = 80^\circ\text{C}$ , per evitare effetti di autoriscaldamento si ipotizza di far dissipare al trasduttore una potenza di  $50\text{ mW}$ . Se la tensione di alimentazione è uguale a  $5\text{ V}$ , la potenza dissipata dalla NTC è:

$$I_{NTC}(80^\circ\text{C}) = \frac{5}{220 + 129} = \frac{5}{349} \cong 14\text{ mA}$$

$$P_{NTC} = R_{NTC} \cdot I_{NTC}^2 = 129 \cdot (14 \cdot 10^{-3})^2 = 25\text{ mW} < 50\text{ mW}$$

Si bilancia il ponte per la temperatura  $T = 20^\circ\text{C}$  con la resistenza  $R_1 = 1000\ \Omega$  costituita da una resistenza fissa pari a  $820\ \Omega$  e da un trimmer resistivo di  $500\ \Omega$ .

La tensione  $V_{AB}$  per la temperatura  $T = 80^\circ\text{C}$  è:

$$V_{AB}(T = 80^\circ\text{C}) = \left( \frac{10^3}{10^3 + 220} - \frac{129}{220 + 129} \right) \cdot 5 = 2,25\text{ V}$$

Poiché la tensione d'uscita  $V_0$  dell'integrato INA111 deve essere uguale a  $8\text{ V}$ , il guadagno  $G$  e la resistenza  $R_G$  sono rispettivamente uguali a:

$$G = \frac{V_0}{V_{AB}(80^\circ\text{C})} = \frac{8}{2,25} = 3,55 \quad R_G = \frac{50 \cdot 10^3}{G - 1} = \frac{50 \cdot 10^3}{2,55} = 22,22\text{ k}\Omega$$



### 1.3 ● Trasduttore di luminosità

I **fotoresistori** (fig. 1.21) sono costituiti con materiali semiconduttori leggermente drogati (solfuro di cadmio, solfuro di piombo, ecc.). Quando la superficie sensibile del fotoresistore viene esposta alla luce, l'energia raggiante assorbita provoca la rottura dei legami covalenti, generando cariche libere (coppie lacune-elettroni).  
Tale fenomeno provoca l'aumento della conducibilità e una conseguente diminuzione della resistenza del semiconduttore.  
La variazione della resistenza  $R$  del trasduttore in funzione dell'illuminamento  $E^2$  è data dalla seguente legge:

$$R = A \cdot E^{-\alpha}$$



Fig. 1.21

dove:

- $A$  è una costante dimensionale che dipende dalla forma geometrica della superficie esposta alla luce;
- $E$  è l'illuminamento (flusso luminoso incidente per una superficie unitaria);
- $\alpha$  è una costante adimensionale minore di 1 che dipende dalle tecnologie utilizzate per la costruzione del dispositivo.

#### 1.3.1 ● Il fotoresistore NORP-12

La caratteristica resistenza/illuminamento del trasduttore NORP-12 [RS 651-507] presenta un'accentuata non linearità ed una pendenza negativa (fig. 1.22).

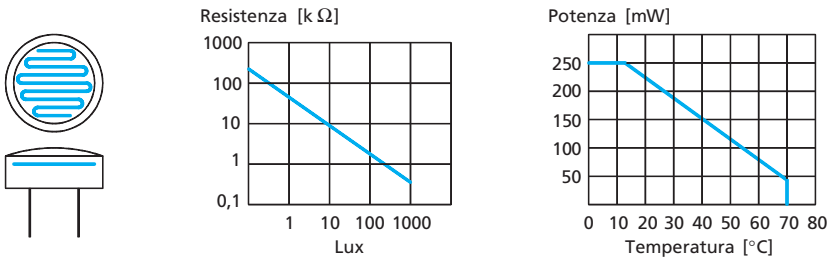


Fig. 1.22

Nella **tabella 1.5** sono riportate le caratteristiche elettriche della fotoresistenza NORP-12.

TAB. 1.5

Caratteristiche elettriche	Valori	Unità di misura
Resistenza d'oscurità (min.)	1	MΩ
Resistenza di cella a 10 lux	9	kΩ
Resistenza di cella a 1000 lux	400	Ω
Tensione max di picco (AC e DC)	320	V
Corrente max	75	mA
Potenza max (a 25 °C)	250	mW
Range di temperatura	- 60 ÷ 75	°C
Capacità d'oscurità (tipica)	3,6	pF
Reazione spettrale di picco	0,53	μm

<sup>2</sup> Nel Sistema Internazionale (SI) il *lux*, il cui simbolo è [lx], è l'unità di misura dell'illuminamento  $E$ . Il lux, rapporto tra il lumen ed il m<sup>2</sup>, è l'illuminamento prodotto dal flusso luminoso di 1 lm (lumen) che incide perpendicolarmente su una superficie di 1 m<sup>2</sup>. A volte il lux è indicato con il phot (ph = 10<sup>4</sup> lx) o con il foot (fc = 10,76 lx).



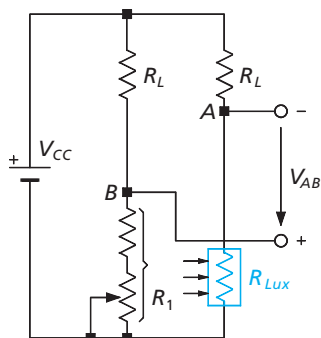


Fig. 1.23

Linearizzare la curva caratteristica dei fotoresistori, per renderla crescente all'aumentare dell'illuminamento e per convertire le variazioni di resistenza in variazioni di tensione, può essere eseguita con gli stessi criteri utilizzati nel dimensionamento dei circuiti delle NTC.

Il circuito proposto in figura 1.23 realizza:

- la conversione resistenza/tensione;
- la linearizzazione della caratteristica con la resistenza  $R_L$ ;
- l'equilibrio del ponte con la resistenza  $R_1$ ;
- una tensione d'uscita  $V_{AB}$  crescente all'aumentare dell'illuminamento.

### → Esempio

Si utilizzi il fotoresistore NORP-12 per progettare un circuito in modo da avere una tensione d'uscita  $0\text{ V} < V_0 < 10\text{ V}$  quando l'illuminamento varia nell'intervallo  $10\text{ lux} \div 1000\text{ lux}$ .

Dalla tabella 1.5 e dalla figura 1.22 si ricava la tabella 1.6.

TAB. 1.6

Lux [lx]	10	505	1000
R [Ω]	9000	$\cong 1000$	400

La resistenza di linearizzazione  $R_L = 305\ \Omega$ . Per la conversione  $\text{Lux}/\text{V}$  si utilizza il ponte di Wheatstone che permette di avere una  $V_{AB} = 0\text{ V}$  per l'illuminamento di  $10\text{ lx}$ .

Il circuito, comprensivo di resistenza di linearizzazione  $R_L = 330\ \Omega$  (valore commerciale) è riportato in figura 1.24.

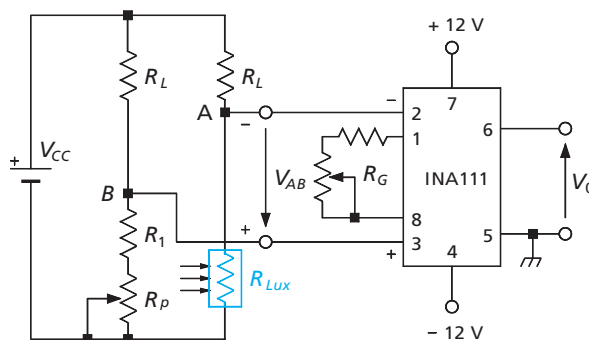


Fig. 1.24

Nel dimensionare i componenti si suppone che nella condizione di funzionamento più gravosa, corrispondente ad un illuminamento di  $1000\text{ lx}$  ( $R_{Lux} = 400\ \Omega$ ), il fotoresistore sia attraversato da una intensità di corrente  $I < 75\text{ mA}$ .

Con i componenti scelti ed una tensione d'alimentazione  $V_{CC} = 12\text{ V}$ , si ha:

$$I(R_{Lux} = 400\ \Omega) = \frac{V_{CC}}{R_{Lux} + R_L} = \frac{12}{730} = 16,4\text{ mA}$$

La resistenza di bilanciamento ( $R_1 + R_p$ ), dimensionata per l'illuminamento di  $10\text{ lux}$ , e la tensione  $V_{AB}$  in corrispondenza del massimo illuminamento sono:

$$(R_1 + R_p) = \frac{R_{Lux} \cdot R_L}{R_L} = 9\text{ k}\Omega \quad (8,2\text{ k}\Omega + 1\text{ k}\Omega)$$

$$V_{AB} = \left( \frac{9000}{9000 + 330} - \frac{400}{330 + 400} \right) \cdot 12 = 5,00\text{ V}$$

Poiché il range della tensione d'uscita  $V_0$  richiesto è compreso tra 0 V e 10 V, si ha:

$$G = \frac{V_0}{V_{AB} (1000 \text{ lux})} = \frac{10}{5,00} = 2 \quad R_G = \frac{50 \cdot 10^3}{G - 1} = \frac{50 \cdot 10^3}{1} = 50 \text{ k}\Omega$$



## 1.4 ● Termocoppia



Fig. 1.25

La termocoppia (fig. 1.25) è un trasduttore di temperatura, definito anche trasduttore attivo o autogenerante perché non richiede circuiti d'alimentazione, molto utilizzato in ambito industriale.

Il principio di funzionamento di una termocoppia è basato sull'effetto Seebeck, secondo cui due metalli omogenei, chimicamente diversi e saldati alle loro estremità, danno origine ad una corrente di debole intensità (*termocorrente*) quando le due saldature (*giunti*) sono mantenute a temperature diverse (fig. 1.26).



Fig. 1.26

Se si apre una saldatura della termocoppia e si collega un voltmetro ad alta impedenza agli estremi liberi, la f.e.m. misurata è (fig. 1.27):

$$V = \alpha \cdot (T_C - T_F)$$

dove:

- $T_C$  è la temperatura del giunto caldo  $J_C$ ;
- $T_F$  è la temperatura del giunto freddo  $J_F$ ;
- $\alpha$  è il coefficiente di proporzionalità di Seebeck  $\left[ \frac{\mu\text{V}}{^\circ\text{C}} \right]$ .

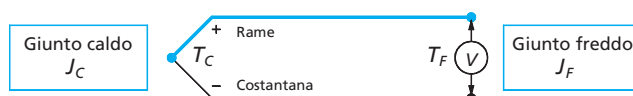


Fig. 1.27

In particolare se è  $T_F = 0^\circ\text{C}$  la tensione misurata è direttamente proporzionale alla temperatura  $T_C$  del giunto caldo:

$$V_C = \alpha \cdot T_C$$

I materiali più comunemente impiegati per la realizzazione delle termocoppie riferiti al campo di lavoro, alla sensibilità ed al tipo sono riportati nella tabella 1.7.

TAB. 1.7

Conduttore positivo (+)	Conduttore negativo (-)	Tipo	Range [°C]	Sensibilità [mV/°C]
Rame	Costantana	T	- 200 ÷ 370	40,5
Cromo	Costantana	E	- 200 ÷ 900	67,9
Ferro	Costantana	J	0 ÷ 760	52,6
Cromo	Alumel	K	- 200 ÷ 1250	38,8